

Шон Харрис "CISSP All-In-One Exam Guide"

CISSP. Руководство для подготовки к экзамену

Домен 01. Информационная безопасность и управление рисками

1. Управление безопасностью

- 1.1. Распределение обязанностей по управлению безопасностью
- 1.2. Подход "сверху-вниз"

2. Администрирование безопасности и защитные меры

- 2.1. Основные принципы безопасности
- 2.2. Определения безопасности
- 2.3. Безопасность посредством неизвестности

3. Организационная модель безопасности

- 3.1. Компоненты программы безопасности
- 3.2. Стандарты безопасности
- 3.3. Управление безопасностью на стратегическом уровне

4. Управление информационными рисками

- 4.1. Кто действительно разбирается в управлении рисками?
- 4.2. Политика управления информационными рисками
- 4.3. Группа управления рисками (IRM-группа)

5. Анализ рисков

- 5.1. Группа анализа рисков
- 5.2. Ценность информации и активов
- 5.3. Определение стоимости и ценности
- 5.4. Идентификация угроз
- 5.5. Анализ сбоев и дефектов
- 5.6. Количественный анализ рисков
- 5.7. Качественный анализ рисков
- 5.8. Количественный или качественный
- 5.9. Защитные механизмы
- 5.10. Обобщая сказанное ранее
- 5.11. Общий риск и Остаточный риск
- 5.12. Обработка риска

6. Политики, стандарты, базисы, руководства и процедуры

- 6.1. Политика безопасности
- 6.2. Стандарты
- 6.3. Базисы
- 6.4. Руководства

- 6.5. Процедуры
- 6.6. Внедрение

7. Классификация информации

- 7.1. Управление классифицированными данными

8. Уровни ответственности

- 8.1. Совет Директоров
- 8.2. Высшее исполнительное руководство
- 8.3. Владелец данных
- 8.4. Ответственный за хранение данных
- 8.5. Владелец системы
- 8.6. Администратор безопасности
- 8.7. Аналитик по безопасности
- 8.8. Владелец приложения
- 8.9. Супервизор
- 8.10. Аналитик управления изменениями
- 8.11. Аналитик данных
- 8.12. Владелец процесса
- 8.13. Поставщик решения
- 8.14. Пользователь
- 8.15. Менеджер по технологиям
- 8.16. Аудитор
- 8.17. Зачем так много ролей?!

9. Персонал

- 9.1. Структура
- 9.2. Правила приема на работу
- 9.3. Контроль сотрудников
- 9.4. Увольнение

10. Обучение (тренинги) по вопросам безопасности

- 10.1. Различные типы обучения (тренинга) по вопросам безопасности
- 10.2. Оценка результатов обучения
- 10.3. Специализированное обучение по безопасности

11. Резюме

12. Тест

Домен 02. Управление доступом

1. Обзор управления доступом

2. Принципы безопасности

3. Идентификация, аутентификация, авторизация и подотчетность

- 3.1. Идентификация и аутентификация

- Управление идентификацией
- Биометрия
- Пароли
- Когнитивные пароли
- Одноразовые пароли
- Токены
- Криптографические ключи
- Парольные фразы
- Карты памяти
- Смарт-карты

- 3.2. Авторизация

- Критерии доступа
- Отсутствие доступа «по умолчанию»
- Принцип «необходимо знать»
- Единый вход

4. Модели управления доступом

- 4.1. Дискреционное управление доступом
- 4.2. Мандатное управление доступом
- 4.3. Ролевое управление доступом

5. Техники и технологии управления доступом

- 5.1. Управление доступом на основе правил
- 5.2. Ограниченный пользовательский интерфейс
- 5.3. Матрица контроля доступа
 - Таблицы разрешений
 - Списки контроля доступа
- 5.4. Контентно-зависимое управление доступом
- 5.5. Контекстно-зависимое управление доступом

6. Администрирование доступа

- 6.1. Централизованное администрирование управления доступом
 - RADIUS
 - TACACS
 - Diameter
- 6.2. Децентрализованное администрирование управления доступом

7. Методы управления доступом

- 7.1. Уровни управления доступом
- 7.2. Административный уровень
- 7.3. Физический уровень
- 7.4. Технический уровень

8. Типы управления доступом

- 8.1. Превентивные: Административные
- 8.2. Превентивные: Физические
- 8.3. Превентивные: Технические

9. Подотчетность

- 9.1. Анализ журналов регистрации событий
- 9.2. Мониторинг нажатия клавиш
- 9.3. Защита данных аудита и журналов регистрации событий

10. Практика управления доступом

- 10.1. Несанкционированное разглашение информации
 - Повторное использование объекта
 - Защита от утечки информации по техническим каналам

11. Мониторинг управления доступом

- 11.1. Выявление вторжений
 - IDS уровня сети
 - IDS уровня хоста
 - Выявление вторжений на основе знаний или сигнатур
 - IDS на основе состояния
 - Выявление вторжений на основе статистических аномалий
 - IDS на основе аномалий протоколов
 - IDS на основе аномалий трафика
 - IDS на основе правил
 - Сенсоры IDS
 - Сетевой трафик
- 11.2. Системы предотвращения вторжений
 - Хосты-приманки
 - Сетевые снифферы

12. Несколько угроз управлению доступом

- 12.1. Атака по словарю
- 12.2. Атака полного перебора (брутфорс-атака)
- 12.3. Подделка окна регистрации в системе

- 12.4. Фишинг
- 12.5. Кража личности

13. Резюме

14. Тест

Домен 03. Архитектура и модель безопасности

1. Архитектура компьютера

- 1.1. Центральный процессор
- 1.2. Многопроцессорная обработка
- 1.3. Архитектура операционной системы
 - Управление процессами
 - Управление потоками
 - Диспетчеризация процессов
- 1.4. Работа процессов
- 1.5. Управление памятью
- 1.6. Типы памяти
 - Память с произвольным доступом
 - Память только для чтения
 - Кэш-память
 - Отображение памяти
 - Утечки памяти
- 1.7. Виртуальная память
- 1.8. Режимы процессора и кольца защиты
- 1.9. Архитектура операционной системы
- 1.10. Домены
- 1.11. Разделение на уровни и скрытие данных
- 1.12. Эволюция терминологии
- 1.13. Виртуальные машины
- 1.14. Дополнительные устройства хранения
- 1.15. Управление устройствами ввода/вывода
 - Прерывания

2. Архитектура системы

- 2.1. Определенные подмножества субъектов и объектов
- 2.2. Доверенная компьютерная база
- 2.3. Периметр безопасности
- 2.4. Монитор обращений и ядро безопасности

- 2.5. Политика безопасности
- 2.6. Принцип наименьших привилегий

3. Модели безопасности

- 3.1. Модель конечных автоматов
- 3.2. Модель Bell-LaPadula
- 3.3. Модель Biba
- 3.4. Модель Clark-Wilson
- 3.5. Модель информационных потоков
 - Скрытые каналы
- 3.6. Модель невлияния
- 3.7. Сетчатая модель
- 3.8. Модель Brewer and Nash
- 3.9. Модель Graham-Denning
- 3.10. Модель Harrison-Ruzzo-Ulman

4. Режимы безопасности функционирования

- 4.1. Специальный режим безопасности
- 4.2. Режим повышенной безопасности системы
- 4.3. Раздельный режим безопасности
- 4.4. Многоуровневый режим безопасности
- 4.5. Доверие и гарантии

5. Методы оценки систем

- 5.1. Зачем проводить оценку продукта?
- 5.2. Оранжевая Книга
- 5.3. Оранжевая книга и Радужная серия
- 5.4. Красная книга
- 5.5. ITSEC
- 5.6. Общие критерии

6. Сертификация vs. Аккредитация

- 6.1. Сертификация
- 6.2. Аккредитация

7. Открытые vs. Закрытые системы

- 7.1. Открытые системы
- 7.2. Закрытые системы

8. Корпоративная архитектура

9. Анализ нескольких угроз

- 9.1. Закладки для поддержки

- 9.2. Атаки времени проверки / времени использования
- 9.3. Переполнение буфера

10. Резюме

11. Тест

Домен 04. Физическая безопасность и безопасность окружения

1. Введение в Физическую безопасность

2. Процесс планирования

- 2.1. Предотвращение преступлений посредством проектирования окружения
 - Естественное управление доступом
 - Естественное наблюдение
 - Естественное укрепление территории
- 2.2. Проектирование программы физической безопасности
 - Здание
 - Конструкция
 - Точки входа
 - Двери
 - Окна
 - Внутренние помещения
 - Серверные и кроссовые помещения

3. Защита активов

4. Внутренние системы поддержки и снабжения

- 4.1. Электроэнергия
 - Защита электроснабжения
 - Проблемы электропитания
 - Превентивные меры и Хорошие практики
- 4.2. Проблемы окружения
- 4.3. Вентиляция
- 4.4. Предотвращение, выявление и тушение пожара
 - Типы пожарных датчиков
 - Тушение пожара
 - Водяные спринклеры

5. Безопасность периметра

- 5.1. Контроль доступа в здание и помещения
 - Замки

- 5.2. Контроль доступа персонала
- 5.3. Механизмы защиты внешних границ
 - Ограждения
 - Столбики ограждения
 - Освещение
 - Устройства наблюдения
 - Устройства видеозаписи
- 5.4. Системы выявления вторжений
- 5.5. Патрульные и охранники
- 5.6. Собаки
- 5.7. Контроль физического доступа
- 5.8. Тестирование и тренировки

6. Резюме

7. Тест

Домен 05. Телекоммуникационная и сетевая безопасность

1. Эталонная модель взаимодействия открытых систем

- 1.1. Протокол
 - Прикладной уровень
 - Представительский уровень
 - Сеансовый уровень
 - Транспортный уровень
 - Сетевой уровень
 - Канальный уровень
 - Физический уровень
- 1.2. Функции и Протоколы модели OSI
- 1.3. Совместная работа уровней

2. TCP/IP

- 2.1. TCP
- 2.2. Адресация IP
- 2.3. IPv6

3. Типы передачи

- 3.1. Аналоговая и цифровая
- 3.2. Асинхронная и синхронная
- 3.3. Однополосная и широкополосная

4. Организация локальных вычислительных сетей

- 4.1. Топология сети
- 4.2. Технологии доступа к среде LAN
 - Ethernet
 - Token Ring
 - FDDI
- 4.3. Кабели
 - Коаксиальный кабель
 - Витая пара
 - Оптоволоконный кабель
 - Проблемы, связанные с кабелями
 - Пожарные рейтинги кабелей
- 4.4. Методы передачи
- 4.5. Технологии доступа к среде
 - CSMA
 - Коллизионные домены
- 4.6. Протоколы LAN
 - ARP
 - DHCP
 - ICMP

5. Протоколы маршрутизации

6. Сетевые устройства

- 6.1. Повторители
- 6.2. Мосты
- 6.3. Маршрутизаторы
- 6.4. Коммутаторы
 - Виртуальные сети (VLAN)
- 6.5. Шлюзы
- 6.6. Офисные автоматические телефонные станции (PBX)
- 6.7. Межсетевые экраны
 - Межсетевые экраны с фильтрацией пакетов
 - Межсетевые экраны с контролем состояния
 - Прокси
 - Динамическая фильтрация пакетов
 - Прокси уровня ядра
 - Архитектура межсетевых экранов

- 6.8. Хост-приманка (honeypot)
- 6.9. Разделение и изоляция сетей

7. Сетевые сервисы и Протоколы

- 7.1. Сетевые операционные системы
- 7.2. Служба доменных имен (DNS)
- 7.3. NIS
- 7.4. Службы каталогов
- 7.5. LDAP
- 7.6. Трансляция сетевых адресов (NAT)

8. Интрасети и Экстрасети

9. Городские вычислительные сети (MAN)

10. Глобальные вычислительные сети (WAN)

- 10.1. Эволюция телекоммуникаций
- 10.2. Выделенные линии
- 10.3. Технологии WAN
 - Frame Relay
 - Виртуальные каналы
 - X.25
 - ATM
 - Качество обслуживания (QoS)
 - SIP

11. Удаленный доступ

- 11.1. Dial-Up и RAS
- 11.2. ISDN
- 11.3. DSL
- 11.4. Кабельные модемы
- 11.5. VPN
 - Протоколы туннелирования
 - PPP
 - PPTP
 - L2TP
- 11.6. Протоколы аутентификации
- 11.7. Рекомендации по удаленному доступу

12. Беспроводные технологии

- 12.1. Беспроводные коммуникации

- 12.2. Компоненты WLAN
- 12.3. Беспроводные стандарты
- 12.4. WAP
- 12.5. i-Mode
- 12.6. Безопасность мобильных телефонов
- 12.7. Вардрайвинг
- 12.8. Спутники
- 12.9. Беспроводные коммуникации 3G

13. Руткиты

- 13.1. Шпионское и рекламное программное обеспечение
- 13.2. Передача мгновенных сообщений

14. Резюме

15. Тест

Домен 06. Криптография

1. История криптографии

2. Определения и концепции криптографии

- 2.1. Принцип Керкхофса
- 2.2. Стойкость криптосистем
- 2.3. Сервисы криптосистем
- 2.4. Одноразовый шифровальный блокнот
- 2.5. Динамические и скрытые шифры
- 2.6. Стеганография

3. Типы шифров

- 3.1. Шифры подстановки
- 3.2. Шифры перестановки

4. Методы шифрования

- 4.1. Симметричные и асимметричные алгоритмы
 - Симметричная криптография
 - Асимметричная криптография
- 4.2. Блочные и поточные шифры
 - Блочные шифры
 - Поточные шифры
 - Векторы инициализации
- 4.3. Гибридные методы шифрования
 - Сеансовые ключи

5. Типы симметричных систем

- 5.1. DES
- 5.2. 3DES
- 5.3. AES
- 5.4. IDEA
- 5.5. Blowfish
- 5.6. RC4
- 5.7. RC5
- 5.8. RC6

6. Типы асимметричных систем

- 6.1. Алгоритм Диффи-Хеллмана
- 6.2. RSA
 - Односторонние функции
- 6.3. Эль Гамаль
- 6.4. Криптосистемы на основе эллиптических кривых
- 6.5. LUC
- 6.6. Knapsack
- 6.7. Доказательство с нулевым разглашением

7. Целостность сообщения

- 7.1. Односторонний хэш
 - HMAC
 - CBC-MAC
- 7.2. Различные алгоритмы хэширования
 - MD2
 - MD4
 - MD5
 - SHA
 - HAVAL
 - Tiger
- 7.3. Атаки на односторонние функции хэширования
- 7.4. Цифровая подпись
- 7.5. Стандарт цифровой подписи

8. Инфраструктура открытых ключей

- 8.1. Центр сертификации
- 8.2. Сертификаты

- 8.3. Центр регистрации
- 8.4. Шаги PKI

9. Управление ключами

- 9.1. Принципы управления ключами
- 9.2. Правила использования ключей и управления ключами

10. Канальное и сквозное шифрование

11. Стандарты электронной почты

- 11.1. MIME
- 11.2. PEM
- 11.3. MSP
- 11.4. PGP
- 11.5. Квантовая криптография

12. Безопасность в сети Интернет

- 12.1. Начнем с основ
 - HTTP
 - HTTPS
 - S-HTTP
 - SET
 - Куки
 - SSH
 - IPSec

13. Атаки

- 13.1. Атака «Только шифротекст»
- 13.2. Атака «Известный открытый текст»
- 13.3. Атака «Выбранный открытый текст»
- 13.4. Атака «Выбранный шифротекст»
- 13.5. Дифференциальный криптоанализ
- 13.6. Линейный криптоанализ
- 13.7. Атаки с использованием побочных каналов
- 13.8. Атаки повтора
- 13.9. Алгебраические атаки
- 13.10. Аналитические атаки
- 13.11. Статистические атаки

14. Резюме

15. Тест

Домен 07. Непрерывность бизнеса и восстановление после аварий

1. Непрерывность бизнеса и восстановление после аварий

- 1.1. Шаги планирования непрерывности бизнеса
- 1.2. ВСП как часть Политики и Программы безопасности
- 1.3. Инициирование проекта

2. Требования к планированию непрерывности бизнеса

- 2.1. Анализ воздействия на бизнес
- 2.2. Превентивные меры
- 2.3. Стратегии восстановления
- 2.4. Восстановление бизнес-процессов
- 2.5. Восстановление здания
 - Соглашение о взаимной помощи
 - Резервные площадки
- 2.6. Восстановление технической среды
 - Резервирование оборудования
 - Резервирование программного обеспечения
 - Документация
 - Люди
- 2.7. Восстановление пользовательской среды
- 2.8. Варианты резервного копирования данных
- 2.9. Средства автоматизированного резервного копирования
- 2.10. Выбор здания для хранения резервной информации
- 2.11. Страхование
- 2.12. Восстановление и реконструкция
- 2.13. Разработка целей плана
- 2.14. Внедрение стратегий
- 2.15. Тестирование и пересмотр плана
- 2.16. Поддержка плана

3. Резюме

4. Тест

Домен 08. Законодательство, требования, соответствие, расследования

1. Многогранное киберправо

2. Проблемы киберправа

3. Сложности борьбы с киберпреступностью

- 3.1. Электронные активы
- 3.2. Эволюция атак
- 3.3. Трансграничные преступления
- 3.4. Типы права

4. Законодательство в области интеллектуальной собственности

- 4.1. Коммерческая тайна
- 4.2. Авторское право
- 4.3. Торговая марка
- 4.4. Патент
- 4.5. Внутренняя защита интеллектуальной собственности
- 4.6. Компьютерное пиратство

5. Неприкосновенность частной жизни

- 5.1. Законодательство и требования
 - Закон Сарбейнза-Оксли
 - Закон о преимущественности страхования и отчетности в области здравоохранения
 - Закон Грэма-Лича-Блилей
 - Закон о борьбе с компьютерным мошенничеством и злоупотреблениями
 - Закон о защите персональных данных
 - Базель II
 - PCI DSS
 - Закон о компьютерной безопасности
 - Закон об экономическом шпионаже
- 5.2. Вопросы неприкосновенности частной жизни сотрудников

6. Обязательства и последствия их нарушения

- 6.1. Персональные данные
- 6.2. Атака хакеров

7. Расследования

- 7.1. Реагирование на инциденты
- 7.2. Процедуры реагирования на инциденты

8. Компьютерная криминалистика и сбор доказательств

- 8.1. Международная организация по компьютерным доказательствам
- 8.2. Мотивы, возможности и средства
- 8.3. Поведение компьютерных преступников
- 8.4. Специалисты по расследованию инцидентов

- 8.5. Процесс проведения компьютерной экспертизы
- 8.6. Что является приемлемым для суда?
- 8.7. Наблюдение, обыск и изъятие
- 8.8. Проведение опросов и допросов
- 8.9. Несколько различных видов мошенничества

9. Этика

- 9.1. Институт компьютерной этики
- 9.2. Совет по архитектуре Интернета
- 9.3. Программа корпоративной этики

10. Резюме

11. Тест

Домен 09. Безопасность приложений

1. Важность программного обеспечения

2. Где нужно размещать безопасность?

3. Различные среды имеют различные потребности в обеспечении безопасности

4. Среда и приложения

5. Безопасность и функциональность

6. Типы, форматы и размер данных

7. Проблемы внедрения приложений и использования настроек «по умолчанию»

8. Сбои и ошибки в приложениях

9. Управление базами данных

- 9.1. Программное обеспечение для управления базами данных
- 9.2. Модели баз данных
- 9.3. Интерфейсы программирования баз данных
- 9.4. Компоненты реляционной базы данных
 - Словарь данных
 - Первичные и внешние ключи
- 9.5. Целостность
- 9.6. Вопросы безопасности баз данных
 - Представления базы данных
 - Многоэкземплядность
 - Обработка транзакций в режиме реального времени
- 9.7. Хранилища и интеллектуальный анализ данных

10. Разработка систем

- 10.1. Управление разработкой
- 10.2. Этапы жизненного цикла

- Инициирование проекта
- Управление рисками
- Анализ рисков
- Функциональное проектирование и планирование
- Техническое задание на разработку системы
- Разработка программного обеспечения
- Установка и внедрение
- Эксплуатация и сопровождение
- Удаление
- Виды тестирования
- Анализ завершенного проекта
- 10.3. Методы разработки программного обеспечения
- 10.4. Средства автоматизированной разработки программного обеспечения (CASE-средства)
- 10.5. Разработка прототипов
- 10.6. Методология безопасного проектирования
- 10.7. Методология безопасной разработки
- 10.8. Проверка на защищенность
- 10.9. Управление изменениями
- 10.10. Модель зрелости процессов разработки программного обеспечения (CMM)
- 10.11. Передача исходного кода программного обеспечения на хранение независимой третьей стороне

11. Методология разработки программного обеспечения

- 11.1. Концепции объектно-ориентированного программирования
- 11.2. Моделирование данных
- 11.3. Архитектура программного обеспечения
- 11.4. Структуры данных
- 11.5. Связность и связанность

12. Распределенные вычисления

- 12.1. CORBA и ORB
- 12.2. COM и DCOM
- 12.3. EJB
- 12.4. OLE
- 12.5. Распределенная вычислительная среда

13. Экспертные системы

14. Искусственные нейронные сети

15. Безопасность веб-приложений

- 15.1. Вандализм
- 15.2. Финансовое мошенничество
- 15.3. Привилегированный доступ
- 15.4. Кража информации о транзакциях
- 15.5. Кража интеллектуальной собственности
- 15.6. Атаки «отказ в обслуживании»
- 15.7. Организация процесса обеспечения качества
- 15.8. Межсетевые экраны для веб-приложений
- 15.9. Системы предотвращения вторжений
- 15.10. Реализация SYN-прокси на межсетевом экране
- 15.11. Специфические угрозы веб-среде
 - Сбор информации
 - Административные интерфейсы
 - Аутентификация и управление доступом
 - Управление конфигурациями
 - Проверка входных данных
 - Проверка параметров
 - Управление сессиями

16. Мобильный код

- 16.1. Java-апплеты
- 16.2. Элементы управления ActiveX
- 16.3. Вредоносное программное обеспечение
 - Вирусы
 - Черви
 - Троянские программы
 - Логические бомбы
 - Ботсети
- 16.4. Антивирусное программное обеспечение
- 16.5. Выявление спама
- 16.6. Противодействие вредоносному коду

17. Управление патчами

- 17.1. Методология управления патчами
- 17.2. Проблемы при установке патчей
- 17.3. Лучшие практики

- 17.4. Атаки
 - Отказ в обслуживании
 - Smurf
 - Fraggle
 - SYN-флуд
 - Teardrop
 - Распределенная атака отказ в обслуживании

18. Резюме

19. Тест

Домен 10. Операционная безопасность

1. Роль Департамента эксплуатации

2. Административное управление

- 2.1. Администратор безопасности и администратор сети
- 2.2. Подотчетность
- 2.3. Уровни отсечения

3. Уровень гарантий

4. Эксплуатационные обязанности

- 4.1. Необычные и необъяснимые события
- 4.2. Отклонения от стандартов
- 4.3. Внеплановая перезагрузка системы
- 4.4. Идентификация и управление активами
- 4.5. Системные защитные меры
- 4.6. Доверенное восстановление
- 4.7. Контроль входных и выходных данных
- 4.8. Укрепление систем
- 4.9. Безопасность удаленного доступа

5. Управление конфигурациями

- 5.1. Процесс управления изменениями
- 5.2. Документация по управлению изменениями

6. Контроль носителей информации

7. Утечки данных

8. Доступность сети и ресурсов

- 8.1. Среднее время безотказной работы (MTBF)
- 8.2. Среднее время восстановления (MTTR)
- 8.3. Единая точка отказа

- Устройства хранения с прямым доступом (DASD)
- RAID-массивы
- Массив с неактивными дисками (MAID)
- Избыточный массив независимых лент (RAIT)
- Сети хранения данных (SAN)
- Кластеризация
- Grid-вычисления
- 8.4. Резервное копирование
 - Иерархическое управление носителями
- 8.5. Планирование действий на случай непредвиденных ситуаций

9. Мейнфреймы

10. Безопасность электронной почты

- 10.1. Как работает электронная почта
 - POP
 - IMAP
 - Ретрансляция сообщений электронной почты
- 10.2. Безопасность факсов
- 10.3. Методы взлома и атак
 - Браузинг
 - Снифферы
 - Перехват коммуникационного сеанса
 - Loki
 - Взлом паролей
 - Бэкдоры

11. Тестирование уязвимостей

- 11.1. Тестирование на проникновение
- 11.2. Сканирование телефонных номеров
- 11.3. Другие виды уязвимостей
- 11.4. Что дальше?

12. Резюме

13. Тест

Домен 01. Информационная безопасность и управление рисками.

1. Управление безопасностью

Управление безопасностью включает в себя управление рисками, политики информационной безопасности, процедуры, стандарты, руководства, базисы, классификацию информации, организацию безопасности и обучение по вопросам безопасности. Эти ключевые аспекты служат основой корпоративной программы безопасности. Целью безопасности и программы безопасности является защита компании и ее активов. Анализ рисков позволяет идентифицировать эти активы, выявить угрозы, вызывающие риски для них, оценить возможные потери и потенциальные убытки, которые компания может понести в случае реализации любой из этих угроз. Результаты анализа рисков помогают руководству подготовить бюджет, учитывающий все необходимые затраты для защиты идентифицированных активов от выявленных угроз, и разрабатывать применимые на практике политики безопасности, которые направляют деятельность по обеспечению безопасности. Обучение и повышение осведомленности по вопросам безопасности позволяет довести необходимый объем информации до сведения всех и каждого сотрудников компании, что упрощает их работу и позволяет достичь целей безопасности.

Процесс управления безопасностью является непрерывным. Он начинается с оценки рисков и определения потребностей, затем следует мониторинг и оценка систем и применяемых методов работы. После этого проводится повышение осведомленности сотрудников компании, которое обеспечивает понимание вопросов, которые должны учитываться. Последним шагом является внедрение политик и защитных мер, направленных на снижение рисков и реализацию потребностей, определенных на первом шаге. Затем цикл начинается сначала. Таким образом, этот процесс постоянно анализирует и контролирует безопасность компании, позволяет ей адаптироваться и развиваться с учетом потребностей в обеспечении безопасности и тех условий, в которых компания существует и работает.

Управление безопасностью со временем меняется, так как меняется сетевое окружение, компьютеры и приложения, обрабатывающие информацию. Интернет, сети экстранет (сети бизнес-партнеров), сети интранет делают безопасность не только более сложной, но и более критичной. Ядро сетевой архитектуры изменилось с локализованных автономных вычислений на среду распределенных вычислений, что многократно увеличило ее сложность. Хотя доступ из внутренней сети в Интернет дает пользователям ряд важных возможностей и удобств, он увеличивает уязвимость компании из Интернета, что может стать источником дополнительных рисков безопасности.

Сегодня большинство организаций не смогут работать без компьютеров и их вычислительных возможностей. Многие крупные корпорации уже осознали, что их данные – это важнейший актив, который нужно защищать наравне со зданиями, оборудованием и другими физическими активами. Безопасность должна меняться одновременно с изменениями сетей и окружения. Безопасность – это больше, чем просто межсетевой экран и маршрутизатор со списком контроля доступа. Эти системы несомненно важны, но гораздо большее значение для безопасности имеет управление действиями пользователей и процедурами, которым они следуют. Это приводит нас к практике управления безопасностью, которая сосредоточена на постоянной защите активов компании.

1.1. Распределение обязанностей по управлению безопасностью

В мире безопасности, в функции руководителя входит определение целей, границ, политик, приоритетов и стратегий. Руководству нужно определить четкие границы и актуальные цели, достижение которых ожидается в результате выполнения программы безопасности. Также руководству нужно оценить цели бизнеса, риски безопасности, продуктивность пользователей, функциональные требования и цели. Наконец, руководство должно определить шаги, обеспечивающие правильное распределение и решение этих задач.

Многие компании смотрят на бизнес как на элементы уравнения и полагают, что обеспечение информационной и компьютерной безопасности входит в обязанности ИТ-администратора. Руководство таких компаний не воспринимает информационную и компьютерную безопасность всерьез, в результате чего безопасность в таких компаниях выглядит недоразвитой, плохо поддерживаемой, недостаточно финансируемой и неудачной. Безопасность должна учитываться на уровне высшего руководства. ИТ-администратор (или администратор безопасности) может консультировать руководство по вопросам безопасности, но безопасность компании не должна быть полностью делегирована ИТ-администратору (администратору безопасности).

Управление безопасностью основывается на четко идентифицированных и оцененных активах компании. После идентификации и оценки активов внедряются политики безопасности, процедуры, стандарты и руководства по обеспечению целостности, конфиденциальности и доступности для этих активов. Для классификации данных, выполнения анализа и оценки рисков используются различные инструменты. Эти инструменты помогают выявить уязвимости и показывают уровень их критичности, что позволяет внедрить эффективные контрмеры для снижения рисков наиболее оптимальным способом. В обязанности руководства входит обеспечение защиты ресурсов компании в целом. Этими ресурсами являются люди, капитал, оборудование и информация. Руководство должно принимать в этом участие, чтобы убедиться, что программа безопасности внедрена, угрозы, которые влияют на ресурсы компании, учтены, а также чтобы иметь уверенность в том, что необходимые защитные средства эффективны.

Должна быть обеспечена доступность необходимых ресурсов и финансирования, ответственные лица должны быть готовы принять участие в программе безопасности. Руководство должно распределить обязанности и определить роли, необходимые для начала выполнения программы безопасности, обеспечения ее успешного развития и эволюционирования по мере изменения окружения. Руководство также должно интегрировать программу безопасности в имеющуюся бизнес-среду и контролировать их работу. Поддержка руководства – одна из важнейших частей программы безопасности.

1.2. Подход "сверху-вниз"

В процессе планирования и внедрения программы безопасности специалист по безопасности должен определить выполняемые функции и ожидаемый конечный результат. Часто компании просто начинают блокировать компьютеры и устанавливать межсетевые экраны, не понимая требования безопасности в целом, цели и уровни доверия, которые они хотели бы получить от безопасности в рамках всего окружения. Группе, вовлеченной в данный процесс, следует начать сверху, с очень широких идей и терминов, и двигаться вниз к детальным конфигурациям и системным параметрам. На каждом этапе члены группы должны держать в уме основные цели безопасности, чтобы каждый новый компонент добавлял больше деталей к соответствующей цели.

Политика безопасности является своеобразным фундаментом для программы безопасности компании. К этой политике нужно относиться серьезно с самого начала, в нее должны быть заложены идеи постоянной актуализации, обеспечивающие постоянное функционирование всех компонентов безопасности и работу по достижению целей, соответствующих целям бизнеса.

Следующим шагом является разработка и внедрение процедур, стандартов и руководств, поддерживающих политику безопасности и определяющих контрмеры и методы, которые должны применяться для обеспечения безопасности. Когда эти элементы разработаны, программу безопасности следует детализировать, разработав базисы и конфигурации для выбранных средств и методов безопасности.

Если безопасность основана на прочном фундаменте и разработана с учетом целей и задач,

компании не придется вносить в нее существенные изменения. В этом случае процесс может быть более методичным, требующим меньше времени, денег и ресурсов, обеспечивая при этом правильный баланс между функциональностью и защитой. Это не является обязательным требованием, но понимание этого может сделать подход вашей компании к безопасности более управляемым. Вы можете объяснить компании каким образом следует планировать, внедрять и обеспечивать безопасность организованными способами, позволяющими избежать гигантской кучи средств безопасности, разрозненных и полных недостатков.

Для программы безопасности следует использовать **подход "сверху-вниз"**, означающий, что инициатива, поддержка и определение направления исходит от топ-менеджмента и идет через руководителей среднего звена к сотрудникам. Противоположный **подход "снизу-вверх"** относится к ситуации, когда ИТ-департамент пытается самостоятельно разработать программу безопасности, без должных указаний и поддержки руководства. Подход "снизу-вверх", как правило, менее эффективен, достаточно узок, и обречен на провал. Подход "сверху-вниз" гарантирует, что движущей силой программы являются люди (высшее руководство), которые действительно ответственны за защиту активов компании.

2. Администрирование безопасности и защитные меры

Если роль администратора безопасности отсутствует, руководство должно создать ее. Роль администратора безопасности напрямую отвечает за контроль основных аспектов программы безопасности. В зависимости от организации, ее размеров и потребностей в безопасности, администрированием безопасности может заниматься один сотрудник или группа сотрудников, работающих централизованно или децентрализованно. Независимо от размеров, администрирование безопасности требует четкой структуры отчетности, понимания обязанностей, а также возможностей проверки и мониторинга, чтобы убедиться в отсутствии нарушений безопасности, вызванных недостатками взаимодействия или понимания.

Владельцы информации должны указывать, какие пользователи могут иметь доступ к их ресурсам и что они могут делать с этими ресурсами. Задача администратора безопасности - убедиться, что этот процесс внедрен. Следующие защитные меры следует использовать для выполнения указаний руководства по вопросам безопасности:

- **Административные меры** включают в себя разработку и публикацию политик, стандартов, процедур и руководств, управление рисками, подбор персонала, проведение тренингов по вопросам безопасности, внедрение процедур управления изменениями.
- **Технические (логические) меры** включают внедрение и поддержку механизмов управления доступом, управления паролями и ресурсами, методами идентификации и аутентификации, устройствами безопасности, а также настройками инфраструктуры.
- **Физические меры** включают в себя контроль доступа людей в здание и различные помещения, использование замков и удаление неиспользуемых дисководов и приводов CD-ROM, защиту периметра здания, выявление вторжений, контроль окружения.

Рисунок 1-1 иллюстрирует совместную работу административных, технических и физических мер безопасности, обеспечивающую необходимый уровень защиты.

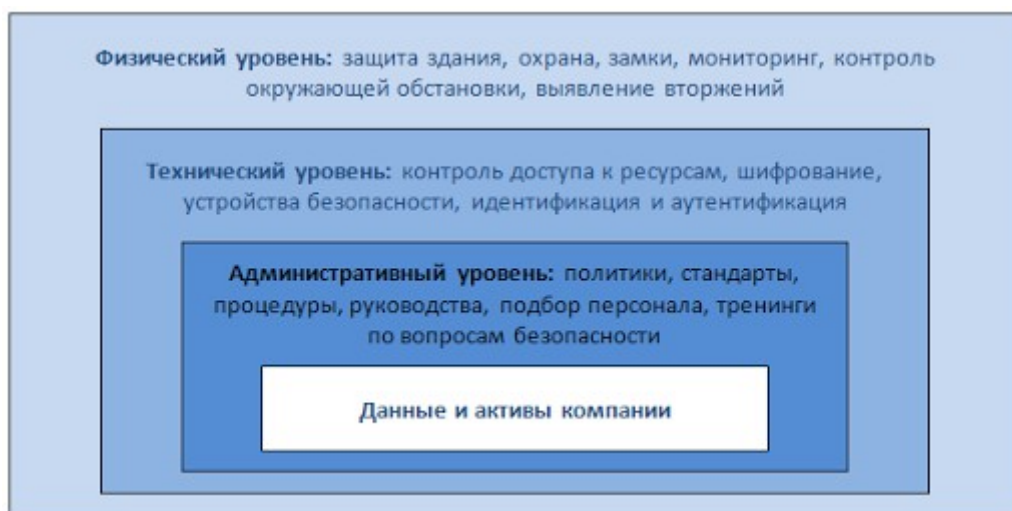


Рисунок 1-1 Административный, технический и физический уровни защитных мер должны работать совместно для защиты активов компании

Владельцем информации обычно является ответственный сотрудник, входящий в руководящий состав компании или руководитель соответствующего департамента. Владелец информации обязан обеспечить надлежащую защиту данных, он несет единоличную ответственность за любую халатность в отношении защиты информационных активов компании. Сотрудник, который выполняет эту роль, несет ответственность за классификацию информации, он указывает, как эта информация должна быть защищена. Если защита данных не основана на требованиях владельца информации, если он не контролирует выполнение своих требований, может быть нарушена концепция *due care* (должной заботы).

Следует обеспечить постоянное взаимодействие между группой администраторов безопасности и высшим руководством, чтобы гарантировать, что программа безопасности получает достаточную поддержку, а руководство принимает необходимые решения по ее реализации. Часто высшее руководство полностью исключает свое участие в вопросах безопасности, не принимая во внимание, что в случае возникновения серьезных инцидентов, связанных с безопасностью, именно высшее руководство будет объяснять их причины бизнес-партнерам, акционерам и публике. После такого случая отношение коренным образом изменяется, руководство максимально включается в вопросы безопасности. Следует обеспечить процесс постоянного взаимодействия между группой администраторов безопасности и высшим руководством, обеспечивающий двусторонние взаимоотношения.

Неадекватное руководство может свести на нет все усилия компании в области безопасности. Возможными причинами неадекватного руководства может быть недостаточное понимание руководством потребностей компании в обеспечении безопасности, конкуренция безопасности с другими целями руководства, взгляд руководства на безопасность как на дорогую и ненужную затею, поддержка безопасности руководством компании только на словах. Мощные и полезные технологии, устройства, программное обеспечение, процедуры и методология обеспечивают определенный уровень безопасности, но без полноценного управления безопасностью и поддержки руководства они не имеют никакого значения.

2.1. Основные принципы безопасности

Существует несколько маленьких и больших задач программы безопасности, но 3 основных принципа есть во всех программах: доступность, целостность и конфиденциальность. Это называется **AIC-триадой** (Availability, Integrity, Confidentiality). Уровень безопасности, необходимый для реализации этих принципов, отличается в различных компаниях, так как

каждая компания имеет собственное уникальное сочетание целей бизнеса и безопасности, а также потребностей. Все защитные меры и механизмы безопасности внедряются для реализации одного (или нескольких) из этих принципов, а все риски, угрозы и уязвимости измеряются по их потенциальной способности нарушения одного или всех принципов АИС. АИС-триада показана на Рисунке 1-2.

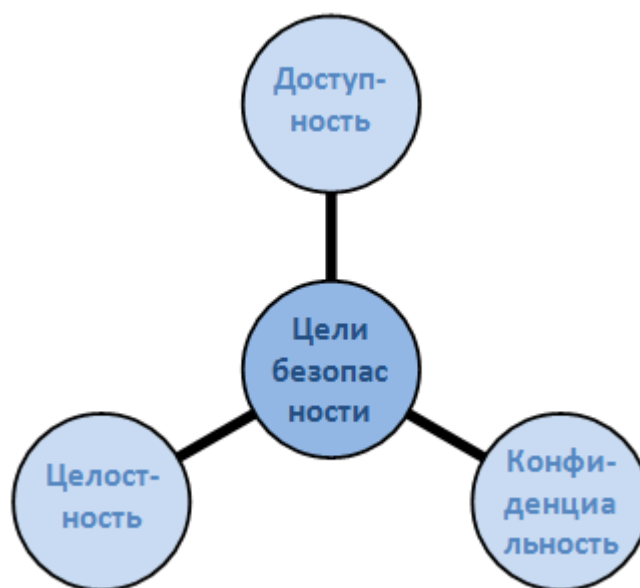


Рисунок 1-2 АИС-триада

Доступность

Системы и сети должны обеспечивать достаточный уровень предсказуемости в сочетании с приемлемым уровнем производительности. Они должны иметь возможность восстанавливаться после сбоев быстро и безопасно, чтобы это не оказывало негативного воздействия на производительность работы компании. Следует избегать "единых точек отказа", осуществлять резервное копирование, при необходимости обеспечивать определенный уровень избыточности, предотвращать негативное влияние со стороны внешней среды. Необходимо внедрить механизмы защиты от внутренних и внешних угроз, которые могут сказаться на доступности и производительности сети, систем и информации. **Доступность** обеспечивает уполномоченным лицам надежный и своевременный доступ к данным и ресурсам.

На доступность системы может повлиять сбой аппаратного или программного обеспечения. Следует использовать резервное оборудование для возможности оперативной замены критически важных систем. Обслуживающий персонал должен обладать всеми необходимыми знаниями и быть доступен для своевременного перехода на резервные системы и выполнения соответствующих настроек. Внешние факторы, такие как температура, влажность, статическое электричество, пыль могут также повлиять на доступность системы. Эти вопросы подробно рассматриваются в Домене 04.

DoS-атаки являются популярной методикой хакеров, нарушающей работу компании. Такие атаки снижают возможности доступа пользователей к ресурсам систем и информации. Чтобы защититься от них, следует ограничивать количество доступных портов, использовать системы IDS, контролировать сетевой трафик и работу компьютеров. Правильная настройка межсетевых экранов и маршрутизаторов также может уменьшить угрозу DoS-атак.

Целостность

Целостность обеспечивает гарантии точности и надежности информации и предоставляющих ее информационных систем, предотвращает возможность несанкционированных изменений. Аппаратные средства, программное обеспечение и

коммуникационное оборудование должны работать совместно для надлежащего хранения и обработки данных, их правильного перемещения до места назначения в неизменном виде. Системы и сети должны быть защищены от вмешательства извне.

Атаки на системы или ошибки пользователей не должны влиять на целостность систем и данных. Если злоумышленник установит вирус, логическую бомбу или скрытый вход (backdoor), целостность системы будет нарушена. Это может негативно повлиять на целостность информации, хранящейся в системе, и привести к мошенничеству, несанкционированным изменениям программного обеспечения и данных. Для борьбы с этими угрозами необходим строгий контроль доступа, системы выявления вторжений.

Пользователи, как правило, влияют на целостность систем или данных в результате ошибок (хотя внутренние пользователи также могут совершать мошеннические или злоумышленные действия). Например, случайное удаление конфигурационных файлов, ввод ошибочной суммы операции и т.д.

Меры безопасности должны ограничить возможности пользователей только минимально необходимым набором функций, что снизит вероятность и последствия их ошибок. Доступ к критичным системным файлам должен быть ограничен для пользователей. В приложениях следует предусмотреть механизмы контроля входящей информации, проверяющие ее корректность и адекватность. Права изменения данных в базах данных должны быть предоставлены только уполномоченным лицам, передаваемые по каналам связи данные должны быть защищены с помощью шифрования или других механизмов.

Конфиденциальность

Конфиденциальность обеспечивает необходимый уровень секретности в каждой точке обработки данных и предотвращает их несанкционированное раскрытие.

Конфиденциальность должна обеспечиваться как при хранении информации, так и в процессе ее передачи.

Атакующие могут нарушить конфиденциальность, перехватывая сетевой трафик, подглядывая за работой сотрудников, похищая файлы с паролями, применяя методы социальной инженерии. Пользователи могут преднамеренно или случайно разглашать конфиденциальную информацию, забывая зашифровать ее перед отправкой другому лицу, став жертвой атаки с использованием социальной инженерии, предоставляя доступ к секретной информации компании, не обеспечивая необходимой защиты при обработке конфиденциальной информации.

Конфиденциальность может быть обеспечена путем шифрования данных при их хранении и передаче, применения строгой системы контроля доступа, классификации данных, а также обучения персонала правильным методам работы с конфиденциальной информацией.

2.2. Определения безопасности

Важно понимать значение слов "уязвимость", "угроза", "риск", "воздействие", а также взаимосвязь между ними.

Уязвимость - это недостаток в программном обеспечении, оборудовании или процедуре, который может предоставить атакующему возможность доступа к компьютеру или сети и получения несанкционированного доступа к информационным ресурсам компании. Уязвимость - это отсутствие или слабость защитных мер. Уязвимостью может являться служба, запущенная на сервере, "непропатченное" приложение или операционная система, неограниченный вход через модемный пул, открытый порт на межсетевом экране, слабая физическая безопасность, позволяющая любому войти в серверную комнату, отсутствие управления паролями на серверах и рабочих станциях.

Угроза - это потенциальная опасность для информации или системы. Угрозой является, если кто-то или что-то выявит наличие определенной уязвимости и использует ее против

компании или человека. Нечто, дающее возможность использования уязвимости, называется **источником угрозы** (threat agent). Источником угрозы может быть хакер, получивший доступ к сети через открытый на межсетевом экране порт; процесс, осуществляющий доступ к данным способом, нарушающим политику безопасности; торнадо, разрушившее здание; сотрудник, совершивший ошибку, которая может привести к утечке конфиденциальной информации или нарушению целостности файлов.

Риск - это вероятность того, что источник угрозы воспользуется уязвимостью, что приведет к негативному воздействию на бизнес. Если межсетевой экран имеет несколько открытых портов, существует высокая вероятность, что злоумышленник воспользуется одним из них для несанкционированного доступа к сети. Если пользователи не обучены правильным процессам и процедурам, существует высокая вероятность совершения ими умышленных и неумышленных ошибок, которые могут привести к уничтожению данных. Если в сети не внедрена система IDS, существует высокая вероятность того, что факт проведенной атаки останется не выявленным, пока уже не будет слишком поздно.

Воздействие (exposure) - это нечто, приводящее к потерям в связи с действиями источника угрозы. Уязвимости воздействуют на компанию, приводя к возможности нанесения ей ущерба. Если управление паролями слабое, а требования к паролям не внедрены, компания подвержена возможному воздействию в результате компрометации паролей пользователей и их использования для несанкционированного доступа. Если компания не следит за своей электропроводкой и не предпринимает шагов для предотвращения пожара, она подвержена потенциальному воздействию пожара.

Контрмеры (или **защитные меры**) - это меры, внедрение которых позволяет снизить уровень потенциального риска. Контрмерами может быть настройка программного обеспечения, оборудования или процедур, устраняющая уязвимости или снижающая вероятность того, что источник угрозы сможет воспользоваться уязвимостью. Примером контрмер является строгое управление паролями, охрана, механизмы контроля доступа операционных систем, установка паролей BIOS, проведение обучения пользователей по вопросам безопасности.

Если компания использует антивирусное программное обеспечение, но не обновляет базы вирусных сигнатур, это уязвимость. Компания уязвима для вирусных атак. Угрозой является то, что вирус проникнет в сеть компании и парализует ее работу. Риск в данном случае - это вероятность проникновения вируса в сеть компании и нанесения ей ущерба. Если вирус проникнет в сеть компании, уязвимость будет использована и компания окажется под воздействием нанесенного им ущерба. Контрмерами в этой ситуации будет установка антивирусного программного обеспечения на все компьютеры компании и поддержка актуальности их баз вирусных сигнатур. Взаимосвязь между рисками, уязвимостями, угрозами и контрмерами показана на Рисунке 1-3.



Рисунок 1-3 Взаимосвязь между различными компонентами безопасности

Ссылки по теме:

- NIST Computer Security Resource Center
- CISSP and SSCP Open Study Guides
- CISSP.com

2.3. Безопасность посредством неизвестности

Неправильное понимание рисков может привести к множеству различных проблем для компании и не позволит обеспечить хорошую работу безопасности. К сожалению, достаточно часто применяется такая практика, как "безопасность посредством неизвестности", которая ведет к плачевным результатам. Корень этой проблемы заключается в отсутствии понимания возможностей современных компьютерных злоумышленников, незнании инструментов, которые они имеют в своем распоряжении, а также недооценке их изобретательности. Это ведет защитников информации к серьезнейшей ошибке – они считают себя умнее своего потенциального противника. Следствием этого являются элементарные ошибки в защите, небрежности и распространение ложного чувства безопасности. Примерами таких распространенных ошибочных мнений могут быть следующие: *уязвимости не могут быть использованы, если они не общеизвестны; скомпилированный код более безопасен, чем открытый исходный код; перевод HTTP-трафика на порт 8088 обеспечит достаточную защиту; алгоритмы шифрования собственной разработки остановят злоумышленника* и т.п. Это лишь немногие варианты потенциально опасных мнений, возникающих вследствие использования подхода к безопасности посредством неизвестности.

Хотя всем хочется верить во врожденную доброту и порядочность своих коллег, если бы это на самом деле было бы так, у специалистов по безопасности не было бы работы. Хороший подход в безопасности иллюстрируется старой поговоркой: *"Есть только два человека в мире, которым я верю – ты и я... но я не уверен в отношении тебя"*. Лучше занимать именно такую позицию, так как безопасность действительно может быть нарушена кем угодно и в любое время.

Еще один хороший пример безопасности посредством неизвестности – это ключ от квартиры, которые многие кладут под коврик, когда уходят из дома. Они полагают, что никто не знает об этом ключе, и считают, что это безопасно. При этом любой человек может легко получить доступ в квартиру, если он найдет этот ключ, а опытные злоумышленники (в данном примере, воры) прекрасно знают такие привычки и в первую очередь будут искать именно их следствия. Безопасность не может строиться на основе неизвестности!

В мире криптографии, аналогичные идеи воплощены в принципе Кирхгофа, который еще в 1880-х годах, заявил о том, что нет смысла хранить в тайне алгоритм, т.к. злоумышленник может узнать (или предположить) его. Единственное, что должно быть тайной – это ключ.

3. Организационная модель безопасности

Организационная модель безопасности является структурой, состоящей из многих элементов, механизмов защиты, логических, административных и физических компонентов, процедур, бизнес-процессов и конфигураций, которые работают совместно, обеспечивая необходимый уровень безопасности окружения. Каждая модель имеет свои отличия, но все модели реализованы в виде слоев: каждый слой поддерживает вышестоящий слой и защищает нижестоящий слой. Поскольку модель безопасности является структурой, компании могут наполнять ее различными видами технологий, методов и процедур для достижения необходимого уровня защиты своего окружения. Рисунок 1-4 иллюстрирует компоненты, из которых может состоять модель безопасности.



Рисунок 1-4. Комплексная и эффективная модель безопасности имеет множество отдельных компонентов

Эффективная безопасность требует взвешенного подхода и применения всех компонентов и процедур безопасности. Некоторые компоненты безопасности являются техническими (списки контроля доступа, шифрование), а некоторые - не техническими (физическими и административными, такими, как разработка политики безопасности и обеспечение соответствия ей), но каждый имеет важное место в рамках общей модели. Если один компонент отсутствует или реализуется не в полной мере, это может оказать негативное воздействие на всю структуру.

Модель безопасности имеет различные слои и различные виды целей, которые должны быть достигнуты за различные промежутки времени. Цели могут быть ежедневными (*операционными*), среднесрочными (*тактическими*) и долгосрочными (*стратегическими*). То же самое происходит и в сфере планирования безопасности. Ежедневные (операционные) цели связаны с продуктивностью и выполнением текущих задач, обеспечивающих функционирование компании предсказуемым образом. Среднесрочной (тактической) целью является, например, объединение всех рабочих станций и ресурсов в один домен, чтобы обеспечить возможность централизованного контроля. Примером долгосрочных (стратегических) целей может являться перевод всех филиалов на связь с головным офисом посредством VPN-соединений, объединение всех беспроводных технологий с целью единого подхода к обеспечению их безопасности.

Стратегическое планирование работает с планами, которые находятся на одном уровне с бизнес-целями и целями ИТ. Цели стратегического планирования долгосрочны и имеют широкий горизонт. Стратегическое планирование может включать некоторые из следующих целей:

- Обеспечить правильное понимание и учет рисков
- Обеспечить соответствие требованиям законодательства и регуляторов
- Интегрировать обязанности по безопасности в деятельность компании
- Создать модель зрелости для обеспечения постоянного улучшения
- Использовать безопасность как бизнес-преимущество, чтобы привлечь больше клиентов

Тактическое планирование относится к деятельности и поддержке, которые необходимы для достижения широких целей, выдвинутых в процессе стратегического планирования. В общем случае, тактические планы имеют более короткие сроки и более узкий горизонт планирования по сравнению со стратегическими планами.

И, наконец, оперативное планирование – это весьма конкретные планы, сроки и цели. Оперативное планирование предполагает указание конкретных мероприятий, установление жестких сроков и графиков выполнения плана. Это конкретные действия, которые нужно предпринять для достижения целей тактических и стратегических планов. Ниже приводятся несколько примеров оперативного планирования:

- Выполнения оценки рисков безопасности
- Недопущение негативного влияния изменений в системе безопасности на продуктивность
- Поддержка и внедрение защитных мер
- Постоянное сканирование уязвимостей и установка программных обновлений
- Контроль соответствия политикам

Такой подход к планированию называется *горизонтом планирования* (planning horizon). Безопасность работает лучше всего, если оперативные, тактические и стратегические цели компании определены и работают поддерживая друг друга.

3.1. Компоненты программы безопасности

В настоящее время компании, корпорации, государственные учреждения и частные лица значительно больше внимания уделяют вопросам информационной безопасности, чем когда-либо прежде. Специалисты по безопасности приветствуют это, поскольку это означает большую степень вовлеченности в вопросы безопасности тех, кто принимает решения в компании. Ведь технологии являются лишь небольшой частью общей организационной безопасности компании. Однако наиболее часто события в компаниях развиваются по следующему сценарию.

Генеральный директор и совет директоров вынуждены обратить внимание на вопросы информационной безопасности, так как появляются новые требования законодательства, существенно возрастает ущерб от вирусных атак, против компании подаются судебные иски за нарушение защиты информации. Компания обычно нанимает консультанта, который объясняет генеральному директору и совету директоров, что они нуждаются в политике безопасности и оценке информационных активов. Компания платит за проведение этих работ и верит в то, что теперь она защищена. Однако это ложное чувство, поскольку в компании до сих пор нет программы безопасности.

Затем компания нанимает специалиста по безопасности (обычно называемого CSO - Corporate Security Officer (руководитель Службы безопасности компании) или CISO - Corporate Information Security Officer (руководитель Службы информационной безопасности компании)) и делегирует ему всю работу по обеспечению безопасности и ответственность за нее, но не дает при этом ему каких-либо реальных полномочий и бюджета. Потом, когда инциденты с безопасностью все же случаются, всю ответственность за это возлагают на CISO (CSO).

Таким образом, специалисты по безопасности имеют 3 возможных варианта действий:

- Спрятать голову в песок, и надеяться, что проблемы обойдут стороной компанию в целом и этого специалиста в частности
- Продолжать работать в том же духе, виня судьбу за все разочарования
- Понимая, что наше общество делает только первые шаги в развитии информационной безопасности, изучать и использовать лучшие практики, уже разработанные в данной отрасли

CISO обязан хорошо понимать бизнес-процессы и цели компании, доводить до сведения высшего руководства информацию о рисках, которые угрожают компании, а также о требованиях законодательства и регуляторов, которым должна соответствовать компания. Он должен разработать и внедрить программу обучения (повышения осведомленности) сотрудников компании по вопросам безопасности. Другими задачами CISO является разработка документов по безопасности: политик, процедур, базисов, стандартов и руководств. CISO должен быть в курсе новых технологий, отслеживать новую информацию, касающуюся вопросов безопасности. Также, в задачи CISO входит оценка реакции на инциденты, подготовка программ обеспечения соответствия компании новым требованиям, разработка метрик безопасности. Выполняя все эти обязанности и требования, CISO будет работать эффективно и обеспечит уверенность компании в надлежащей работе безопасности и учете рисков.

Важно, чтобы вопросы безопасности обсуждались и указывались в отчетах на максимально высоком уровне управления компанией, так как негативное воздействие на бизнес проблем безопасности и несоответствия требованиям может быть катастрофическим. Отчитываясь перед CEO (Chief Executive Officer – генеральный директор или президент компании) и другими руководителями высшего звена, CISO должен предоставить достоверную информацию и исключить любое недопонимание. Помимо высшего руководства CISO должен предоставлять отчеты и информацию в департамент ИТ, административный департамент, департамент страхования и управления рисками, юридический департамент, департамент внутреннего аудита, а также в бизнес-подразделения.

3.2. Стандарты безопасности

CobiT (Control Objectives for Information and related Technology) – это набор стандартов и лучших практик, разработанный ISACA (Information Systems Audit and Control Association) и ITGI (IT Governance Institute). CobiT определяет цели контроля ИТ, которые следует использовать для надлежащего управления ИТ и обеспечения соответствия информационных технологий компании потребностям ее бизнеса. CobiT состоит из четырех доменов: Планирование и Организация, Приобретение и Внедрение, Эксплуатация и Сопровождение, Мониторинг и Оценка. Каждый домен делится на подкатегории. Таким образом, домены CobiT предоставляют компаниям цели и инструкции, применимые при покупке, установке, тестировании, сертификации и аккредитации ИТ-продуктов. CobiT очень полезен, т.к. большинство компаний используют неформальные и непродуманные подходы при покупке ИТ-продуктов и выполнении процедур. Многие требования, а также аудиты основываются на стандарте CobiT. Поэтому, если вы хотите сделать своих аудиторов счастливыми, изучайте, используйте в работе, внедряйте контрольные объекты, которые

считаются лучшими практиками.

Людей, которые впервые видят CobiT, он просто ошеломляет, так как он имеет очень большой объем и не поддается полномасштабному внедрению даже за пару лет. По каждой из категорий CobiT определяет цели и методы контроля, целевые показатели, факторы успеха, а также модель зрелости. В нем излагается подробный план, которому можно следовать для выполнения каждой из 34 предусмотренных в нем целей контроля.

Рисунок 1-5 показывает, как структура CobiT объединяет бизнес требования, ИТ-ресурсы и ИТ-процессы. Многие аудиторы информационной безопасности используют CobiT в качестве критериев оценки результативности применяемых защитных мер. Поэтому, если вы хотите успешно пройти аудит, компании было бы неплохо знать и осмысленно выполнять указанные в CobiT задачи управления.

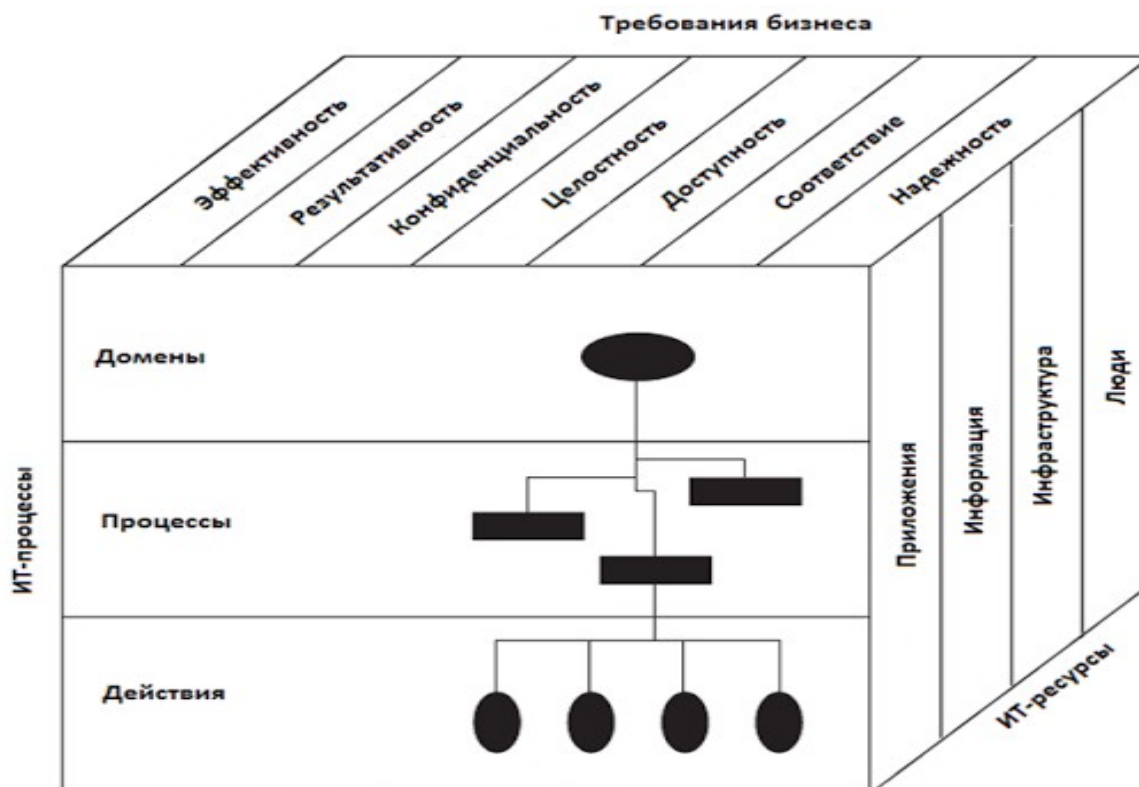


Рисунок 1-5. Компоненты CobiT

CobiT основан на стандарте COSO, разработанном в 1985 году (разработчик: Committee of Sponsoring Organizations of the Treadway Commission) для борьбы с мошеннической финансовой деятельностью и недостоверной отчетностью. Структура COSO состоит из следующих компонентов:

- Управление средой
 - Философия управления и стиль работы
 - Культура компании, как отношение к этике и мошенничеству
- Оценка риска
 - Определение целей в области рисков
 - Возможность управлять внутренними и внешними изменениями
- Деятельность по контролю
 - Политики, процедуры и практика, применяемая для снижения риска

- Информация и коммуникации
 - Структура, обеспечивающая что только уполномоченные лица получают достоверную информацию в то время, когда она им необходима
- Мониторинг
 - Выявление и реакция на недостатки контроля

COSO – это модель корпоративного управления, а CobiT – модель управления ИТ. COSO в большей степени относится к стратегическому уровню, а CobiT больше сосредоточен на оперативном уровне.

Разработка и внедрение программы безопасности не так сложны, как кажутся многим компаниям, однако это новые для них вещи, которые представляются страшными и запутанными. Поэтому им следует обратиться к отраслевым стандартам и лучшим практикам, которые дают конкретные рекомендации по созданию и реализации полноценной программы безопасности.

Наиболее широко для этих целей используется стандарт ISO 17799, основой которого является Британский Стандарт BS 7799. Это признанный на международном уровне стандарт управления информационной безопасностью, который предоставляет высокоуровневые концептуальные рекомендации по обеспечению безопасности компании. BS 7799 состоит из двух частей: в первой части описываются цели управления и ряд средств управления, которые могут быть использованы для достижения этих целей, а во второй части приводится порядок внедрения и поддержки программы безопасности. Вторая часть BS 7799 является базисом, на соответствие которому может быть сертифицирована компания. Сертификация компании может проводиться, например, с целью повышения доверия к ней со стороны клиентов и партнеров, а также с целью использования факта сертификации в качестве инструмента маркетинга. Сертификацию проводит уполномоченная независимая третья сторона, при этом компания может быть сертифицирована на соответствие всему ISO 17799 Part II, либо только отдельным его частям.

В настоящее время мы перешли от стандарта ISO 17799 к целой группе стандартов ISO, которые помогут вам понять и структурировать эти лучшие практики. ISO использует различные номера серий для различных видов стандартов. Например, серия ISO 9000 содержит ряд стандартов, которые относятся к управлению качеством для бизнес-процессов. Новая серия ISO/IEC 27000 используется для стандартов безопасности и гарантий. ISO подкорректировал стандарты 17799 для их соответствия новому формату нумерации. Ниже представлены стандарты ISO/IEC, которые следует использовать компаниям при разработке своей программы безопасности:

- **ISO/IEC 27001.** Основан на стандарте BS7799 Part II, связанном с организацией, внедрением, контролем и совершенствованием Системы управления информационной безопасностью.
- **ISO/IEC 27002.** Свод правил по управлению защитой информации (предыдущее название – ISO 17799), основан на стандарте BS 7799 Part I.
- **ISO/IEC 27004.** Стандарт для измерений в области управления информационной безопасностью.
- **ISO/IEC 27005.** Предназначен для реализации надлежащей информационной безопасности на основе ориентированного на риски подхода.
- **ISO/IEC 27006.** Руководство по процессу сертификации / регистрации.
- **ISO/IEC 27799.** Руководство по защите персональных данных о здоровье.

Домены стандарта ISO/IEC 27002 (который ранее назывался ISO 17799), приведенные ниже,

очень близки CISSP CBK (Common Body of Knowledge):

- **Политика информационной безопасности компании.** Карта бизнес-целей в отношении безопасности, поддержки со стороны руководства, целей безопасности и обязанностей.
- **Создание инфраструктуры информационной безопасности.** Создание и поддержка организационной структуры безопасности посредством распределения обязанностей по безопасности, процессов авторизации, учета требований безопасности при аутсорсинге, независимого контроля обеспечения информационной безопасности.
- **Классификация и управление активами.** Разработка инфраструктуры безопасности для защиты активов компании посредством учета, инвентаризации, классификации, процедур использования активов.
- **Безопасность, связанная с персоналом.** Снижение рисков, вызванных "человеческим фактором" за счет тщательного отбора персонала, определения ролей и обязанностей, надлежащего обучения сотрудников, документирования мер воздействия в случае несоблюдения требований.
- **Физическая безопасность и безопасность окружения.** Защита активов компании посредством правильного размещения здания компании, установления и поддержания периметра безопасности, внедрения контроля доступа, защиты оборудования.
- **Управление коммуникациями и функционированием.** Обеспечение безопасности функционирования посредством операционных процедур, надлежащего управления изменениями, разделения обязанностей, планирования ресурсов, управления сетью, работы с носителями информации, мониторинга.
- **Контроль доступа.** Контроль доступа к активам на основе требований бизнеса, управления пользователями, методов аутентификации.
- **Разработка и поддержка систем.** Обеспечение безопасности на всех этапах жизненного цикла систем посредством разработки требований безопасности, криптографии, контроля целостности и процедур разработки программного обеспечения.
- **Управление инцидентами безопасности.** Обеспечение своевременного получения сведений о произошедших инцидентах и возникших уязвимостях, разработка порядка управления инцидентами и их анализа.
- **Управление непрерывностью бизнеса.** Противодействие нарушению нормального функционирования посредством планирования непрерывности и тестирования планов.
- **Соответствие требованиям.** Обеспечение соответствия требованиям регуляторов, законодательства, условиям договоров и другим предписанным требованиям, посредством технических средств контроля и аудита.

CobiT и COSO говорят о том, что должно быть достигнуто, но не говорят как. На помощь здесь приходят ITIL и серия ISO/IEC 27000. ITIL (Information Technology Infrastructure Library) фактически является стандартом оптимального управления ИТ-службой. ITIL был создан для удовлетворения потребностей бизнеса, в связи с его растущей зависимостью от ИТ. К сожалению, слишком часто в компаниях существует целая пропасть между персоналом бизнес-подразделений и ИТ-подразделений, поскольку они используют различную терминологию и имеют разные цели в компании. Отсутствие понимания между бизнесом и ИТ ведет к неправильному (неэффективному) сочетанию целей бизнеса и функций ИТ, что, в свою очередь, ведет к путанице, нарушению сроков, упущенным

возможностям, увеличению затрат времени и сил, а также разочарованиям с обеих сторон. Для решения этой проблемы предназначен ITIL. Там, где CobiT определяет ИТ-цели, ITIL указывает шаги на уровне процессов, предназначенные для достижения этих целей. Хотя в ITIL есть раздел, связанный с безопасностью, его внимание в основном сконцентрировано на внутренних соглашениях об уровне обслуживания (SLA – Service Level Agreement) между ИТ-департаментом и другими подразделениями компании, которые он обслуживает.

Ссылки по теме:

- ISO
- The ISO 17799 Community Portal
- ISACA CobiT Framework
- IT Infrastructure Library (ITIL)

3.3. Управление безопасностью на стратегическом уровне

Стратегическое управление безопасностью - очень похоже по своему характеру на корпоративное управление и управление ИТ на том же уровне, поскольку существуют дублирующие функции и задачи во всех трех. Все три вида управления работают в рамках организационной структуры компании, и имеют одни и те же цели – обеспечение выживания и процветания компании, просто каждый вид управления фокусируется на своей части. Количество требований, которым должны удовлетворять компании, постоянно растет. Советы Директоров компаний несут все больше и больше ответственности за работу бизнеса и эффективность всей компании. В связи с этим, более важную роль начинает играть стратегическое управление безопасностью и защитой информации, что обеспечивает использование надлежащих механизмов и предоставление Совету Директоров (и руководству компании) возможности эффективного надзора с целью управления рисками, поддержания их на приемлемом для компании уровне и ограничения потенциального ущерба.

Существует множество различных определений стратегического управления безопасностью. Вот вариант определения, подготовленного ITGI: *"Стратегическое управление (governance) – это набор функций, выполняющихся Советом Директоров и высшим руководством, которые задают стратегическое направление, контролируют достижение целей, надлежащее управление рисками и ответственное использование ресурсов компании"*.

Это абсолютно правильное определение, но оно очень высокоуровневое и его трудно понять. Чтобы упростить понимание, давайте сравним две компании. Компания "А" внедрила эффективную программу стратегического управления безопасностью, а компания "Б" нет. Обе компании имеют политику безопасности, процедуры, стандарты, одинаковые технологии управления безопасностью (межсетевые экраны, системы выявления вторжений, системы управления идентификацией и т.д.), подразделение безопасности, которой руководит CISO. Может показаться, что уровень безопасности компаний "А" и "Б" одинаков... Но это не так. Посмотрите на некоторые критические различия, приведенные в таблице 1-1.

Компания "А"	Компания "Б"
Члены Совета Директоров понимают, что информационная безопасность имеет решающее значение для компании, и требуют ежеквартально предоставлять сведения об эффективности безопасности и выявленным недостаткам.	Члены Правления не понимают, что информационная безопасность находится в их сфере ответственности, и сосредотачиваются исключительно на корпоративном управлении и прибыли.
CEO, CFO, CIO и руководители бизнес-подразделений участвуют в работе Комитета по управлению рисками, который собирается каждый месяц, тема информационной безопасности всегда является одной из тем повестки дня.	CEO, CFO и руководители бизнес-подразделений считают, что информационная безопасность находится в сфере ответственности CIO, CISO и ИТ-департамента. И поэтому не вмешиваются.
Высшее руководство устанавливает приемлемый уровень риска, что является основой для политики безопасности компании и всей деятельности в области безопасности.	CISO использует шаблонные политики безопасности, включая в них название компании и подпись CEO.
Высшее руководство делегировало руководителям бизнес-подразделений обязанности по управлению рисками, относящимися непосредственно к их подразделениям.	Вся деятельность, касающаяся безопасности, осуществляется департаментом безопасности. Таким образом, безопасность работает в бункере и не интегрирована в работу компании.
Критические бизнес-процессы документированы с учетом рисков, которые им присущи на различных шагах.	Бизнес-процессы не документированы, не проанализированы на наличие потенциальных рисков, которые могут повлиять на операции, производительность и рентабельность.
Сотрудники несут ответственность за любые нарушения безопасности, произошедшие по их вине умышленно или случайно.	Политики и стандарты разработаны, но не исполняются или не осуществляется контроль, так как он не был предусмотрен или внедрен.
Средства безопасности и различные услуги покупаются и внедряются эффективными способами. Их применение постоянно анализируется, чтобы убедиться, что они остаются экономически целесообразными.	Средства безопасности и различные услуги покупаются и внедряются без каких-либо исследований, подготовки показателей эффективности, позволяющих определить отдачу от инвестиций и эффективность.
Компания постоянно пересматривает свои процессы, включая вопросы обеспечения безопасности, с целью их постоянного совершенствования.	Компания не анализирует свою деятельность с целью ее улучшения, но постоянно движется вперед, совершая одни и те же ошибки снова и снова.

Таблица 1-1. Сравнение компании "А" и компании "Б"

Большинство компаний на сегодняшний день имеют множество частей программы безопасности (политики, стандарты, межсетевые экраны, подразделение безопасности, системы выявления вторжений и т.д.), но руководство в обеспечении безопасности не участвует, а безопасность не интегрирована в деятельность компании. За обеспечение безопасности компании отвечает исключительно подразделение безопасности, что является невозможным практически. Безопасность в таких компаниях является вопросом техники. Но сегодня в мире информационной безопасности такой подход не работает. Сегодняшняя безопасность – это намного большее, чем чисто техническое решение. Специалисты по безопасности должны понимать, что безопасность должна соблюдаться в рамках всей компании, и крайне важно иметь несколько центров ответственности и подотчетности. Стратегическое управление безопасностью является целостной системой, состоящей из различных компонентов (технических средств, персонала, процессов, политик и т.д.), которые существуют для выживания и процветания компании.

ПРИМЕЧАНИЕ. Следует также учитывать такой важнейший фактор, как корпоративная культура компании. Даже если компания использует самые современные и передовые решения на рынке, она не сможет обеспечить необходимый уровень безопасности, если эти решения используются необученным, апатичным и беззаботным персоналом. Оценка культуры компании, очень важна при оценке положения безопасности в ней.

Для того, чтобы обеспечить управление безопасностью, должно быть нечто такое, чем можно управлять. Набор объектов управления, которыми должна обладать компания, в общем виде называется программой безопасности.

Разработка программы безопасности

Важно понимать, что программа безопасности имеет непрерывный жизненный цикл, т.к. она должна постоянно оцениваться и совершенствоваться. Для описания жизненного цикла программы безопасности, будем использовать следующие шаги:

1. Планирование и Организация
2. Реализация (внедрение)
3. Функционирование и Поддержка

4. Мониторинг и Оценка

Однако многие компании не применяют подход жизненного цикла при разработке, внедрении и поддержании своей программы управления безопасностью. Они не знают, как это делать, или считают, что это слишком сложно и бесполезно. В результате это, как правило, приводит к следующим последствиям:

- Написанные политики и процедуры, не отражаются на деятельности по обеспечению безопасности и не поддерживают ее.
- Отсутствует взаимодействие и координация между различными сотрудниками компании, задействованными в обеспечении защиты ее активов.
- Не проводится оценка результатов, отдачи от инвестиций и распределения ресурсов.
- Отсутствует понимание недостатков программы безопасности, не применяются единообразные способы исправления недостатков.
- Нет гарантий соответствия требованиям законодательства, регуляторов, требованиям политик.
- Компания полностью полагается на технологии для решения всех вопросов безопасности.
- Отсутствует единый орган принятия решений в компании.
- К любым нарушениям применяется подход "пожарной тревоги", а не спокойный проактивный, детективный подход.
- Появляется ложное чувство безопасности.

Также, отсутствие жизненного цикла программы безопасности и управления безопасностью приводит к тому, что безопасность рассматривается просто как еще одна задача. А все задачи имеют дату начала и дату завершения, которая означает, что все участники переключаются на другую задачу. В результате компания с течением времени выполняет одни и те же задачи, на них расходуются большие средства, но результат при этом минимален.

Основные элементы каждой фазы жизненного цикла программы безопасности представлены ниже:

- **Планирование и Организация**

- Получение одобрения от руководства.
- Создание руководящего комитета по надзору (oversight steering committee).
- Оценка бизнес-драйверов (бизнес-драйверы – это люди, информация или задачи, которые обеспечивают реализацию бизнес-целей компании).
- Создание профиля угроз компании.
- Проведение оценки рисков.
- Разработка архитектуры безопасности на организационном, прикладном, сетевом и компонентном уровнях.
- Определение решений на каждом уровне архитектуры.
- Получение согласия руководства на дальнейшие действия.

- **Реализация (внедрение)**

- Распределение ролей и обязанностей
- Разработка и внедрение политик безопасности, процедур, стандартов, базисов и

руководств

- Выявление критичных данных на этапах хранения и передачи
- Реализация следующих **проектов**:
 - Идентификация и управление активами
 - Управление рисками
 - Управление уязвимостями
 - Соответствие требованиям
 - Управление идентификацией и доступом
 - Управление изменениями
 - Жизненный цикл разработки программного обеспечения
 - Планирование непрерывности бизнеса
 - Обучение и повышение осведомленности
 - Физическая безопасность
 - Реакция на инциденты
- Внедрение решений (административных, технических, физических) по каждому проекту
- Разработка решений по аудиту и мониторингу для каждого проекта
- Установка целей, соглашений об уровне обслуживания (SLA) и метрик по каждому проекту
- **Эксплуатация и Сопровождение**
 - Соблюдение установленных процедур для обеспечения соблюдения базисных уровней в каждом реализованном проекте.
 - Проведение внутренних и внешних аудитов.
 - Выполнение задач, намеченных в каждом проекте.
 - Управление соглашениями об уровне обслуживания по каждому проекту.
- **Мониторинг и Оценка**
 - Анализ лог-файлов, результатов аудита, собранных значений метрик и SLA по каждому проекту
 - Оценка достижения целей по каждому проекту
 - Проведение ежеквартальных встреч с руководящими комитетами
 - Совершенствование действий каждого этапа и их интеграция в фазу Планирования и Организации

ПРИМЕЧАНИЕ. Различные компании, консультанты и специалисты по безопасности могут использовать различные подходы к созданию программы безопасности, но в общем они охватывают те же разделы. Каждая компания имеет различные приемлемые для нее уровни риска, внедренные защитные меры, угрозы и бизнес-драйверы, однако их программы безопасности в основном содержат похожие элементы – просто одна компания больше внимания уделяет одним элементам, а другая – другим. Это связано с деятельностью компаний и их потребностях в безопасности.

Модели и структуры весьма полезны, но они находятся на очень высоком уровне. Для

воплощения их в жизнь необходимо разработать соответствующие **проекты** (blueprint). Проекты должны соответствовать требованиям безопасности компании, основанным на обязательствах компании, бизнес-драйверах и внутренних требованиях. Например, компания "В" имеет политику конфиденциальности, подразделение безопасности компании разработало стандарты и процедуры, реализующие стратегию обеспечения конфиденциальности, которым все сотрудники компании должны следовать. Затем разрабатывается проект, в проектной документации к которому указывается больше деталей, учитываются процессы и компоненты, к которым относятся требования политики, стандартов и процедур. В проектной документации должно быть, как минимум, следующее:

- Схема сети компании
- Места в сети, где находятся критичные данные
- Сегменты сети, через которые проходят критичные данные
- Различные применяемые решения по безопасности (VPN, SSL, PGP), которые защищают критичные данные
- Подключения третьих сторон, с которыми совместно используются критичные данные
- Используемые меры безопасности в отношении подключений третьих сторон
- И многое другое ...

Проекты должны соответствовать потребностям компании. Если компания "В" использует систему управления идентификацией (identity management), у нее должен быть соответствующий проект, проектная документация к которому содержит описание ролей, управления регистрацией, источников авторизации, хранения идентификационных данных, применения решений единого входа (single sign-on) и т.д. Если компания "В" не использует систему управления идентификацией, то ей не нужен проект для этой системы. Ряд проектов, которые следует разработать большинству компаний, приведен ниже:

- Управление безопасностью
- Непрерывность бизнеса
- Журналирование и мониторинг
- Управление идентификацией
- Целостность приложений
- Инфраструктура
- Управление активами
- Физическая безопасность и безопасность окружения
- И многое другое ...

Таким образом, проект должен учитывать решения безопасности, процессы и компоненты, которые использует компания на основании своих потребностей в безопасности. Эти проекты имеют отношение к различным бизнес-подразделениям компании. Например, использование системы управления идентификацией предусматривает участие каждого подразделения компании. Четкое выполнение компанией этих проектов позволяет обеспечить стандартизацию, упростить сбор метрик и управление. Проекты следует разрабатывать с учетом лучших практик (как правило, с учетом ISO 17799). На рисунке 1-6 показано, где эти проекты вступают в игру при разработке программы безопасности.

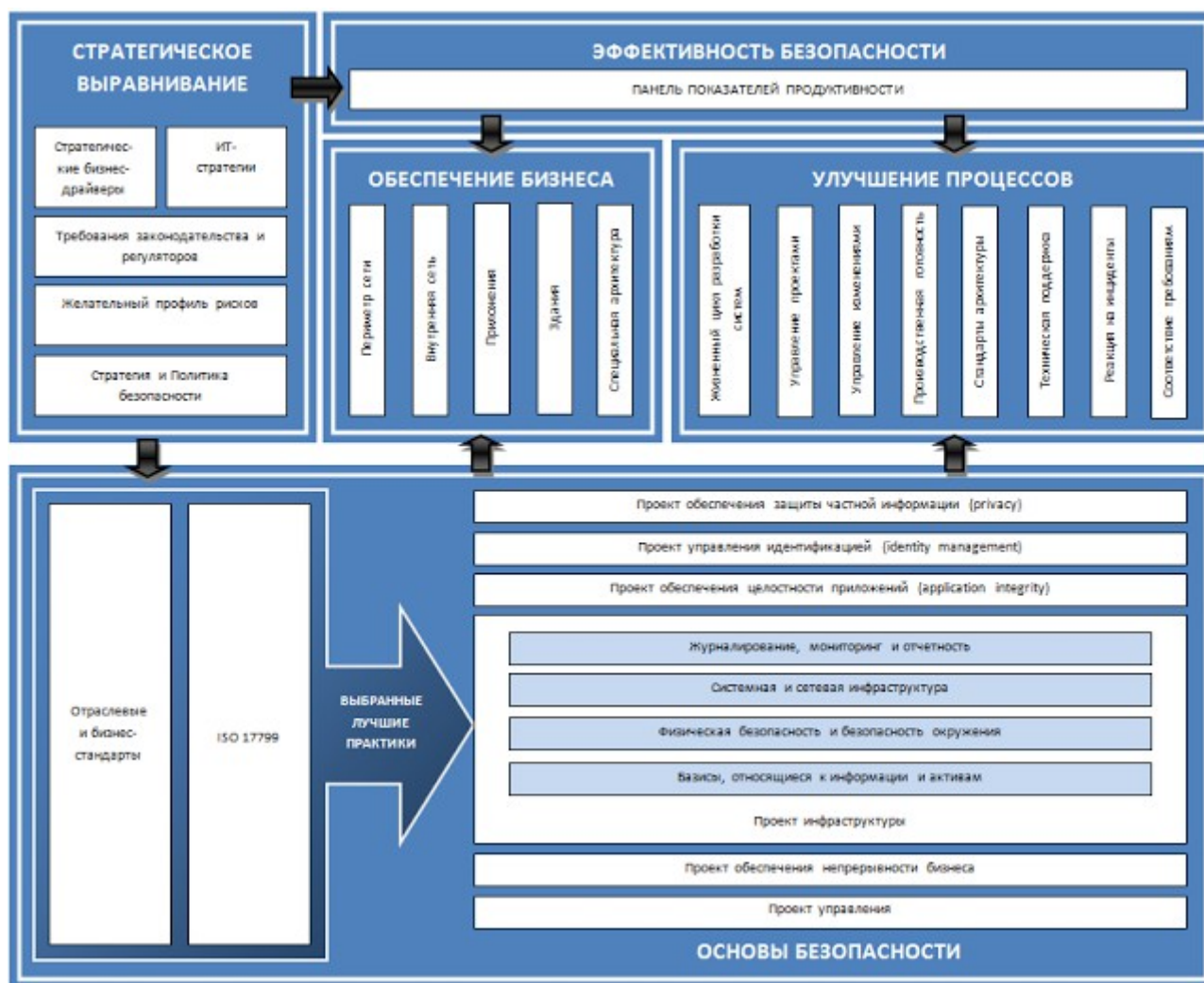


Рисунок 1-6. Проекты должны быть отображением требований бизнеса и безопасности

4. Управление информационными рисками

Риск - это вероятность причинения ущерба и возможные последствия. **Управление информационными рисками (IRM - Information Risk Management)** представляет собой процесс выявления и оценки рисков, снижения их до приемлемого уровня, а также внедрения адекватных механизмов для поддержания этого уровня. Стопроцентной защиты не существует. Каждая система имеет уязвимости и подвержена угрозам. Необходимо выявлять эти угрозы, оценивать вероятность их реализации и ущерб, к которому это может привести. Затем нужно предпринимать правильные шаги для снижения общего уровня риска всего окружения до уровня, считающегося в компании приемлемым.

Риски могут иметь различные формы, не обязательно связанные с компьютерами. С точки зрения информационной безопасности, существует несколько видов рисков, которые компания должна понимать и учитывать. Ниже представлен список основных категорий:

- **Физический ущерб.** Пожар, затопление, вандализм, отсутствие электроэнергии, стихийные бедствия.
- **Действия человека.** Случайные или намеренные действия или бездействие, которые могут нарушить работу компании.
- **Неисправность оборудования.** Неработоспособность систем или периферийных устройств.
- **Внутренние и внешние атаки.** Хакинг, крекинг и проведение атак.

- **Неправильное использование данных.** Предоставление совместного доступа к конфиденциальной информации компании, мошенничество, шпионаж, кражи.
- **Утрата данных.** Преднамеренное или непреднамеренное уничтожение информации.
- **Ошибки приложений.** Ошибки в вычислениях, ошибки ввода информации, переполнения буфера.

Эти угрозы должны быть выявлены, категорированы, и оценены с точки зрения масштабов потенциальных потерь. Реальные риски очень трудно измерить, однако расставить приоритеты в списке потенциальных рисков и выбрать те, которыми следует заняться в первую очередь, вполне достижимо.

4.1. Кто действительно разбирается в управлении рисками?

В действительности людей, которые понимают как управлять рисками, очень мало (в том числе и за рамками вопросов безопасности). В информационной безопасности часто концентрируются на приложениях, устройствах, протоколах, вирусах и т.д. Эти детали безусловно должны быть учтены в процессе управления рисками, но они не должны быть в центре внимания при управлении рисками. Управление рисками в целом значительно важнее отдельных технических мер.

Безопасность - задача бизнеса. Однако бизнес работает, чтобы зарабатывать деньги, а не только чтобы быть безопасным. Бизнес часто обращает внимание на безопасность только тогда, когда появляются реальные угрозы основной деятельности - потеря репутации и клиентов из-за утечки информации, ущерб в результате вирусной атаки и т.д. Специалисты по информационной безопасности должны понимать эти угрозы, но гораздо важнее, чтобы они понимали как можно рассчитать риски этих угроз и предоставить карту рисков руководителям бизнеса.

Чаще всего бюджет является ограниченным, а количество уязвимостей - нет. В этом случае необходимо сконцентрироваться на наиболее критичных уязвимостях, устранение которых даст реальную отдачу для бизнеса.

4.2. Политика управления информационными рисками

Полноценное управление рисками требует полноценной поддержки высшего руководства, документированного процесса поддержки миссии организации, IRM-политики и IRM-группы.

IRM-политика должна быть частью общей политики управления рисками компании. IRM-политика должна быть отражена и в организационной политике безопасности компании. В IRM-политике должны учитываться следующие аспекты:

- Цели IRM-группы
- Уровень риска, который компания приняла для себя как приемлемый
- Формальные процессы выявления рисков
- Связь между IRM-политикой и процессами стратегического планирования компании
- Обязанности, связанные с IRM, и роли, необходимые для их выполнения
- Связь между рисками и внутренним контролем
- Подходы к изменению поведения сотрудников и распределения ресурсов с учетом анализа рисков
- Связь между рисками и целевыми показателями и бюджетами
- Ключевые показатели для мониторинга эффективности защитных мер

IRM-политика предоставляет инфраструктуру для процессов и процедур управления рисками компании. Она должна охватывать все вопросы информационной безопасности, начиная от подбора персонала и инсайдерских угроз, заканчивая физической безопасностью и межсетевыми экранами. Она должна предоставлять механизм передачи информации о рисках от IRM-группы высшему руководству, а также механизм принятия решений о необходимости снижения рисков высшим руководством.

4.3. Группа управления рисками (IRM-группа)

В зависимости от размеров компании и бюджета безопасности компания может иметь одного или нескольких сотрудников, ответственных за IRM (IRM-группу). Основная цель IRM-группы - обеспечить защиту компании наиболее выгодным (экономически) и эффективным способом. Эта цель может быть достигнута только при наличии следующих компонентов:

- Установленный высшим руководством приемлемый уровень риска
- Документированные процессы и процедуры оценки рисков
- Процедуры выявления и снижения рисков
- Адекватные ресурсы и бюджет, предоставленные высшим руководством
- Планы действий при возникновении непредвиденных обстоятельств (для тех областей, оценка которых указывает на необходимость таких планов)
- Тренинги по вопросам безопасности для всех сотрудников, использующих информационные активы
- Возможность расширения (развития) группы в отдельных областях, в случае необходимости
- Учет и выполнение требований законодательства и регуляторов
- Разработка метрик и показателей эффективности, позволяющих измерить и управлять различными видами рисков
- Возможность выявления и оценки новых рисков при изменениях в компании или окружении
- Интеграция IRM и процессов управления изменениями компании, чтобы изменения не приводили к появлению новых уязвимостей

В большинстве случаев, в IRM-группу включают не отдельных, специально нанятых, сотрудников, а тех, которые уже работают в компании и выполняют другие задачи. В таких случаях совершенно необходима поддержка высшего руководства для надлежащего распределения ресурсов.

Как и в любой другой команде, IRM-группе нужен лидер. Он должен заниматься только этим вопросом (либо, в крупных компаниях, он должен уделять этому 50-70% рабочего времени). Руководство должно выделить этому человеку адекватный бюджет, в случае необходимости обеспечить профессиональную подготовку, наличие инструментов для успешного анализа рисков.

5. Анализ рисков

Анализ рисков, который на самом деле представляет собой инструмент для управления рисками, является методом выявления уязвимостей и угроз, оценки возможного воздействия, что позволяет выбирать адекватные защитные меры именно для тех систем и процессов, в которых они необходимы. Анализ рисков позволяет сделать безопасность экономически эффективной, актуальной, своевременной и способной реагировать на угрозы. Он также

помогает компании приоритезировать список рисков, определить и обосновать разумную стоимость защитных мер.

Анализ рисков имеет четыре основные цели:

- Идентификация активов и их ценности для компании
- Идентификация угроз и уязвимостей
- Количественная оценка вероятности и влияния на бизнес этих потенциальных угроз
- Обеспечение экономического баланса между ущербом от воздействия угроз и стоимостью контрмер

Анализ рисков позволяет сравнить годовую стоимость защитных мер с потенциальным ущербом. Годовая стоимость защитных мер не должна превышать потенциальный годовой ущерб. Также, анализ рисков позволяет связать программу безопасности с целями и требованиями бизнеса компании, что крайне важно для успеха и в том, и в другом.

Перед началом работы по выявлению и анализу рисков важно понять цель данной работы, ее объем и ожидаемый результат. Следует учитывать, что попытка проанализировать все риски во всех областях за один раз может оказаться невыполнимой.

Одной из первых задач группы анализа рисков является подготовка детального отчета по стоимости активов. Высшее руководство должно проанализировать этот отчет и определить сферу деятельности для IRM-проекта (объем работы), исключив из него те активы, которые не важны на данном этапе. При определении объема работ следует также учитывать бюджет проекта, а также требования законодательства. В ходе обсуждений с руководством, все участники должны иметь ясное представление о ценности обеспечения АИС-триады (доступность, целостность и конфиденциальность) и ее непосредственной связи с потребностями бизнеса.

Анализ рисков должен осуществляться при поддержке и управлении со стороны высшего руководства. Только в этом случае он будет успешным. Руководство должно определить цели и масштабы анализа, назначить членов группы для проведения оценки, а также выделить необходимое время и средства для проведения этой работы. Крайне важно, чтобы высшее руководство внимательно отнеслось к результатам проведенной оценки.

5.1. Группа анализа рисков

Для наиболее эффективного анализа рисков, компания должна включить в состав группы анализа рисков сотрудников большинства (или всех) своих подразделений, что необходимо для выявления и учета всех рисков. Членами группы могут быть руководители подразделений, разработчики приложений, ИТ-персонал - любые ключевые сотрудники ключевых подразделений компании. Это совершенно необходимо, т.к. группа, состоящая только из ИТ-специалистов, не сможет выявить множество рисков (например, риски, связанные с работой бухгалтерии). Желательно в состав группы включать руководителей подразделений, а не рядовых сотрудников, которые могут не представлять себе работу всего подразделения в целом и, соответственно, не могут выявить все угрозы. Для этого целесообразно установить соответствующий минимальный уровень должности для члена группы.

Если по какой-либо причине компания не может включить в группу сотрудников из различных подразделений, необходимо, как минимум, организовать проведение интервью с ключевыми сотрудниками каждого подразделения.

При анализе рисков следует задавать следующие вопросы: Что случится при реализации угрозы? Какими могут быть потенциальные последствия? Как часто это может происходить? Какой уровень достоверности ответов на первые три вопроса? Большинство такой

информации собирается в ходе внутренних исследований, интервью, собраний рабочих групп.

Владельцы рисков

Один из наиболее важных вопросов – кто в компании владеет рисками? Ответить на него непросто, т.к. это зависит от ситуации и от того, о каких рисках идет речь. Высшее руководство владеет рисками, связанными с процессом функционирования компании, но оно может переложить их на ответственных за хранение данных или бизнес-подразделения для проведения определенных работ, и на это время они должны выполнять отдельные обязанности владельцев рисков. Конечно, риски в конечном итоге всегда остаются у высшего руководства и оно должно быть уверено, что делегированная работа выполняется понятными методами, в процессе нее учитываются существующие риски и предпринимаются действия по их минимизации.

5.2. Ценность информации и активов

Ценность информации определяется трудоемкостью ее подготовки (сбора), стоимостью ее поддержки (сопровождения), величиной возможного ущерба в случае ее потери или уничтожения, стоимостью, которую другие лица (конкуренты, злоумышленники) готовы заплатить за нее, а также величиной возможных последствий и штрафов, в случае ее утраты (утечки). Без проведения оценки информации невозможно адекватно оценить целесообразность затрат денег и ресурсов на ее защиту. Ценность информации обязательно должна учитываться при выборе защитных мер.

5.3. Определение стоимости и ценности

Оценка актива может проводиться как количественными, так и качественными методами. Фактическая стоимость актива определяется на основании стоимости его приобретения, разработки и поддержки. Ценность актива определяется его значением, которое он имеет для владельцев, уполномоченных и неуполномоченных пользователей. Некоторая информация является важной для компании и ей присваивается гриф конфиденциальности.

Например, стоимость сервера составляет \$4000, но это не является его ценностью, учитываемой при оценке рисков. Ценность определяется затратами на его замену или ремонт, потерями из-за снижения производительности, ущербом от повреждения или утраты хранящихся на нем данных. Именно это будет определять ущерб для компании в случае повреждения или утраты сервера по той или иной причине.

Следующие вопросы должны быть учтены при определении ценности активов:

- Затраты на получение или разработку актива
- Затраты на поддержку и защиту актива
- Ценность актива для владельцев и пользователей
- Ценность актива для злоумышленников (конкурентов)
- Ценность интеллектуальной собственности, использованной при разработке актива
- Цена, которую другие готовы заплатить за актив
- Затраты на замену актива при утрате
- Операционная и производственная деятельность, которая зависит от доступности актива
- Ответственность в случае компрометации актива
- Польза и роль актива в компании

Понимание ценности актива является первым шагом к пониманию того, какие средства и механизмы безопасности должны использоваться для его защиты. Ценность актива определяет стоимость защитных мер, которые следует использовать для его защиты.

Определение стоимости активов полезно для компании по целому ряду причин, включая следующие:

- Для проведения анализа затраты/выгоды (cost/benefit analyse)
- Для выбора конкретных контрмер и защитных средств
- Для определения необходимого уровня страхового покрытия
- Для понимания, чем именно рискует компания
- Для выполнения требований законодательства, регуляторов, соблюдения должной заботы (due care)

Активы могут быть материальным (компьютеры, оборудование, материалы) или нематериальными (репутация, данные, интеллектуальная собственность). Обычно трудно оценить количественно ценность нематериальных активов (например, репутации), которая может меняться с течением времени.

5.4. Идентификация угроз

Как было сказано ранее, риск - это вероятность того, что источник угрозы воспользуется уязвимостью, что приведет к негативному воздействию на бизнес. Существует множество видов источников угрозы, которые могут использовать разные типы уязвимостей, что может привести к определенным угрозам. Некоторые примеры рисков показаны в таблице 1-2.

Источник угрозы	Может использовать эту уязвимость	В результате возникнет угроза
Вirus	Отсутствие антивирусного программного обеспечения	Заражение вирусом
Хакер	Большое количество служб, запущенных на сервере	Несанкционированный доступ к конфиденциальной информации
Пользователи	Неверно настроенный параметр операционной системы	Неисправность системы
Пожар	Отсутствие огнетушителей	Здание и компьютеры повреждены, возможны человеческие жертвы
Сотрудник	Отсутствие обучения или требований Отсутствие контроля	Общий доступ к критичной информации Внесение изменений во вводимую информацию и выводимую из приложений обработки данных
Подрядчик	Слабые механизмы контроля доступа	Утечка конфиденциальной информации
Атакующий	Плохо написанное приложение Нестрогие настройки межсетевых экранов	Проведение атаки "переполнение буфера" Проведение DoS-атаки
Нарушитель	Отсутствие охраны	Похищение компьютеров и других устройств

Таблица 1-2 Взаимосвязь угроз и уязвимостей

Существуют и другие, гораздо более сложные для выявления виды угроз, которые могут произойти в компьютерной среде. Эти угрозы связаны с ошибками в приложениях и ошибками пользователей. Однако, при надлежащей организации контроля и аудита действий пользователей их ошибки (умышленные или случайные) выявить существенно проще.

После выявления уязвимостей и связанных с ними угроз должны быть проанализированы последствия их использования, т.е. риски потенциального ущерба. Ущерб может быть связан с повреждением данных или систем (объектов), несанкционированным разглашением конфиденциальной информации, снижением производительности работы и т.д. При проведении анализа рисков, группа должна также рассмотреть вероятный *отложенный ущерб* (delayed loss), который может произойти по прошествии некоторого времени (от 15

минут до нескольких лет) после реализации риска. Отложенный ущерб может быть вызван, например, снижением производительности работы через определенный период времени, снижением дохода компании, ущербом ее репутации, накопительными штрафами, дополнительными расходами на восстановление окружения, приостановкой приема средств от клиентов и т. д.

Например, если в результате атаки на веб-серверы компании они перестали обслуживать клиентов, непосредственным ущербом может быть повреждение данных, затраты рабочего времени на восстановление работы серверов, обновление уязвимого программного обеспечения на них. Кроме того, компания потеряет определенный доход в следствие невозможности обслуживания клиентов в течение времени, которое потребуется ей на восстановление работы своих веб-серверов. Если восстановительные работы займут значительное время (например, неделю), компания может потерять настолько большой объем прибыли, что будет уже не в состоянии оплачивать счета и другие расходы. Это и будет являться отложенным ущербом. А если кроме всего прочего компания еще и потеряет доверие клиентов, она может полностью потерять свой бизнес (на некоторое время или навсегда). Это является крайним случаем отложенного ущерба.

Такого рода вопросы существенно усложняют количественную оценку ущерба, но они обязательно должны быть приняты во внимание для получения достоверной оценки.

Методики оценки рисков. Для оценки рисков используется множество различных методик. Давайте рассмотрим некоторые из них.

NIST SP 800-30 и 800-66 являются методологиями, которые могут использоваться коммерческими компаниями, хотя 800-66 изначально разрабатывалась для здравоохранения и других регулируемых отраслей. Подход NIST учитывает угрозы ИТ и соответствующие риски информационной безопасности. Он предусматривает следующие шаги:

- Описание характеристик системы
- Идентификация угроз
- Идентификация уязвимостей
- Анализ защитных мер
- Определение вероятности
- Анализ воздействия
- Определение риска
- Рекомендации защитных мер
- Документирование результатов

Методология оценки рисков NIST SP 800-30 часто используется консультантами и специалистами по безопасности, внутренними ИТ-подразделениями. Она ориентирована в основном на компьютерные системы. Отдельные люди или небольшие группы собирают данные из сети, из используемых практик по безопасности, а также от людей, работающих в компании. Собранные данные используются в качестве исходных данных для выполнения шагов анализа рисков, описанных в документе 800-30.

Другой методологией оценки рисков является **FRAP** (Facilitated Risk Analysis Process – Групповой процесс анализа рисков). Она создана для проведения качественной оценки рисков способом, который позволяет вести проверки по различным аспектам и с использованием различной методологии. Она предоставляет способы, позволяющие компании принимать решения о направлении действий и конкретных действиях в конкретных обстоятельствах для учета различных проблем. Это дает возможность посредством предварительного отбора определить те области в компании, которые действительно нуждаются в анализе рисков. FRAP построен таким образом, что любой человек с хорошими навыками организации групповой работы сможет успешно провести анализ рисков по этой методике.

Еще одним видом методологии является **OCTAVE** (Operationally Critical Threat, Asset, and Vulnerability Evaluation – Оценка критичных угроз, активов и уязвимостей). Это методология, которую следует применять в ситуациях, когда весь процесс анализа рисков информационной

безопасности проводится силами сотрудников компании (без привлечения внешних консультантов). Она основана на идее, что сотрудники компании лучше всех понимают, что действительно нужно компании, и перед лицом каких рисков она стоит. Выбранные для участия в процессе оценки сотрудники сами решают, какой подход будет наилучшим для оценки безопасности их компании.

В то время, как методологии NIST и OCTAVE направлены на угрозы ИТ и риски информационной безопасности, *AS/NZS 4360* использует гораздо более широкий подход к управлению рисками. Эта методология может использоваться для понимания рисков компании в области финансов, защиты людей, принятия бизнес-решений и т.д. Она не была разработана специально для анализа рисков безопасности, хотя может успешно использоваться и для этой цели.

ПРИМЕЧАНИЕ. Вы можете найти дополнительную информацию по этим подходам к анализу рисков и их использованию в статье Шон Харрис на странице http://searchsecurity.techtarget.com/generic/0,295582,sid14_gci1191926,00.html.

CRAMM (CCTA Risk Analysis and Management Method – Метод анализа и управления рисками Центрального агентства по компьютерам и телекоммуникациям (CCTA) Великобритании) разделен на три сегмента: идентификация и оценка активов, анализ угроз и уязвимостей, выбор контрмер. Эта методика учитывает как технические аспекты компании, так и нетехнические.

Анализ связующего дерева (Spanning Tree Analysis) – это методология, которая создает дерево всех потенциальных угроз и недостатков, которые могут нарушить работу системы. Каждая ветвь является обобщенной темой или категорией, неприменимые ветви могут удаляться в процессе проведения анализа рисков.

5.5. Анализ сбоев и дефектов

FMEA (Failure Modes and Effect Analysis) – это метод определения функций, выявления функциональных дефектов, оценки причин дефекта и его последствий с помощью структурированного процесса. Применение этого процесса в случае постоянных дефектов позволяет определить место, где ошибка, скорее всего, произойдет. Это очень помогает выявить уязвимые места, точно определить границы уязвимостей и последствия их эксплуатации. В свою очередь, это позволяет не только упростить применение исправлений, устраняющих уязвимости, но и обеспечить более эффективное использование ресурсов в рамках этой задачи.

Следуя определенной последовательности шагов, можно достичь наилучших результатов в анализе дефектов.

1. Начните с блок-схемы системы или контроля (объекта анализа).
2. Рассмотрите, что произойдет, если каждый блок диаграммы даст сбой.
3. Нарисуйте таблицу, и укажите в ней дефекты в паре с их последствиями и оценкой этих последствий.
4. Корректируйте проект системы и вносите соответствующие изменения в таблицу до тех пор, пока не станет ясно, что система не подвержена проблемам.
5. Получите несколько инженерных обзоров характера дефектов и анализа последствий.

В таблице 1-3 приведен пример проведения и документирования FMEA. Хотя большинство компаний не имеют ресурсов для столь детальной проработки каждой системы и контроля, она должна проводиться для критичных функций и систем, которые могут оказать существенное влияние на компанию. Очень важно проанализировать защитную меру или систему от микро- до макро-уровня, чтобы в полной мере понять, где могут находиться потенциальные уязвимости или дефекты и каковы последствия эксплуатации этих недостатков. Каждая компьютерная система может состоять из множества различных временных бомб на разных уровнях ее структуры. На уровне компонентов это может быть переполнение буфера или опасные компоненты ActiveX, что может позволить злоумышленнику получить контроль над системой, воспользовавшись уязвимостью. На программном уровне приложение может небезопасно проводить авторизацию или может не

защищать свои криптографические ключи должным образом. На системном уровне ядро операционной системы может иметь недостатки, что позволит злоумышленнику без особого труда получить административный доступ. Различные ужасные вещи могут произойти на любом уровне, поэтому необходим столь детальный подход.

Подготовлено:							
Согласовано:							
Дата:							
Версия:							
				Дефект воздействует на...			
Идентификация элемента	Функция	Характер сбоя	Причина сбоя	Последствия	Последствия на более высоком уровне	Последствия для системы	Метод выявления сбоя
Контентный фильтр прикладного уровня IPS	Защита периметра	Сбой, приводит к отключению	Перегрузка вследствие большого объема трафика	Единая точка отказа; отказ в обслуживании	IPS блокирует входящий поток трафика	IPS выходит из строя	Сообщение о текущем состоянии (работоспособности), отправляется на консоль и по электронной почте администратору безопасности
Центральный "движок" обновления антивирусных сигнатур	Передаёт обновления сигнатур на все серверы и рабочие станции	Сбой не позволяет обеспечить адекватную и своевременную защиту от вредоносного кода	Отключается центральный сервер	Не обновляются антивирусы на серверах и рабочих станциях	Сеть заражается вредоносным кодом	Центральный сервер может быть заражен и/или заразить другие системы	Сообщение о текущем состоянии (работоспособности), отправляется на консоль и на страницу сетевого администратора
Водяные трубы системы пожаротушения	Тушение пожара в пяти зонах здания 1	Неисправности, приводят к неработоспособности	Вода в трубах замерзает	Нет	В здании 1 тушение пожара не производится	Трубы системы пожаротушения ломаются	Датчики системы пожаротушения напрямую связываются с центральной консолью пожарной системы
И т.д.							

Таблица 1-3 Пример проведения и документирования FMEA

Изначально FMEA была разработана для исследования систем. Ее цель заключается в том, чтобы изучить потенциальные дефекты в продуктах и связанных с ними процессах. Этот подход оказался успешным и был адаптирован для использования при оценке приоритетов в области управления рисками и минимизации известных угроз-уязвимостей.

Однако FMEA недостаточно эффективна при выявлении сложных дефектов, в которые могут быть вовлечены несколько различных систем или подсистем. В этом случае более целесообразно использовать анализ дерева сбоев (fault tree analysis). Для анализа с помощью дерева сбоев берется основной процесс. В качестве его корня (самого верхнего элемента этого логического дерева) указывается нежелательное событие. Затем, в качестве ветвей, добавляется ряд логических выражений и событий, которые могут привести к реализации вышестоящего нежелательного события. После этого дерево сбоев помечается цифрами, соответствующими вероятности сбоев (обычно это делается с помощью специализированных компьютерных программ, которые могут рассчитывать вероятности в дереве сбоев). На рисунке 1-7 показано упрощенное дерево сбоев и различные логические знаки, используемые для представления того, что должно произойти, чтобы вызвать сбой.

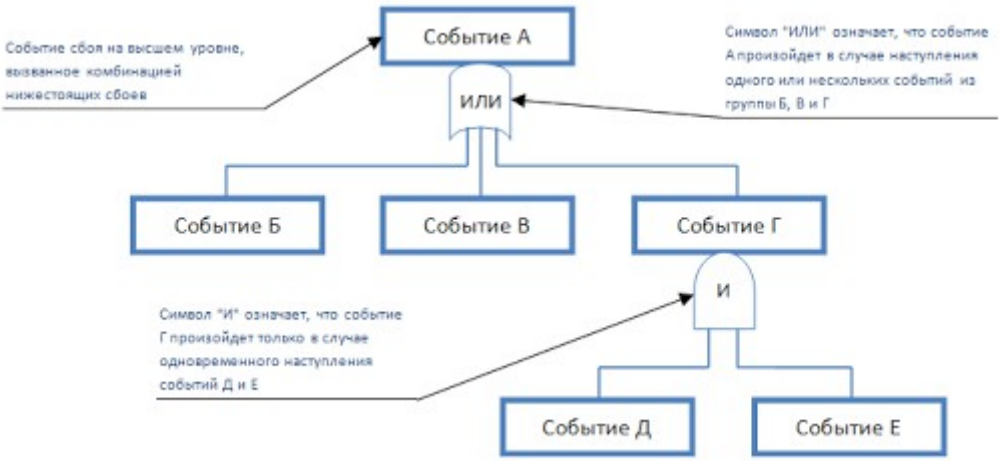


Рисунок 1-7 Дерево сбоя и логические элементы

При создании дерева, необходимо точно указать все угрозы или сбои, которые могут произойти с системой. Ветви дерева можно разделить на категории, например, физические угрозы, сетевые угрозы, компьютерные угрозы, интернет-угрозы и угрозы сбоя. Когда все возможные категории отмечены, вы можете подрезать ветви с дерева, удаляя угрозы, неприменимые в данном случае (если система не подключена к Интернету, то связанные с Интернетом ветви можно спокойно срезать с дерева).

Некоторые из наиболее распространенных программных сбоев, которые могут быть исследованы с помощью анализа дерева сбоев, приведены ниже:

- Ложные срабатывания (тревоги или защиты)
- Недостаточная обработка ошибок
- Нарушение последовательности или порядка
- Некорректная синхронизация выдачи результатов
- Корректные, но неожиданные результаты

Итак, мы получили надлежащую поддержку руководства в рамках задачи по анализу рисков, создали группу анализа рисков из сотрудников различных подразделений компании, определили ценность каждого из активов компании, определили все возможные угрозы, которые могут повлиять на активы. Мы также приняли во внимание все возможные варианты отложенного ущерба, который может выдержать компания в отношении каждого актива и угрозы. Мы провели анализ сбоев и дефектов (или анализ дерева сбоев) для понимания причин, лежащих в основе выявленных угроз. Следующим шагом является расчет актуальных для компании рисков с использованием качественных и количественных методов.

5.6. Количественный анализ рисков

Количественный анализ рисков пытается присвоить реальные и осмысленные числа всем элементам процесса анализа рисков. Этими элементами могут быть стоимость защитных мер, ценность актива, ущерб для бизнеса, частота возникновения угрозы, эффективность защитных мер, вероятность использования уязвимости и т.д. Количественный анализ рисков позволяет получить конкретное значение вероятности (в процентах) реализации угрозы. Каждый элемент в процессе анализа вставляется в количественном виде в уравнение определения общего и остаточного риска.

Нужно понимать, что количественный анализ рисков в чистом виде невозможен, так как всегда есть некоторая степень неопределенности в значении отдельных количественных величин (особенно если угрозы сложны, а частота их возникновения невелика). Например, как узнать насколько часто уязвимостью будут пользоваться? Или как узнать точную денежную сумму потерь компании? Даже если вы проанализировали все произошедшие ранее события, максимально точно определили ценность активов, обратились за информацией в организацию, которая оценивает частоту стихийных бедствий в вашей местности, вы все равно не сможете точно сказать, что для вашего дата-центра есть 10%-ная вероятность пожара и что пожар приведет к ущербу ровно в \$230,000. Будущее предсказать невозможно.

Автоматизированные методы анализа рисков

Сбор всех необходимых данных, которые необходимы для подстановки в уравнения анализа рисков и правильной интерпретации результатов, может быть крайне трудоемким, если он производится вручную. На рынке существует несколько автоматизированных средств анализа рисков, что может существенно упростить данную задачу и повысить точность

результатов. Собранные данные могут использоваться повторно, что значительно сократит время проведения повторного анализа. Имеющиеся автоматизированные инструменты могут помочь также при подготовке отчетов и всевозможных графиков для руководства.

Цель этих инструментов - сократить объем ручной работы, оперативно выполнять расчеты, оценивать ожидаемые потери, оценивать эффективность и преимущества выбранных контрмер.

Шаги процесса анализа рисков

Шаг 1: Определить ценность активов. Для каждого актива необходимо ответить на следующие вопросы для определения его ценности:

- В чем заключается ценность данного актива для компании?
- Сколько стоит его поддержка?
- Какую прибыль он приносит компании?
- Сколько за него готовы заплатить конкуренты?
- Сколько будет стоить его повторное создание или восстановление?
- Сколько стоило его получить или разработать?
- Какова мера ответственности в случае компрометации данного актива?

Шаг 2: Оценить потенциальные потери от угрозы. Для оценки потенциальных потерь, необходимо ответить на следующие вопросы:

- К каким физическим повреждениям может привести угроза и сколько это будет стоить?
- Какую потерю продуктивности может вызвать угроза и сколько это будет стоить?
- Какие потери понесет компания в случае разглашения конфиденциальной информации?
- Какова стоимость восстановления после воздействия угрозы?
- Какова стоимость потерь в случае неисправности критичных устройств?
- Каков ожидаемый ущерб от единичного инцидента (SLE – Single Loss Expectancy) для каждого актива и каждой угрозы?

Это только небольшой список вопросов, на которые необходимо получить ответы. Специфичные вопросы будут зависеть от типов угроз и выявленных группой особенностей.

Шаг 3: Выполнить анализ угроз. Для анализа угроз нужно выполнить следующие действия:

- Собрать информацию о вероятности каждой угрозы, опросив сотрудников каждого подразделения, проанализировав произведенные ранее записи, а также официальные источники по безопасности, которые предоставляют такую информацию.
- Рассчитать среднегодовую частоту возникновения инцидентов (ARO – Annualized Rate of Occurrence), которая показывает сколько инцидентов может произойти за год.

Шаг 4: Определить общие годовые потери на угрозу. Для этого нужно выполнить следующее:

- Объединить потенциальные потери и вероятность.
- Рассчитать ожидаемый среднегодовой ущерб (ALE – Annualized Loss Expectancy) на

угрозу, используя информацию, собранную на первых трех шагах.

- Выбрать контрмеры для противодействия каждой угрозе.
- Выполнить анализ затрат/выгод выбранных контрмер.

Шаг 5: Уменьшить, перенести, избежать или принять риск. Для каждого риска необходимо выбрать меры по его снижению, переносу, либо принять его.

• **Методы снижения риска:**

- Внедрить защитные меры и средства управления;
- Усовершенствовать процедуры;
- Изменить окружение;
- Внедрить методы раннего обнаружения для своевременного выявления факторов воздействия угрозы и снижения возможных последствий;
- Разработать план действий в непредвиденных ситуациях, позволяющий продолжить работу в случае воздействия определенных угроз и снизить последствия от угрозы;
- Создать препятствия для реализации угрозы;
- Провести тренинг по вопросам безопасности.

• **Перенос риска** – например, застраховать некоторые риски.

• **Избежание риска** – прекратить деятельность, вызывающую риск.

• **Принятие риска** – смириться с риском и не тратить деньги на защиту от него (это целесообразно, если стоимость защитных мер превышает величину возможного ущерба). Однако при этом нужно учитывать, что реализация риска может вести к дополнительным последствиям (например, потере репутации).

Принятие риска. Если компания решает принять риск, это решение должно основываться на стоимости (стоимость контрмер превышает возможные потери) и приемлемом уровне риска (при котором компания может жить с уязвимостью или угрозой). Но компания должна также понимать, что это не полноценное решение. Реализация риска может нести также и ущерб репутации, причем не только репутации компании, но и репутации целой отрасли.

При проведении количественного анализа рисков необходимы, реальные цифры и расчеты. Ранее уже были упомянуты показатели ***SLE (ущерб от единичного инцидента)*** и ***ALE (ожидаемый среднегодовой ущерб)***. SLE - это потенциальная сумма (в деньгах) ущерба для компании в результате единичного факта реализации соответствующей угрозы:

Ценность актива x Фактор воздействия (EF – Exposure Factor) = SLE

EF (фактор воздействия) – это процент ущерба для актива от реализовавшейся угрозы, т.е. часть значения (ценности), которую актив потеряет в результате инцидента. Например, ценность актива "хранилище данных" составляет \$150,000. В случае пожара может быть повреждено 25% хранилища (но не более, так как установлена система пожаротушения, поблизости находится пожарная часть и т.д.). В этом случае SLE будет составлять \$37,500. Значение SLE используется при расчете ALE:

SLE x Среднегодовая частота возникновения инцидентов (ARO – Annualized Rate of Occurrence) = ALE

ARO (среднегодовая частота возникновения инцидентов) – это величина, представляющая собой ожидаемую частоту реализации соответствующей угрозы в год. Значение ARO может быть от 0,0 (никогда) до 1,0 (по крайней мере, раз в год) и выше

(несколько раз в год). Например, наводнения в той местности, в которой расположено здание компании, происходят в среднем раз в 1000 лет. Значит величина ARO составляет 0,001.

Таким образом, если SLE для хранилища данных компании при пожаре равно \$37,500, а пожары в аналогичных условия случаются примерно раз в 10 лет (ARO равно 0,1), величина ALE будет равна \$3,750 ($\$37,500 \times 0,1 = \$3,750$).

Значение ALE используется при оценке целесообразности внедрения тех или иных мер защиты соответствующего актива от соответствующей угрозы - годовая стоимость защитных мер, обеспечивающих необходимый уровень безопасности актива, не должна превышать значение ALE. Применение более дорогих защитных мер не будет эффективным и целесообразным.

Рассмотрим пример результатов анализа рисков (Таблица 1-4). Используя полученные данные компания может принять обоснованное решение о том, какие угрозы необходимо рассматривать в первую очередь, основываясь на их последствиях и вероятности реализации. Также компания может оценить целесообразный уровень затрат на защиту от каждой угрозы.

Актив	Угроза	SLE	ARO	ALE
Здание	Пожар	\$230,000	0,1	\$23,000
Коммерческая тайна	Хищение	\$40,000	0,01	\$400
Файловый сервер	Неисправность	\$11,500	0,1	\$1,150
Данные	Вирус	\$6,500	1,0	\$6,500
Информация о кредитной карте клиента	Хищение	\$300,000	3,0	\$900,000

Таблица 1-4 Пример результатов анализа рисков

Результаты анализа рисков

Группа анализа рисков должна иметь четко определенные цели. Следующий список показывает что в основном можно ожидать от результатов анализа рисков:

- Ценность активов в денежном выражении;
- Полный список всех возможных и существенных угроз;
- Вероятная частота возникновения каждой угрозы;
- Потенциальные потери компании от угроз, которые она может понести в 12-ти месячный срок;
- Рекомендуемые меры безопасности, контрмеры и действия.

Хотя данный список следует по возможности детализировать, следует сделать краткое резюме для руководства, на основании которого можно быстро сделать выводы о результатах анализа.

5.7. Качественный анализ рисков

Другим методом анализа рисков является **качественный метод**, который не присваивает количественные или денежные значения компонентам и потерям, вместо этого он использует различные сценарии вероятности риска, уровней серьезности угроз и обоснованности различных возможных контрмер. Для качественного анализа рисков применяются суждения, лучшие практики, интуиция и опыт. Примерами техник сбора данных для качественного анализа рисков являются: метод Delphi, мозговой штурм, работа с архивными документами, опросы целевых групп, анкетирование, чек-листы, личные встречи, интервью. Группа анализа рисков должна выбрать лучшую технику в зависимости от видов оцениваемых угроз, культуры компании, людей, вовлеченных в процесс анализа.

В группу анализа рисков должны быть включены сотрудники, которые имеют опыт и образование в той области, в которой они будут оценивать угрозы. В процессе оценки

угрозы и ущерба каждый член группы высказывает свою оценку, основываясь на своем предчувствии и опыте.

Каждый член группы описывает приблизительный сценарий (не более страницы) для каждой существенной угрозы. Тот "эксперт", кто лучше всех разбирается в данном виде угроз, делает общий сценарий, описывающий процесс реализации угрозы. Затем оцениваются защитные меры, уменьшающие опасность этой угрозы, и описывается сценарий противодействия для каждой защитной меры. Возможное воздействие и возможный ущерб должны быть проранжированы по трех (высокий-средний-низкий), пяти или десятибалльной шкале. Уровни вероятности угрозы, потенциальных потерь и преимущества каждой защитной меры объединяются в отчет, который предоставляется руководству для принятия правильного решения. Преимущество данного метода анализа заключается в постоянном взаимодействии между всеми членами группы в процессах ранжирования рисков, определения сильных и слабых сторон защитных мер. Также преимуществом является подготовка окончательного отчета именно тем членом группы, который наиболее компетентен в соответствующей области.

Рассмотрим простой пример качественного анализа рисков. Группа анализа рисков написала одностраничный сценарий, описывающий угрозу получения хакером доступа к конфиденциальной информации, хранящейся на файловых серверах компании. Группа распространяет этот сценарий между пятью сотрудниками (ИТ-директор, администратор базы данных, программист, системный инженер и руководители подразделения технической поддержки), которые заполняют лист оценки, ранжируя (от 1 до 5) серьезность угрозы, уровень потенциального ущерба, эффективность каждой защитной меры. Результаты показаны в Таблице 1-5. Затем на основе этих результатов один из членов группы готовит отчет для руководства, в котором указывает, что из проанализированных трех вариантов защиты (межсетевой экран, система IDS и honeypot (приманка)), наиболее эффективной является защита с помощью межсетевого экрана.

<p>Угроза: Доступ хакера к конфиденциальной информации</p>						
	Серьезность угрозы	Вероятность реализации угрозы	Потенциальный ущерб для компании	Эффективность защиты с помощью межсетевого экрана	Эффективность защиты с помощью IDS	Эффективность защиты с помощью honeypot
ИТ-директор	4	2	4	4	3	2
Администратор БД	4	4	4	3	4	1
Программист	3	3	3	4	2	1
Системный инженер	4	4	3	4	2	1
Руководитель тех.поддержки	5	4	4	4	4	2
Итого:	3,6	3,4	3,8	3,8	3,0	1,4

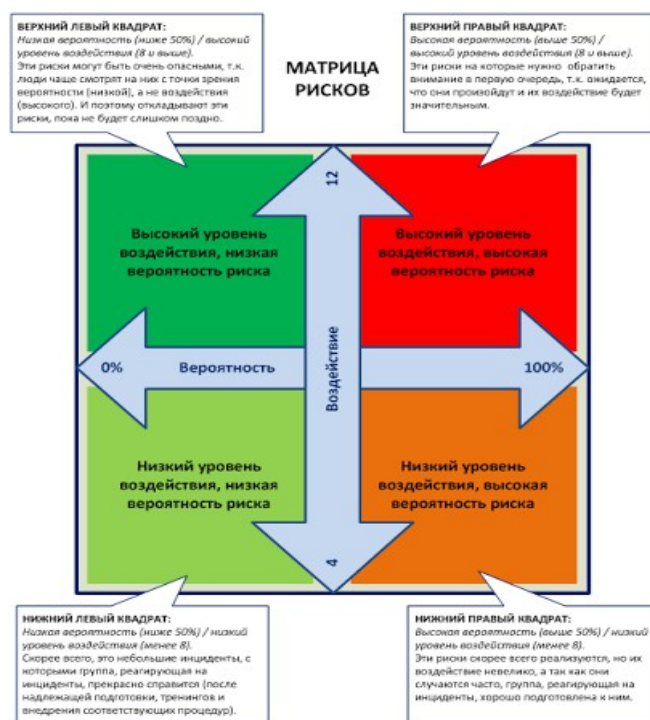
Таблица 1-5 Пример качественного анализа рисков

Анализируя отчет по анализу рисков, руководство видит оценки серьезности угроз, вероятности их реализации, а также уровень потенциальных потерь от каждой угрозы. На основе этой информации руководство может выбрать наиболее критичные для компании риски, которые должны быть учтены в первую очередь.

Техника DELPHI

Техника Delphi – это метод группового принятия решений, обеспечивающий получение от каждого члена его истинного мнения, не подверженного влиянию мнения других. Каждый член группы указывает свое мнение на листке бумаги, после чего все члены группы показывают свои листки для выполнения окончательного анализа. Результаты объединяются и распространяются между членами группы, которые пишут свои комментарии и возвращают их. Комментарии объединяются и снова распространяются до тех пор, пока не будет достигнут консенсус. Этот метод позволяет получить согласованное общее мнение по стоимости, уровню потерь, вероятности события без устного обсуждения. С использованием

данной техники возможно проведение анонимных опросов



5.8. Количественный или качественный

Каждый метод имеет свои преимущества и недостатки. Некоторые из них приведены в Таблице 1-6. Критерием выбора могут быть особенности группы анализа рисков, мнение руководства, имеющиеся инструменты анализа рисков, культура компании. Целью обоих методов является оценка реальных рисков компании, их классификация по уровню серьезности угроз, что позволит выбрать правильные контрмеры в рамках имеющегося бюджета.

Атрибут	Количественный	Качественный
Требует простых расчетов		X
Требует более сложных расчетов	X	
Основывается в значительной степени на предположениях		X
Предоставляет основные области и показатели риска		X
Проще автоматизировать	X	
Позволяет контролировать эффективность управления рисками	X	
Предоставляет заслуживающий доверия анализ стоимости / выгоды	X	
Использует независимо проверяемые и объективные метрики	X	
Предоставляет мнения людей, которые в совершенстве знают анализируемые процессы		X
Четко показывает потери, которые могут нарастать в течение года	X	

Таблица 1-6 Характеристики количественного и качественного методов

Недостатки качественного анализа:

- Оценка и результаты в основном субъективны;
- Обычно исключает возможность денежной оценки затрат/выгод;
- Сложно сопоставить цели управления рисками с субъективными оценками;
- Нет стандартов. Каждый специалист имеет свою интерпретацию процесса и результатов.

Недостатки количественного анализа:

- Расчеты более сложны. Сложно объяснить руководству как эти значения были

получены;

- Процесс очень трудоемок без применения средств автоматизации;
- Больше предварительной работы, связанной с получением детальной информации об окружении;
- Нет стандартов. Каждый специалист имеет свою интерпретацию процесса и результатов.

Уровень неопределенности – это степень неуверенности в оценке. Он измеряется от 0% до 100%. При проведении анализа следует указывать уровень неопределенности, так как это способствует лучшему пониманию результатов анализа.

5.9. Защитные механизмы

Следующим шагом является идентификация доступных защитных механизмов (контрмер) и оценка их эффективности.

Выбор защитных мер

Преимущества внедряемых защитных механизмов должны превышать затраты на них, только в этом случае они будут эффективными. Для проведения такой оценки используется анализ затрат/выгод. Обычно это считают следующим образом:

$$(\text{ALE до ввода защитных мер}) - (\text{ALE после ввода защитных мер}) - (\text{годовая стоимость защитных мер}) = \text{ценность защитных мер}$$

Например, ALE угрозы выведения веб-сервера из строя хакерами составляет \$12,000 до внедрения соответствующих защитных мер, а после их внедрения ALE снижается до \$3,000. При этом годовая стоимость функционирования и поддержки этих защитных мер - \$650. В этом случае ценность защитных мер составляет \$8,350 в год.

Стоимость контрмер – это не просто цена их покупки. Нужно учитывать следующие элементы при расчете полной стоимости контрмер:

- Стоимость продукта
- Стоимость проектирования / планирования
- Стоимость внедрения
- Необходимость внесения изменений в окружение
- Совместимость с другими контрмерами
- Эксплуатационные требования
- Тестирование
- Стоимость восстановления, замены и обновления
- Стоимость эксплуатации и поддержки
- Влияние на производительность
- Стоимость подписки
- Работа сотрудников в нерабочее время и по выходным дням для мониторинга и реакции на сообщения об инцидентах

ПРЕДУПРЕЖДЕНИЕ. Очень часто компании покупают новые продукты безопасности, не понимая, что им нужен дополнительный персонал на использование этих продуктов. Хотя инструменты автоматизируют задачи, многие компании ни разу не выполняли эти задачи перед покупкой, поэтому с помощью них они не сэкономят время, а наоборот начнут тратить его еще

больше.

Например, компания А решает защитить ряд своих ресурсов с помощью IDS, стоимостью \$5,500. Эта IDS должна быть протестирована в отдельном тестовом сегменте сети, чтобы ИТ-службы убедились в безопасности ее ввода в промышленную эксплуатацию. После этого ИТ-службы должны установить и настроить сенсоры и программное обеспечение для мониторинга. Затем ИТ-службы должны перенастроить коммуникационное оборудование, направив все потоки трафика через IDS, а также ограничить доступ к консоли IDS, настроить базу данных сигнатур атак. Только после этого IDS можно включить в работу.

При этом нужно учесть вероятное снижение производительности сети, возможные неудобства для пользователей, проявление "тонкостей", о которых "забыл" заранее сообщить поставщик, а также затраты времени и денег на обучение персонала, а затем затраты на реагирование на реальные и ложные срабатывания IDS. Кроме того, может возникнуть необходимость закупки средств оповещения администратора безопасности об инцидентах, выявленных IDS (например, пейджером или Blackberry), от которых будет зависеть время реакции на инциденты.

Таким образом изначальная цена увеличивается: \$5,500 стоит сама система IDS, \$2,500 стоит обучение персонала, \$3,400 стоит тестирование системы, \$2,600 - потери производительности пользователей, вызванные внедрением этой системы и еще \$4,000 стоит перенастройка коммуникационного оборудования, установка IDS, выявление ошибок, установка патчей. Реальная стоимость этой защитной меры составит \$18,000. Если при этом рассчитанная величина вероятного ущерба составляет только \$9,000, применение этой IDS неэффективно и приведет к перерасходу адекватного бюджета на 100%.

Функциональность и эффективность защитных мер

Группа анализа рисков должна оценить функциональность и эффективность защитных мер. При выборе защитных мер некоторые характеристики будут важнее, чем другие. В таблице 1-7 перечислены и описаны характеристики, которые должны быть тщательно проанализированы до покупки и внедрения средства обеспечения безопасности.

Характеристика	Описание
Модульная архитектура	Система может быть установлена или удалена из окружения без неблагоприятного воздействия на другие механизмы
Обеспечивает стандартную защиту	Стандартный уровень безопасности, обеспечивается стандартными настройками всех механизмов
Предоставляет возможность отмены функциональности	Администратор может отменить ограничения при необходимости
Минимальные привилегии по умолчанию	После установки должны применяться настройки по умолчанию, не предусматривающие какие-либо разрешения и права, кроме заданных явно
Независимость средств защиты и защищаемых активов	Средства защиты могут быть использованы для защиты различных активов, а различные активы, в свою очередь, могут быть защищены различными средствами
Гибкость и безопасность	Должно быть предоставлено как можно больше функций безопасности, однако настройки при этом должны быть гибкими и позволять выбрать нужный набор функций (а не «все», либо «ничего»)
Явное разделение «пользователя» и «администратора»	Пользователь должен иметь минимум разрешений на изменение настроек или отключение механизмов защиты
Минимальное вмешательство человека	Средства защиты должны предусматривать минимальное вмешательство человека, так как любые производимые вручную настройки и изменения с высокой степенью вероятности приводят к ошибкам
Простота обновления	Программное обеспечение продолжает развиваться, поэтому важно, чтобы обновления устанавливались безболезненно
Средства аудита	Должен существовать механизм, являющийся неотъемлемой частью средств защиты, предоставляющий минимальный и/или подробный аудит событий
Минимальная зависимость от других компонентов	Средства защиты должны быть гибкими и не предъявлять жестких требований к окружению, в которое они устанавливаются
Простота использования, незаметность для персонала	Средства защиты не должны снижать производительность работы, добавлять дополнительные шаги к простым задачам; пользователи не должны испытывать дискомфорт из-за их применения
Исходящая информация (отчеты) должна предоставляться в простом и понятном виде	Важная информация должна быть представлена в простой для человека форме, понятной и применимой для анализа изменений и тенденций
Должна существовать возможность сброса настроек средства защиты	Средства защиты должны позволять сбросить настройки и вернуться к оригинальной конфигурации и настройкам, что не должно оказывать влияние на защищаемые системы и активы
Возможность тестирования	Должна существовать возможность протестировать работу средств защиты в различных окружениях и в различных ситуациях
Отсутствие компромиссов	Средства защиты не должны содержать скрытых каналов и потайных входов (back doors)
Производительность систем и пользователей	Применение средств защиты не должно оказывать существенного влияния на производительность систем и пользователей
Качественная система оповещений	Порог срабатывания системы оповещения персонала об инцидентах безопасности должен быть настраиваемым, типы оповещений должны быть приемлемыми
Не должно оказываться влияние на активы	Средства защиты не должны оказывать неблагоприятного воздействия на активы

Таблица 1-7 Характеристики, которые следует учесть при выборе средств защиты

Защитные средства могут также иметь сдерживающие атрибуты, говорящие потенциальному злоумышленнику, что здесь внедрена надежная защита и ему лучше поискать другую, более легкую цель. Однако здесь также следует соблюдать определенную степень осторожности и не предоставлять потенциальному злоумышленнику излишней информации о применяемых средствах защиты и методах их работы, так как это может позволить ему обойти их. Если пользователи знают, как отключить антивирусную программу, чтобы повысить производительность своего компьютера, или как обойти прокси-сервер, чтобы получить неограниченный доступ в Интернет, они будут делать это.

5.10. Обобщая сказанное ранее

Для проведения анализа рисков, компания сначала решает, какие активы нуждаются в защите и в какой степени. Указывается, какие суммы денежных средств могут быть потрачены на защиту данных активов. Далее, оценивается функциональность доступных средств защиты и определяются те, которые будут наиболее эффективны для компании. После этого, компания должна оценить и сопоставить расходы на защитные меры. Эти шаги позволят руководству принять разумное и осознанное решение о выборе и покупке контрмер.

Необходимо учитывать, что только переоценивая риски на периодической основе можно обеспечить постоянную эффективность защитных мер и поддерживать уровень рисков информационной безопасности на приемлемом уровне. Если риск не изменился и защитные меры внедрены и хорошо работают, значит риски снижены достаточно. Продолжение анализа уязвимостей и выявления нуждающихся в защите активов также является важной

задачей управления рисками.

5.11. Общий риск и Остаточный риск

Общий риск (total risk) - это риск, перед лицом которого стоит компания, не внедрившая никаких защитных мер. Если его уровень не приемлем для компании (вероятный ущерб от реализации риска превышает стоимость защитных мер), она внедряет защитные меры чтобы снизить общий риск до приемлемого уровня. Однако систем или сред, защищенных на 100%, не существует - всегда есть некоторый **остаточный риск (residual risk)**. Необходимо обеспечить, чтобы уровень остаточного риска был приемлем для компании.

Общий риск = угроза x уязвимость x ценность актива

Остаточный риск = (угроза x уязвимость x ценность актива) x
недостаток контроля

Это никогда не закончится. Необходимо учитывать, что только переоценивая риски на периодической основе можно обеспечить постоянную эффективность защитных мер и поддерживать уровень рисков информационной безопасности на приемлемом уровне. Если риск не изменился и защитные меры внедрены и хорошо работают, значит риски снижены достаточно. Продолжение анализа уязвимостей и выявления нуждающихся в защите активов также является важной задачей управления рисками.

5.12. Обработка риска

Когда компания рассчитала величину общего и остаточного риска, она должна принять решение о дальнейших действиях. Существует 4 варианта действий, которые можно предпринять в отношении риска: перенести, избежать, уменьшить или принять риск.

Если общий или остаточный риск слишком высок для компании, она может купить страховку, чтобы **перенести** риск на страховую компанию.

Если компания решает прекратить деятельность, которая вызывает риск, это называется **избеганием** риска. Например, компания может запретить использование сотрудниками программ передачи мгновенных сообщений (IM – Instant Messenger), вместо того, чтобы бороться с множеством рисков, связанных с этой технологией.

Другим подходом является **снижение** риска до уровня, считающегося приемлемым для компании. Примером может быть внедрение межсетевого экрана, проведение обучения сотрудников и т. д.

И последний подход заключается в осознанном **принятии** риска компанией, которая осознает его уровень, размеры потенциального ущерба, и, тем не менее, решает жить с этим риском и не внедрять контрмеры. Для компании целесообразно принять риск, когда анализ затрат / выгоды показывает, что расходы на контрмеры превышают размеры потенциальных потерь.

Ключевым вопросом при принятии риска является понимание того, почему это является наилучшим выходом из конкретной ситуации. К сожалению, в наше время многие ответственные лица в компаниях принимают риски, не понимая в полной мере, что они принимают. Обычно это связано с относительной новизной процессов управления рисками в области безопасности, недостаточным уровнем образования и опытом работы этих людей. Когда руководителям бизнес-подразделений вменяются обязанности по борьбе с рисками в их подразделениях, чаще всего они принимают любые риски, т.к. их реальные цели связаны с производством компанией готовой продукции и выводом ее на рынок, а вовсе не с рисками. Они не хотят увязать в этой глупой, непонятной и раздражающей безопасности...

Принятие риска должно быть основано на нескольких факторах. В частности, нужно ответить на следующие вопросы. Потенциальные потери меньше стоимости контрмер?

Сможет ли компания жить с той "болью", которую причинит ей принятие этого риска?

Второй вопрос имеет отношения в том числе и к бесплатным решениям. Например, если компания примет этот риск, она должна будет добавить еще три шага в свой производственный процесс. Имеет ли это смысл для нее? В другом случае, принятие риска может привести к возрастанию количества инцидентов безопасности – готовы ли компания справиться с этим?

Человек или группа, принимающая риск, должны понимать потенциальные последствия этого решения. Предположим, было установлено, что компания не нуждается в защите имен клиентов, но она должна защищать другие сведения, такие как номера социального страхования, номера счетов и т.д. При этом ее деятельность останется в рамках действующего законодательства. Но что будет, если ее клиенты узнают что компания не защищает должным образом их имена? Ведь они из-за отсутствия знаний по этому вопросу могут подумать, что это может стать причиной «кражи личности», что приведет к серьезному удару по репутации компании, с которым она может и не справиться. Восприятие клиентов часто не обосновано, и всегда существует вероятность, что они перенесут свой бизнес в другую компанию, и вам нужно считаться с этой потенциальной возможностью.

Рисунок 1-8 показывает, как может быть создана программа управления рисками, которая объединяет все понятия, описанные в настоящем разделе.

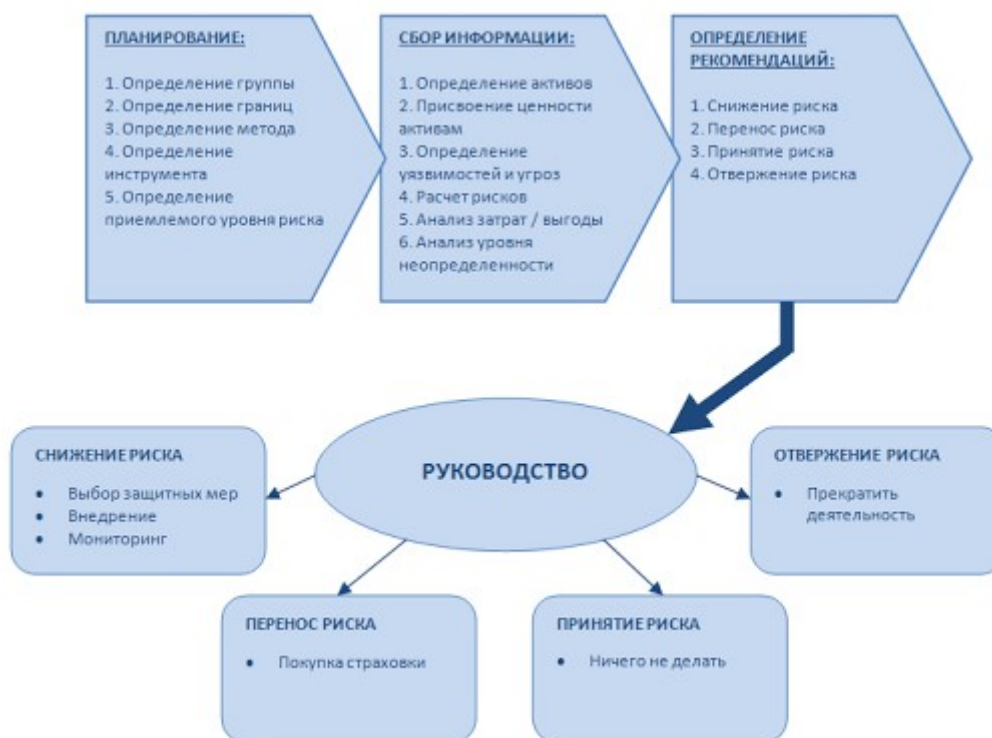


Рисунок 1-8. Как может быть создана программа управления рисками

Ссылки по теме:

- Risk Management Guide for Information Technology Systems, Recommendations of the National Institute of Standards and Technology
- Carnegie Mellon Software Engineering Institute
- Handbook of Information Security Management, Domain 3, “Risk Management and Business Continuity Planning,” Micki Krause and Harold F. Tipton, editors (CRC Press LLC)
- Security Metrics Guide for Information Technology Systems
- Threat and Risk Assessment Working Guide (Government of Canada, 1999)

6. Политики, стандарты, базисы, руководства и процедуры

Компьютеры и обрабатываемая ими информация обычно имеют прямое отношение к

критичным целям и миссии компании. Поэтому для высшего руководства обеспечение безопасности должно иметь высокий приоритет., а подход руководства должен быть комплексным. Руководством должна обеспечиваться необходимая поддержка, финансирование, выделение других необходимых ресурсов. У каждого сотрудника в компании есть свои взгляды на безопасность, обусловленные его ценностями и опытом в данной сфере. Важно, чтобы эти взгляды соответствовали тому уровню безопасности, в котором нуждается компания.

Высшее руководство должно определить область безопасности, решить, что нуждается в защите и в какой степени. Руководство должно понимать требования, законы и обязательства, которым должна соответствовать компания. Также оно должно определить, что ожидается от сотрудников и каковы для них последствия нарушения требований. Эти решения должны быть сделаны людьми, которые в конечном счете несут ответственность, если что-то в компании идет не так.

Программа безопасности включает все необходимые части для защиты компании в целом и определяет долгосрочную стратегию безопасности. Программа безопасности должна содержать политики, процедуры, стандарты, руководства, базисы, обучение по вопросам безопасности, план реагирования на инциденты, программу соответствия требованиям. К подготовке этих компонентов должны привлекаться департамент по работе с персоналом и юридический департамент.

Разработчиками политики должны быть проанализированы такие вопросы, как "язык", уровень детализации, степень формальности политики и механизмы ее поддержки. Все документы по безопасности должны быть реалистичны, это позволит им быть эффективными. Кроме того, документы должны быть в меру детализированными, так как излишне детализированные документы скорее всего будут чересчур утомительными и сложными для восприятия, что сильно снизит их эффективность. Также следует учитывать, что более формальные требования проще внедрить. "Язык" документов должен соответствовать типу бизнеса, его целям, культуре компании, а также ориентироваться на ту аудиторию, для которой предназначается документ (для руководителей, для рядовых сотрудников, для ИТ-сотрудников и т.д.).

6.1. Политика безопасности

Политика безопасности – это всеобъемлющее основное заявление высшего руководства компании, указывающее роль безопасности в организации. Политика должна быть независимой с точки зрения технологий и решений. Она должна очерчивать цель и миссию, но не привязывать компанию к конкретным способам их достижения.

Политика безопасности может быть организационной, ориентированной на задачи или ориентированной на системы. **В организационной политике безопасности (*organizational security policy*)** руководство устанавливает порядок внедрения программы безопасности, определяет цели программы, распределяет ответственность, показывает стратегическое и тактическое значение безопасности, указывает как все это будет реализовано. Такая политика должна учитывать законы, требования, обязательства, описывать порядок обеспечения соответствия им. Организационная политика безопасности определяет область и направление всей будущей деятельности по безопасности в компании. Также она описывает величину риска, которую высшее руководство считает приемлемой.

Зачем нужны политики? Ниже приводится краткое резюме в отношении важности политики безопасности:

- Определяет активы, которые компания считает ценными
- Предоставляет полномочия группе безопасности и ее деятельности
- Является основанием в процессе разрешения возникающих конфликтов, связанных с безопасностью

- Фиксирует цели и задачи компании, относящиеся к безопасности
- Очерчивает персональную ответственность
- Помогает предотвратить необъяснимые события (сюрпризы)
- Определяет границы работы и функции для группы безопасности
- Очерчивает обязанности в отношении реагирования на инциденты
- Очерчивает действия компании в части должной заботы, связанной с требованиями законодательства, регуляторов и стандартов

Организационная политика безопасности имеет несколько важных характеристик, которые должны быть поняты и внедрены:

- Бизнес-цели должны управлять созданием, внедрением и исполнением политики. Политика не должна создавать бизнес-цели.
- Политика должна быть простой для понимания и использоваться как ориентир всеми сотрудниками и руководством компании.
- Политика должна быть разработана и использоваться для интеграции безопасности во все бизнес-функции и процессы компании.
- Политика должна учитывать все законы и требования, предъявляемые к компании.
- Политика должна пересматриваться и модифицироваться при изменениях самой компании, таких как принятие новой модели бизнеса, объединение с другой компанией или изменение владельца компании.
- Каждое обновление политики должно быть датировано и ему должен быть присвоен номер версии.
- Подразделения и сотрудники, которые руководствуются политикой, должны иметь доступ к той части политики, которая применима к ним. Не нужно требовать от них читать всю политику, чтобы найти ответы на свои вопросы.
- Политика должна быть рассчитана на несколько лет вперед и учитывать возможные изменения безопасности окружения, которые могут произойти в ближайшем будущем.
- Следует использовать прямой и командный язык. Слова типа «следует» или «может» нужно заменить на «должен».
- Уровень профессионализма в презентации политик усиливает их значимость и заставляет присоединиться к ним.
- Политика не должна содержать формулировок, которые могут быть понятны не всем. Используйте ясные и декларативные заявления, которые легко понять и принять.
- Политика должна пересматриваться на регулярной основе и адаптироваться с учетом инцидентов, которые произошли с момента последнего пересмотра Политики.

Должен быть разработан и внедрен четкий и понятный порядок применения мер воздействия в отношении тех, кто не соблюдает требования политик безопасности. Это позволяет всем понять не только то, что от них ожидается, но и каковы будут последствия несоблюдения.

Политика, ориентированная на задачи (*issue-specific policy*), учитывает конкретные вопросы безопасности, которые руководство считает необходимым разъяснить и акцентировать на них внимание, чтобы убедиться в создании всеобъемлющей структуры безопасности и понимании всеми сотрудниками, как они должны исполнять задачи безопасности. Примером такой политики может быть политика использования электронной

почты, которая описывает цели и порядок использования электронной почты сотрудниками компании, устанавливает меры ответственности за нарушение требований политики, а также определяет все действия, которые может производить руководство с электронной перепиской сотрудников для выполнения задач контроля использования электронной почты, а также для выявления возможной утечки конфиденциальной информации.

Политика, ориентированная на системы (system-specific policy), описывает решения руководства в отношении конкретных компьютеров, сетей, приложений и данных. Такой тип политики может содержать, например, принятый в компании список программного обеспечения, которое может быть установлено на рабочие станции компании. Ориентированная на системы политика может описывать как базы данных должны использоваться и защищаться, как должны работать межсетевые экраны, системы выявления вторжений (IDS), сканеры и т. д.

Политики пишутся в широких терминах и покрывают множество однотипных устройств. Более детализированные документы требуют частой актуализации – ими являются процедуры, стандарты, руководства, а политики предоставляют основу. Процедуры, стандарты и руководства составляют структуру безопасности. А необходимые компоненты и механизмы безопасности заполняют эту структуру, обеспечивая выполнение программы безопасности и предоставляя защищенную инфраструктуру.

Типы политик. Политики в основном попадают в одну из следующих категорий:

- **Регулирующая (regulatory)**, обеспечивающая следование компании набору специальных отраслевых требований и стандартов. Такая политика очень детальна и зависит от отрасли компании. Используется в финансовых, медицинских и других управляемых государством отраслях.
- **Рекомендательная (advisory)**, настоятельно рекомендуемая сотрудникам придерживаться определенного поведения, указывающая разрешенные и запрещенные в компании действия и предусматривающая последствия для сотрудников за нарушения. Такая политика может использоваться, например, для описания порядка обращения с конфиденциальной информацией, финансовыми транзакциями.
- **Информативная (informative)**, информирующая сотрудников по определенным вопросам, описывающая подход компании. Например, она может объяснять порядок общения с партнерами, цели и миссию компании и т.д.

6.2. Стандарты

Стандарты (standards) – это обязательные действия или правила. Стандарты поддерживают и развивают политику по определенным направлениям. Стандарты могут быть внутренними и внешними (например, законодательство).

Стандарты могут, например, указывать как следует использовать программное обеспечение и оборудование, как следует вести себя пользователям. Они могут обеспечить единообразие технологий, приложений, параметров, процедур в рамках всей компании. Организационные стандарты могут требовать от всех сотрудников постоянно носить идентификационные бейджи (badge), чтобы было проще отличить сотрудников компании от посетителей.

Как было указано ранее, существует большая разница между стратегическими и тактическими целями. Стратегические цели описывают окончательный результат, а тактические – шаги его достижения. Стандарты, руководства и процедуры – это тактические инструменты для поддержания и достижения положений политики безопасности, которая соответствует стратегическим целям (см. Рисунок 1-9).



Рисунок 1-9. Политика определяет стратегические планы, а нижестоящие документы обеспечивают тактическую поддержку.

6.3. Базисы

Базис (baseline) – это точка во времени, которая используется для сравнения с будущими изменениями. Когда риски снижены и меры безопасности внедрены, базисы пересматривают, чтобы иметь возможность более адекватно контролировать эффективность будущих изменений. Базис - это некий ориентир.

Например, ваш доктор говорит вам, что вы весите 200 килограмм потому что сидите на диете из пончиков, пиццы и газированной воды. Доктор говорит, что вам нужно ежедневно по 30 минут делать упражнения, в процессе которых ваш пульс должен увеличиваться вдвое по сравнению с нормальным пульсом. Как вы определите что ваш пульс увеличился вдвое? Вам нужно сначала определить свой нормальный пульс (базис), исходя из которого вы будете измерять свой пульс при выполнении упражнений.

Базисы также применяют как определение минимально необходимого уровня защиты для определенных типов систем. Например, компания может поставить требование, что все ее бухгалтерские системы должны соответствовать 4 уровню требований EAL (Evaluation Assurance Level). Правильная настройка соответствующих требованиям систем будет являться необходимым базисом. Однако нужно учитывать, что установка нового программного обеспечения, обновлений, патчей, либо проведение других изменений может привести к снижению уровня защиты ниже минимального уровня (базиса). Поэтому сотрудники безопасности должны предварительно оценивать все планируемые изменения, чтобы обеспечить постоянное соблюдение базисного уровня безопасности.

ПРИМЕЧАНИЕ. Базисы не следует жестко ориентировать на конкретные технологии, применяемые в компании.

ПРЕДУПРЕЖДЕНИЕ. Термин "базис" часто интерпретируется по-разному в различных отраслях. Чаще всего базисом называется определенная конфигурация программного или аппаратного обеспечения, обеспечивающая минимально необходимый уровень безопасности.

6.4. Руководства

Руководства (guidelines) описывают рекомендуемые действия и являются эксплуатационными инструкциями для пользователей, ИТ-специалистов и других сотрудников, там, где не применяются соответствующие стандарты. Рекомендации могут касаться технологических методик, персонала или физической безопасности.

Рекомендации, в отличие от обязательных к исполнению жестких стандартов, показывают основной подход, имеющий определенную гибкость в непредвиденных обстоятельствах. Например, политика может потребовать, чтобы доступ к конфиденциальным данным регистрировался. Поддерживающие политику руководства в дальнейшем поясняют, что

регистрируемые события должны содержать достаточно информации для отслеживания фактов доступа. Поддерживающие политику процедуры описывают необходимые шаги конфигурации, внедрения и сопровождения этого типа аудита.

6.5. Процедуры

Процедуры (procedures) – это детальные, описанные «шаг за шагом» задачи, выполняемые чтобы достичь определенной цели. Шаги могут выполняться пользователями, ИТ-специалистами, сотрудниками безопасности и другими сотрудниками, выполняющими специфические задачи. Многие компании пишут процедуры установки операционных систем, настройки механизмов безопасности и списков контроля доступа, регистрации новых пользователей, присвоения им привилегий, ведения аудита журналов регистрации событий, уничтожения информации, отчета об инцидентах и т. д.

Процедуры занимают низший уровень в цепочке политик, т.к. они относятся к компьютерам и пользователям и описывают, например, шаги по настройке конфигурации или инсталляции ПО.

Процедуры детально описывают, как политики, стандарты и рекомендации будут фактически внедряться в операционную среду. Например, если стандарт требует создания резервных копий, процедуры описывают детальные шаги, необходимые для выполнения резервного копирования, время создания копий, место для размещения копий и т.д. Процедуры должны быть достаточно детальными, чтобы быть понятными и полезными различным группам людей.

Чтобы связать все эти элементы вместе, давайте рассмотрим пример. Корпоративная политика безопасности указывает, что конфиденциальная информация должна быть защищена. Это очень широкая и общая формулировка. Поддерживающий политику стандарт требует шифровать всю клиентскую информацию при ее хранении в базах данных с помощью алгоритма AES, а при передаче через Интернет использовать технологию IPSec. По сравнению с политикой, стандарт более детален – он содержит информацию о конкретных типах защиты. Поддерживающие процедуры детально объясняют процесс внедрения и использования технологий AES и IPSec, а руководства разъясняют что следует делать при умышленном повреждении данных или их компрометации в процессе передачи. Все эти документы работают совместно для обеспечения структуры безопасности компании.

ПРИМЕЧАНИЕ. Не следует объединять стандарты, руководства и базисы в один большой документ, т.к. каждый из них имеет свою цель и свою аудиторию.

6.6. Внедрение

Политики безопасности и поддерживающие их документы должны быть не только разработаны, но и внедрены. Для всех документов должно быть обеспечено их исполнение.

Сотрудники должны узнавать о безопасности именно из этих документов, поэтому они должны быть доступны. Для доведения документов до сведения сотрудников можно использовать тренинги по вопросам безопасности, инструкции, презентации, письма, баннеры и т.д. Сотрудники должны понимать, что от них ждут, как они должны вести себя, какая требуется отчетность. Следует информировать сотрудников о последствиях несоблюдения требований и об их ответственности.

Внедрение в реальную работу политик безопасности и поддерживающих их документов показывает действительную заботу компании о своей безопасности.

Ссылки по теме:

- NCSA Security Policies and Procedures
- SANS Institute Security Policy Project

- Information Security Policy World

7. Классификация информации

Ранее мы уже касались вопроса о важности понимания ключевого значения информации для бизнеса компании. Однако не вся информация имеет одинаковую ценность для компании и было бы неэффективным расходовать одинаковые ресурсы на защиты различных типов информации, имеющих различный уровень ценности. Поэтому, после идентификации всей важной информации, она должна быть правильно классифицирована в соответствии с уровнем ущерба в случае ее разглашения или недоступности.

Результаты классификации информации позволят компании определить, какие средства и меры защиты должны применяться для каждого класса, решить какие задачи защиты информации являются наиболее приоритетными. Первичная цель классификации информации – показать необходимый для каждого типа информации уровень защиты конфиденциальности, целостности и доступности. Кроме того, классификация позволяет обеспечить защиту информации наиболее эффективным (экономически) способом. Защита и содержание информации стоит денег, желательно расходовать эти деньги именно на ту информацию, для которой это действительно необходимо.

Для каждого класса должны быть определены отдельные требования и процедуры по использованию, контролю доступа и уничтожению. Например, компания может установить следующие процедуры:

- **Конфиденциальная информация.** Доступ к конфиденциальной информации имеет только высшее руководство и еще несколько ответственных (назначенных) сотрудников компании. Для получения доступа два ответственных сотрудника (руководителя) должны одновременно ввести коды доступа. Все факты доступа должны регистрироваться и ежедневно контролироваться. Все бумажные копии конфиденциальных документов должны храниться в хранилище. При удалении конфиденциальных данных с электронных носителей информации должны использоваться устройства размагничивания, либо специализированные программные средства, затирающие информацию без возможности восстановления.
- **Критичная информация.** Доступна гораздо большему количеству сотрудников. Для доступа к этой информации достаточно только ввести пароль. Журналы регистрации событий доступа должны просматриваться еженедельно. Бумажные копии должны храниться в запираемых шкафах в офисных помещениях. С электронных носителей данные должны удаляться с помощью штатных средств операционной системы.
- **Открытая информация.** Остальную информацию компания классифицирует как открытую и не устанавливает ограничений по ее использованию, контролю доступа и порядку уничтожения.

Модели безопасности в различных видах организаций могут существенно различаться. Например, военные организации больше внимания уделяют вопросам конфиденциальности информации, а для частного бизнеса более важны целостность и доступность. Это существенно влияет на классификацию данных. Для начала нужно решить, какую схему классификации будет использовать компания. Некоторые компании используют только два класса информации, другие – больше.

Коммерческие организации часто используют следующие уровни критичности: конфиденциально (confidential), для внутреннего использования (private), критичная информация (sensitive), открытая информация (public). А военные организации используют другие уровни критичности: совершенно секретно (top secret), секретно (secret), конфиденциально (confidential), критичная неклассифицированная информация (sensitive but unclassified), неклассифицированная информация (unclassified).

Классификация, обычно применяемая в коммерческом секторе, описана ниже:

- **Информация для служебного пользования** – критичная финансовая информация компании
- **Коммерческая тайна** – информация, обеспечивающая конкурентное преимущество компании
- **Конфиденциальная информация** – информация, защита которой требуется в соответствии с бизнес-стандартами и законодательством
- **Персональные данные** – информация, содержащая записи о людях

После того, как схема классификации определена, компания должна разработать критерии для отнесения информации (данных) к тому или иному классу. Следующий список содержит некоторые критерии, которые можно использовать для определения критичности данных:

- Полезность данных
- Ценность данных
- Новизна данных
- Уровень ущерба в случае разглашения данных
- Уровень ущерба в случае модификации или повреждения данных
- Обязанности по защите данных в соответствии с законодательством, требованиями регуляторов, договорами с контрагентами
- Влияние данных на национальную безопасность
- Кто должен иметь доступ к данным
- Кто должен поддерживать данные
- Где данные должны храниться
- Кто должен иметь возможность воспроизводить (reproduce) данные
- Какие отметки или специальную маркировку требуют данные
- Требуется ли шифрование данных
- Требуется ли разделение обязанностей (separation of duties)
- Величина упущенной выгоды, вызванной недоступностью или повреждением данных

Информация – это не единственная вещь, которая требует классификации. Приложения, а в некоторых случаях и целые системы, также должны быть классифицированы. Приложение, которое хранит или обрабатывает классифицированные данные, должно быть оценено по уровню безопасности, которое оно предоставляет. Классификацию приложений следует основывать на уверенности (уровне доверия) компании в этом программном обеспечении и на типе информации, которая хранится и обрабатывается в нем.

ПРИМЕЧАНИЕ. Также, не следует забывать про резервные копии классифицированных данных – к ним должны иметь доступ только те, кто имеет соответствующий уровень допуска. Большой риск для компании представляет технический персонал, который, не обладая никакими допусками, работает с классифицированными данными при выполнении своих обязанностей. Компания должна удостовериться, что каждый, кто имеет доступ к данным и их резервным копиям, ясно понимает уровень их важности, знает свои обязанности по отношению к ним.

ПРЕДУПРЕЖДЕНИЕ. Правила классификации должны применяться к данным независимо от формы, в которой они представлены: электронная информация, бумага, видео, аудио, факс и т.д.

После выбора порядка оценки критичности данных, необходимо определить для каждого

класса требования по контролю доступа, идентификации, маркировке (на всех этапах хранения данных), поддержке, передаче, уничтожению. Также нужно учесть вопросы аудита, мониторинга, контроля соответствия. Каждый класс данных требует различного уровня безопасности, поэтому для каждого класса определяются свои требования по защите и управлению.

Процедуры классификации данных. Следующие шаги необходимы для качественной программы классификации:

1. Определить уровни классификации
2. Установить критерии определяющие порядок классификации данных
3. Назначить владельцев данных, которые укажут, к какому классу следует отнести данные в зоне их ответственности
4. Определить ответственных за хранение (custodian) данных, в обязанности которых входит поддержка данных и обеспечение необходимого уровня их безопасности
5. Указать средства управления, защитные меры и механизмы, необходимые для каждого уровня классификации
6. Документировать все исключения из предыдущих пунктов
7. Определить процедуры переноса ответственности за данные от одного владельца другому
8. Организовать периодический пересмотр классификации и распределения данных по владельцам (о любых изменениях необходимо уведомлять ответственных за хранение данных)
9. Определить процедуры рассекречивания данных по истечении определенного срока
10. Организовать обучение по этим вопросам всех сотрудников, чтобы они понимали, как нужно работать с данными различного уровня классификации

При разработке набора уровней классификации информации, учитывайте следующее:

- Слишком большое количество уровней классификации непрактично и увеличивает путаницу.
- Слишком малое количество уровней классификации говорит о том, что данному процессу не уделено должного внимания.
- Не должно быть никаких совпадений в критериях, определяющих различные уровни классификации
- Уровни классификации должны быть разработаны как для данных, так и для программного обеспечения.

7.1. Управление классифицированными данными

Ниже представлены некоторые универсальные рекомендации, относящиеся к критичным данным и приложениям, применимые для большинства компаний:

- Жесткий и детальный контроль доступа
- Шифрование данных при их хранении и передаче
- Аудит и мониторинг (нужно определить необходимый уровень аудита, время хранения лог-файлов)
- Разделение обязанностей (нужно определить двух или более людей, одновременное присутствие которых необходимо для получения доступа к информации)
- Периодический пересмотр уровня классификации
- Документированные процедуры резервного копирования и восстановления
- Документированные процедуры управления изменениями
- Документированные процедуры физической безопасности

- Управление информационными потоками
- Пересмотр словаря данных
- Документированные процедуры надлежащего уничтожения информации (применение shredders, размагничивание и т.п.)
- Документированное разграничение доступа на уровне файлов и файловых систем
- Пометка и маркирование
- Маркирование обложек и внутренних документов
- Все носители информации должны помечаться и маркироваться (магнитные, оптические носители информации и т.п.)

Ссылки по теме:

- Guide for Mapping Types of Information and Information Systems to Security Categories
- Handbook of Information Security Management, Domain 4, Chapter 4-1-1, “Information Classification: A Corporate Implementation Guide,” by Jim Appleyard (CRC Press LLC)

8. Уровни ответственности

Высшее руководство понимает миссию компании, цели и задачи бизнеса. Следующий уровень – это функциональное руководство (руководители департаментов, исполнительные директора), которое понимает, как работают отдельные департаменты, какое влияние на них оказывает безопасность, какую роль в компании играют люди. Следующий уровень – оперативное руководство (руководители подразделений) и штат сотрудников, которые непосредственно обеспечивают работу компании. Они детально знают технические и процедурные требования, а также системы и порядок их использования. Они понимают, как механизмы безопасности интегрированы в эти системы, как их настраивать и какое влияние они оказывают на работу. Люди, входящие в состав каждого уровня, должны быть ознакомлены с лучшими практиками по безопасности, процедурами и выбранными защитными мерами, что позволит обеспечить необходимый уровень защиты без негативного воздействия на работу.

Хотя все уровни важны для общей безопасности компании, есть специфические роли, которые должны быть явно определены. Например, это владельцы данных, ответственные за хранение данных, владельцы систем, администраторы безопасности, аналитики безопасности, владельцы приложений, супервизоры, аналитики управления изменениями, аналитики данных, владельцы процессов, поставщики решений, пользователи, менеджеры по технологиям и т.д.

8.1. Совет Директоров

Совет Директоров – это группа лиц, которые избираются акционерами компании для осуществления надзора за исполнением Устава компании. Целью Совета Директоров является обеспечение защиты интересов акционеров в процессе работы компании. Критически важно, чтобы члены Совета Директоров были беспристрастны и независимы. На протяжении многих лет, слишком много людей, занимавших эти позиции, не замечали корпоративного мошенничества и злоупотреблений, т.к. слишком многое в их работе зависит от обратной связи с исполнительным руководством, а не от попыток самостоятельно узнать правду о здоровье своей компании. Мы знаем об этом по череде корпоративных скандалов, произошедших за последние годы (Enron, WorldCom, Global Crossing и т.д.). При этом Советы Директоров этих корпораций были обязаны выявлять подобные мошеннические действия и пресекать их для защиты акционеров. Эти скандалы заставили правительство США и SEC (Комиссия по торговле ценными бумагами) разработать дополнительные

требования и штрафные санкции для Советов Директоров публичных компаний. Именно поэтому многие компании сегодня сталкиваются с трудностями при поиске кандидатов на выполнения этих ролей.

Независимость является крайне важным фактором, позволяющим членам Совета Директоров реально работать на благо акционеров. Это означает, что у них нет близких родственников, являющихся сотрудниками компании, они не получают финансовые выгоды от компании (которая могла бы повлиять на их решения или создать конфликт интересов), никакая другая деятельность не влияет на их работу, они руководствуются только интересами акционеров. Особое внимание следует обратить на это в том случае, если компания обязана соблюдать требования SOX (Sarbanes-Oxley Act), согласно которому Совет Директоров несет персональную ответственность, если компания не может поддерживать структуру внутреннего стратегического корпоративного управления и/или если в SEC предоставляется недостоверная финансовая информация.

8.2. Высшее исполнительное руководство

Это люди, названия должностей которых начинаются с английской буквы "С".

Самым высоким должностным в компании является **Генеральный директор (CEO – Chief Executive Officer)**, который выполняет повседневные управленческие функции в компании. Часто именно этот человек является председателем Правления. CEO следит за финансами компании, занимается стратегическим планированием, высокоуровневыми операциями, разрабатывает и вносит изменения в бизнес-планы компании. CEO устанавливает бюджеты, формы сотрудничества, принимает решения о том, на какие рынки нужно выходить компании, какую линию продуктов выпускать, как дифференцировать компанию и т.д. Эта роль полностью отвечает за развитие и процветание компании.

Финансовый директор (CFO – Chief Financial Officer) отвечает за счета, финансовую деятельность и всю финансовую структуру компании. Он отвечает за определение будущих финансовых потребностей компании и порядка финансирования этих потребностей. Он должен создать и поддерживать структуру капитала компании, являющегося сочетанием акций, кредитов, наличных средств и финансовой задолженности. Также, CFO контролирует процессы прогнозирования и бюджетирования, процессы представления квартальной и годовой финансовой отчетности в SEC и заинтересованным сторонам.

CFO и CEO отвечают за информирование заинтересованных лиц (кредиторов, аналитиков, сотрудников, руководства, инвесторов) о финансовом состоянии и здоровье компании.

На рисунке 1-10 показано, что члены Совета Директоров отвечают за определение стратегии компании и риск-аппетит (сколько рисков следует взять на себя компании). Совет Директоров также несет ответственность за получение информации от высшего руководства и гарантий (комитет по аудиту). Получая все это на входе, Совет Директоров должен обеспечить надлежащую работу компании и защитить интересы акционеров. Необходимо особо отметить, что именно руководители бизнес-подразделений являются владельцами рисков, а не департамент безопасности. Слишком многие компании не распространяют ответственность за риски на бизнес-подразделения, поэтому CISO часто называют жертвенным ягненком...

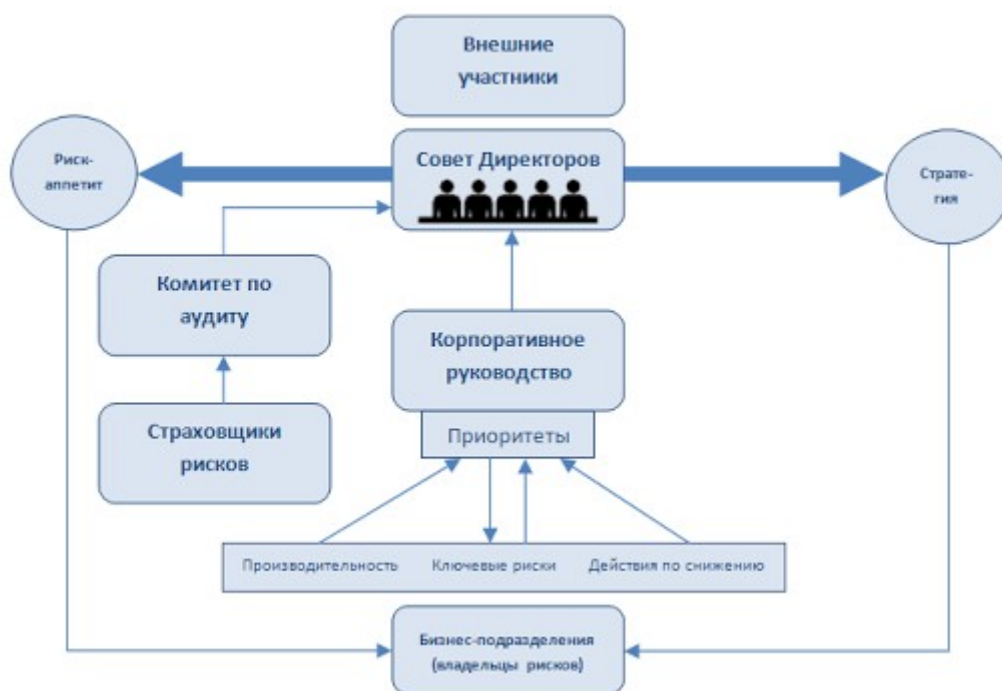


Рисунок 1-10. Риски должны понимать в различных департаментах и на различных уровнях
Директор по ИТ

На ступень ниже находится *Директор по ИТ (CIO – Chief Information Officer)*, который отчитывается перед CEO (или CFO – в зависимости от корпоративной структуры). CIO отвечает за стратегию использования и управление информационными системами и технологиями в рамках компании. Со временем, во многих компаниях эта позиция стала более стратегической и менее оперативной. CIO осуществляет контроль и несет ответственность за непрерывное функционирование ИТ компании.

В обязанности CIO входит совместная с CEO (и другими руководителями) работа над управлением бизнес-процессами, получением доходов, порядком реализации бизнес-стратегии компании с помощью имеющихся у нее технологий. Как правило CIO должен стоять одной ногой в технике, а другой – в бизнесе. Только это позволит ему быть эффективным, так как именно он является мостиком между этими двумя совершенно разными мирами.

Директор по защите конфиденциальной информации

Директор по защите конфиденциальной информации (CPO – Chief Privacy Officer) – это новая должность, созданная в основном из-за повышения требований к компаниям в части защиты большого перечня различных типов данных. CPO отвечает за обеспечение безопасного хранения данных клиентов, сотрудников и самой компании, что позволит компании избежать попадания в уголовные и гражданские суды, а также в заголовки газет. CPO, как правило, является юристом, он непосредственно участвует в разработке политики о порядке сбора, защиты конфиденциальной информации, а также ее передачи третьим сторонам. В большинстве случаев CPO отчитывается перед CSO.

Очень важно, чтобы компания знала и понимала действующие требования законодательства по защите конфиденциальной информации (в т.ч. по защите персональных данных), которым она должна соответствовать. Только это позволит разрабатывать полноценные политики, стандарты, процедуры, защитные меры, соглашения направленные на соблюдение требований по защите конфиденциальной информации. Помните также, что компания должна знать, как защищают конфиденциальную информацию ее поставщики, партнеры и другие третьи стороны. Некоторые компании проводят оценку рисков, не учитывая санкции и последствия, которые может понести компания, если не будет должным образом защищать

информацию, за которую она несет ответственность. Без этого, оценка рисков не может быть проведена должным образом.

Компания должна документировать порядок обеспечения конфиденциальности при сборе, использовании, раскрытии, архивировании и уничтожении данных. Сотрудники должны нести персональную ответственность за невыполнение этого порядка.

Международные требования. Если компания обменивается данными с европейскими компаниями и организациями, ей необходимо соблюдать требования «безопасной гавани» (safe harbor). Европа всегда придерживалась более жесткого контроля за защитой конфиденциальной информации, чем США и других части мира. В прошлом, когда американским и европейским компаниям нужно было обмениваться данными, часто возникали различные проблемы и сложности, иногда приводившие к остановке бизнеса, поскольку приходилось привлекать адвокатов, чтобы понять, как правильно работать в рамках различных законов. Чтобы устранить этот хаос, были разработаны требования «безопасной гавани», которые определяют, что должна делать любая компания для защиты конфиденциальных данных при их передаче в (или из) Европу. Американские компании, которые взаимодействуют с европейскими компаниями могут пройти сертификацию в соответствии с этими базовыми требованиями, после чего они смогут обмениваться данными быстрее и проще. Более подробную информацию об этих требованиях можно найти по адресу http://www.export.gov/safeharbor/eg_main_018236.asp.

Глобальные компании, перемещающие данные между границами различных стран, должны также знать и следовать руководящим принципам «Организации экономического сотрудничества и развития» (OECD – Organisation for Economic Co-operation and Development) и правилам трансграничной передачи информации. Почти каждая страна имеет свои собственные правила в отношении конфиденциальных данных, их защиты и обращения с ними. По мере нашего перехода в цифровую и информационную эру, различия в этих законах начали негативно влиять на бизнес и международную торговлю. OECD является международной организацией, которая помогает правительствам различных стран собраться вместе и решить экономические, социальные и управленческие задачи в условиях глобализации экономики. С этой целью OECD предоставил различным странам свои руководящие принципы, чтобы все следовали одним и тем же правилам и при этом конфиденциальные данные были надлежащим образом защищены. Более подробную информацию об этом можно найти по адресу http://www.oecd.org/document/18/0,2340,en_2649_34255_1815186_1_1_1_1,00.html.

Компании, которые не следуют этим правилам и руководящим принципам (сознательно или нет), могут быть оштрафованы, на них могут быть поданы судебные иски, их бизнес может быть остановлен.

Директор по безопасности

Директор по безопасности (CSO – Chief Security Officer) обязан понимать риски, с которыми сталкивается компания, и снижать их до приемлемого уровня, он отвечает за то, чтобы бизнес компании не был нарушен из-за проблем с безопасностью. Также, он обязан понимать бизнес-драйверы компании, создать и поддерживать программу безопасности, которая обеспечит работу этих драйверов в соответствии с требованиями безопасности, длинным списком правил и законов, ожиданиями клиентов и договорными обязательствами.

Работа CSO распространяется далеко за пределы ИТ и учитывает бизнес-процессы компании, правовые вопросы, вопросы функционирования, доходов, защиты репутации, управления рисками. Причем все это должно быть реализовано экономически эффективными способами!

CSO vs. CISO. CSO и Директор по информационной безопасности (CISO – Chief Information Security Officer) могут иметь схожие или очень разные обязанности. Это зависит от решения самой компании. В большинстве случаев CSO имеет больше обязанностей по сравнению с CISO. CISO, как правило, в большей степени сосредоточен на технологиях и ИТ-инфраструктуре, а CSO обязан работать с более широким кругом бизнес-рисков, включая физическую безопасность, а не только технологические риски. Если компания имеет обе роли, обычно CISO отчитывается перед CSO.

Руководящий комитет по безопасности

Руководящий комитет по безопасности отвечает за принятие решений по стратегическим

и тактическим вопросам безопасности в рамках всей компании. Этот Комитет должен состоять из представителей всех подразделений компании, чтобы они могли учитывать риски и последствия решений по безопасности, как для отдельных департаментов, так и для всей компании в целом. Возглавлять этот Комитет должен CEO, а участие в нем должны принимать CFO, CIO, руководители подразделений, а также руководитель Службы внутреннего контроля (CIA – Chief Internal Auditor). Комитет должен собираться не реже, чем ежеквартально, и иметь четкую повестку дня. Вот некоторые из обязанностей Комитета:

- Определение приемлемого уровня риска для компании
- Разработка целей и стратегий безопасности
- Определение приоритетных инициатив в области безопасности, основанных на потребностях бизнеса
- Анализ оценки рисков и аудиторских отчетов
- Мониторинг воздействия рисков безопасности на бизнес
- Анализ основных нарушений безопасности и инцидентов
- Утверждение любых существенных изменений в политике и программе безопасности.

Комитет по аудиту

Комитет по аудиту должен быть организован Советом Директоров. Целью Комитета по аудиту является помощь Совету Директоров в вопросах анализа и оценки внутренних процессов компании, системы внутреннего контроля, а также прозрачности и достоверности финансовой отчетности, обеспечивающей доверие к компании со стороны инвесторов, клиентов и кредиторов. Комитет по аудиту, как правило, отвечает за следующие элементы:

- Целостность финансовой отчетности и другой финансовой информации компании, предоставляемой акционерам и другим лицам
- Система внутреннего контроля компании
- Привлечение независимых аудиторов и обеспечение эффективности их работы
- Выполнение функции внутреннего аудита
- Соответствие требованиям законодательства и политикам компании в отношении норм этического поведения.

Задача этого Комитета состоит в том, чтобы обеспечить независимую и открытую связь между Советом Директоров, высшим руководством компании, внутренними и внешними аудиторами. Целостность и надежность финансовой отчетности имеет решающее значение для каждой компании, но возможное давление со стороны акционеров, руководства, инвесторов или общественности может повлиять на объективность и точность этих финансовых документов. В свете громких корпоративных скандалов, роль Комитета по аудиту сместилась от функций исключительно надзора, контроля и консультирования руководства к реализации и обеспечению подотчетности всех вовлеченных лиц. Этот Комитет принимает данные от внешних и внутренних аудиторов, внешних экспертов, чтобы помочь обеспечить правильную организацию внутреннего контроля компании и подготовки финансовой отчетности.

8.3. Владелец данных

Владелец данных (информации) (data owner) – обычно руководитель соответствующего подразделения, который несет персональную ответственность за защиту и использование определенного подмножества информации. Владелец данных проявляет необходимую заботу об информации и несет ответственность за любые халатные действия, искажение или

разглашение информации. Владелец данных принимает решения по классификации информации и обязан своевременно пересматривать уровень классификации в случае изменения потребностей бизнеса. Также он должен обеспечить внедрение необходимых мер безопасности, убедиться в правильном разграничении прав доступа, сформулировать требования безопасности в соответствии с классификацией, а также требования по организации резервного копирования информации, согласовывать любые действия по санкционированному разглашению (передаче) информации, определять критерии доступа к информации. Владелец данных согласует запросы на доступ к информации, но может делегировать эту функцию одному из нижестоящих руководителей своего подразделения. Владелец данных явно делегирует обязанности по ежедневной поддержке механизмов защиты данных ответственному за хранение данных.

Задачи Владельца данных. Каждое бизнес-подразделение должно иметь владельца данных для защиты критичной информации этого подразделения. Политики компании должны давать владельцам данных определенную власть, необходимую им для выполнения своих обязанностей.

Владелец данных – это не техническая роль. Это бизнес-роль, которая должна понимать взаимосвязь между успехом подразделения и защитой критичных активов. Не все руководители понимают это, поэтому может возникнуть необходимость в проведении тренинга для них по данному вопросу.

8.4. Ответственный за хранение данных

Ответственный за хранение данных (информации) (data custodian) – это лицо, ответственное за поддержку и защиту данных. Эта роль обычно выполняется ИТ-департаментом или департаментом безопасности, в ее обязанности входит выполнение регулярного резервного копирования, периодической проверки целостности, восстановления данных, сохранения лог-файлов, выполнение требований политики безопасности компании, стандартов, руководств, относящихся к информационной безопасности и защите данных.

8.5. Владелец системы

Владелец системы (system owner) отвечает за одну или несколько систем, в каждой из которых хранятся и обрабатываются данные, принадлежащие различным владельцам данных. Владелец системы отвечает за соответствие покупаемых и разрабатываемых систем требованиям безопасности, а также за соблюдение требований безопасности в самих процессах закупки или разработки систем. Владелец системы отвечает за надлежащий уровень ее безопасности, обеспечиваемый защитными мерами, паролями, настройками и т.д. Эта роль необходима при оценке уязвимостей; лицо, выполняющее эту роль, отчитывается перед группой реагирования на инциденты и владельцем данных.

8.6. Администратор безопасности

Любой, кто имеет административную учетную запись, фактически имеет права администратора безопасности (к сожалению, зачастую слишком большое количество людей обладают такими правами). Это означает, что он может предоставлять и отзывать любые разрешения, изменять настройки безопасности, может серьезно навредить, если у него выдался плохой день...

Однако роль администратора безопасности не может выполнять просто человек, имеющий административные права. У **администратора безопасности (security administrator)** много задач. Например, создание новых системных учетных записей, внедрение новых программных средств безопасности, тестирование обновлений и компонентов безопасности, выпуск новых паролей и т.д. Администратор безопасности не должен просто подтверждать новые учетные записи – это задача супервизора. Администратор безопасности должен обеспечить распределение прав доступа в соответствии с требованиями политик безопасности и распоряжений владельца данных.

8.7. Аналитик по безопасности

Роль *аналитика по безопасности (security analyst)* работает на более высоком и более стратегическом уровне, чем описанные выше роли. Данная роль оказывает помощь при разработке политик, стандартов, руководств и наборов базисов. Предыдущие роли фокусируются на своих частях программы безопасности, а аналитик по безопасности помогает определить элементы программы безопасности в целом, обеспечить их внедрение и правильное функционирование. Человек, выполняющий эту роль, больше работает на уровне планирования, чем на уровне реализации.

8.8. Владелец приложения

Некоторые приложения специально предназначены для отдельных подразделений (например, бухгалтерское программное обеспечение – для бухгалтерии). **Владельцами приложений (application owner)** обычно являются руководители подразделений, определяющие права доступа к их приложениям (в соответствии с политиками безопасности) и принимающие соответствующие решения в конкретных случаях.

В каждом подразделении должен быть владелец приложений подразделения, который отвечает за безопасность этих приложений, включая тестирование, установку обновлений, управление изменениями, обеспечение внедрения соответствующих защитных мер, обеспечение необходимого уровня безопасности.

8.9. Супервизор

Роль *супервизора (supervisor)* несет персональную ответственность за все действия пользователей и любые активы, которые создали пользователи и которыми они владеют. Супервизор обязан обеспечивать понимание пользователями (входящими в его зону ответственности) их обязанностей в части безопасности, выдавать им первоначальные пароли, актуализировать информацию учетных записей пользователей, уведомлять администратора безопасности об увольнении сотрудников, их временном отсутствии, переводе в другие подразделения. Любые изменения в ролях сотрудников компании обычно влияют на то, какие права им нужны для работы. О таких изменениях супервизор должен незамедлительно информировать администратора безопасности.

8.10. Аналитик управления изменениями

Когда требуются изменения, кто-то должен убедиться, что это безопасно. **Аналитик управления изменениями (change control analyst)** анализирует запросы на проведение изменений в сети, программном обеспечении или системах, после чего одобряет, либо отвергает запрашиваемые изменения. Эта роль обеспечивает отсутствие новых уязвимостей при внедрении изменений, обеспечивает проведение необходимого тестирования, а также возможность корректного «отката» к старой версии. Аналитик управления изменениями должен понимать, как различные изменения влияют на безопасность, совместимость, производительность и продуктивность.

8.11. Аналитик данных

Для компании очень важно иметь правильные структуры данных, их определения и организацию. **Аналитик данных (data analyst)** отвечает за организацию наилучшего (для компании и пользователей) хранения данных. Например, чтобы информация о платежах не перемешивалась с данными инвентаризации, базы данных имели удобную схему имен и т.д. Аналитик данных отвечает за проектирование архитектуры новых систем, либо предоставляет рекомендации для покупки систем.

Аналитик данных постоянно взаимодействует с владельцами данных, чтобы обеспечить соответствие структуры данных бизнес-целям компании.

8.12. Владелец процесса

Все компании имеют множество процессов (например, оформление заказов клиентов, проведение платежей, отправка товара и т.д.), они не могут функционировать без правильно построенных процессов. **Владелец процесса (process owner)** отвечает за правильное определение, повышение качества и мониторинг процессов компании. Владелец процесса не обязательно должен быть «привязан» к отдельному подразделению или приложению. Большинство процессов являются достаточно сложными, в их реализацию вовлечены различные подразделения и сотрудники компании, различные виды данных, технологии и т. д.

Вы слышали фразу, что «безопасность – это не продукт, а процесс»? Это действительно так. Безопасность должна быть тщательно продуманной и также как и любой бизнес-процесс.

8.13. Поставщик решения

Роль **поставщика решения (solution provider)** требуется, когда у компании возникает проблема или требуется улучшение какого-либо процесса. Например, если компания решила внедрить инфраструктуру открытых ключей и организовать аутентификацию с их использованием, она обращается к поставщику решений PKI. Поставщик решения работает с руководителями подразделений, владельцами данных и высшим руководством компании для разработки и внедрения своего решения.

8.14. Пользователь

Пользователь (user) – любой человек, который санкционировано использует данные, имея к ним уровень доступа, достаточный для выполнения своих должностных обязанностей. Пользователь несет ответственность за соблюдение процедур безопасности, обеспечивающих конфиденциальность, целостность данных и их доступность для других.

8.15. Менеджер по технологиям

Менеджер по технологиям (product line manager) объясняет бизнес-требования поставщикам, оценивает – подходит ли продукт компании, отвечает за соблюдение лицензионных требований, переводит бизнес-требования в задачи и спецификации для разработчиков продуктов и решений. Он решает, например, действительно ли компании необходимо перейти на новую версию операционной системы.

Эта роль должна понимать подходы бизнеса, бизнес-процессы и необходимые для их функционирования технологии. Менеджер по технологиям оценивает различные продукты на рынке, взаимодействует с поставщиками и производителями, анализирует различные функции, которые могли бы пригодиться компании, и рекомендует руководству и подразделениям решения, подходящие для достижения их целей.

8.16. Аудитор

Задачей аудитора является предоставление гарантированной независимости, на которую может положиться руководство и акционеры компании в вопросах оценки адекватности задач безопасности с учетом текущего состояния компании в целом. Аудитор должен проконтролировать внедрение защитных мер, достижение определенных технических и физических характеристик, реализацию целей безопасности, поставленных требованиями действующего законодательства или требованиями руководства самой компании.

Аудиты компании могут быть как внутренними, так и внешними. Их сочетание, как правило, предоставляет наиболее полную и объективную оценку деятельности компании.

Самым большим вопросом, вызывающим наибольшее беспокойство в отношении аудиторов, является соблюдение ими непредвзятости и объективности. Снизить это

беспокойство отчасти может проведение обзоров какой-либо третьей стороной. В некоторых случаях существуют законодательные требования, которые не позволяют даже сторонним аудиторам работать на протяжении многих лет подряд в одной компании, так как это может привести к их слишком близким контактам с компанией, и тем самым подорвать уверенность в объективности их оценок.

8.17. Зачем так много ролей?!

К сожалению, очень часто компании пытаются решать все вопросы безопасности на уровне системного администратора. При этом вопросы безопасности, как правило, рассматриваются достаточно узко, риски не анализируются. Средства на безопасность выделяются только после очередного инцидента. Такой подход к безопасности не может быть успешным.

Безопасность компании – это не только установленный межсетевой экран и обновления операционной системы. Компьютерная среда компании наполнена различными ресурсами, процессами, людьми. Необходимо обеспечить целостную защиту этой среды, каждый аспект безопасности требует учета и взвешенного подхода. Хотя большинство компаний не имеет всех перечисленных ранее ролей, все обязанности этих ролей должны выполняться.

Ссылки по теме:

- Security School for CISSP Training: Domain Spotlight on Security Management Practices
- NIST Administrative Manual, Subchapter 11.02, “Information Technology Security”
- Information Security Management Handbook, Domain 3, Chapter 26, “Information Protection: Organization, Roles, and Separation of Duties,” by Rebecca Herold (CRC Press LLC)

9. Персонал

Многие обязанности персонала имеют прямое отношение к безопасности компании. Хотя общество стало чрезвычайно зависимым от технологий, люди остаются ключевым фактором успеха компании. В безопасности чаще всего именно человек является слабым звеном. Это является следствием человеческих ошибок, недостатков в обучении и приводит к мошенничеству, несанкционированным или небезопасным действиям. Человек во многих случаях является причиной успеха хакерских атак, шпионажа, повреждения оборудования. Хотя будущие действия людей нельзя предсказать, можно внедрить определенные превентивные меры, которые помогут минимизировать риски. Такими превентивными мерами могут быть следующие: прием на работу только квалифицированного персонала, контрольные мероприятия, детальные должностные инструкции, обучение персонала, строгий контроль доступа, обеспечение безопасности при увольнении сотрудников.

9.1. Структура

Если компания хочет иметь эффективную безопасность на уровне персонала, руководство должно внедрить четкую структуру и следовать ей. Эта структура включает в себя четкое описание обязанностей, разграничение полномочий, ответственность (например, объявление выговоров) за определенные действия. Четкая структура исключает непонимание, кто и что должен делать в различных ситуациях.

Есть несколько аспектов, которые должны быть внедрены для снижения возможностей для мошенничества, саботажа, краж, неправильного использования информации, а также уменьшения вероятности возникновения других проблем безопасности. Одним из таких аспектов является *разделение обязанностей (separation of duties)*, которое гарантирует, что один человек не сможет самостоятельно выполнить критичную задачу. Разделение обязанностей также снижает вероятность ошибок – если один человек ошибся, другой, скорее всего, увидит ее и исправит. Разделение обязанностей снижает риски мошенничества, так как сотрудник не может выполнить самостоятельно критичную операцию и ему

необходимо вступить в сговор с другим сотрудником, а вероятность сговора двух и более людей существенно ниже вероятности действий отдельного человека.

При разработке программного обеспечения должно быть явное разделение между средой разработки, средой тестирования, библиотекой программного обеспечения и производственной средой. Программистам должна быть доступна только среда разработки, в которой они могут разрабатывать программное обеспечение и производить его предварительное тестирование. После разработки исходные тексты передаются другому человеку, который проводит контроль качества кода и тестирование его работы в отдельной среде, являющейся копией производственной среды. Только когда код прошел все необходимые тесты, он размещается в библиотеке программного обеспечения. Для внедрения в производственную среду, программное обеспечение берется только из библиотеки. Код ни в коем случае не должен попадать напрямую от программиста в производственную среду без тестирования и сохранения в библиотеке! Тестовая среда должна быть полностью отделена от производственной, чтобы непроверенное еще программное обеспечение не нанесло вреда данным и системам, находящимся в реальной работе. Программист не должен «латать» свои программы непосредственно в процессе их эксплуатации! Должны быть внедрены простые и эффективные методы, не позволяющие вносить изменения в программное обеспечение компании небезопасными способами.

9.2. Правила приема на работу

В зависимости от позиции, на которую компания ищет нового сотрудника, должен выбираться уровень фильтрации кандидатов, проводимой специалистами кадровой службы компании, чтобы найти именно того человека, который лучше всего подходит к условиям этой позиции. Сотрудники – это своего рода инвестиции компании, поэтому тратя дополнительное время на поиск и найм нужных сотрудников, компания может увеличить возврат своих инвестиций.

Следует тестировать и оценивать навыки, особенности характера кандидатов, проверять прошлое кандидатов, их рекомендации, образование, данные военного учета и т.д. Во многих случаях важные черты поведения и характера кандидата могут быть не заметны, поэтому следует готовить специальные вопросы, персональные тесты, наблюдать за кандидатом, а не просто читать его резюме.

Более детальная проверка информации о кандидате может выявить массу интересных сведений. Например, необъяснимые пробелы в трудовом стаже, различия в заявленной и фактической квалификации, судимости, штрафы за нарушение правил вождения автомобиля, искажение названий компаний-работодателей, плохая кредитная история, увольнения не по собственному желанию, нахождение в различных криминальных списках, реальные основания увольнения с предыдущих мест работы. Это дает реальную выгоду компании, т.к. является, по сути, первой линией обороны для защиты от нападения изнутри. Любая негативная информация, выявленная на данном этапе, может указывать на потенциальные проблемы, которые потенциальный сотрудник может создать для компании позднее. Например, не стоит принимать на работу в бухгалтерию или кассу человека с плохой кредитной историей.

Такая проверка преследует сразу несколько целей – это снижение рисков, уменьшение расходов по найму, минимизация текучести кадров в компании. Важно провести эту проверку до приема сотрудника в компанию, т.к. провести проверку уже работающего сотрудника гораздо сложнее – для этого нужны конкретные и веские причины. Такой причиной может быть, в частности, переход сотрудника на более критичную должность (с точки зрения безопасности компании).

Возможные критерии проверки могут включать:

- Отслеживание Номера карточки социального страхования

- Проверка по криминальным спискам города
- Проверка по федеральной криминальной базе
- Проверка на наличие в реестре сексуальных преступников
- Проверка предыдущей работы
- Проверка образования
- Проверка профессиональных отзывов
- Иммиграционный контроль

Дополнительный контроль для позиций высокого уровня или критичных позиций может включать следующее:

- Проверка профессиональных лицензий/сертификатов
- Отчет по кредитной истории
- Проверка на наркотики

Новым сотрудником должно быть подписано соглашение о неразглашении конфиденциальной информации. Должны быть учтены любые возможные конфликты интересов и при необходимости с новым сотрудником должны быть заключены дополнительные соглашения. Аналогичные требования следует соблюдать и в отношении временных сотрудников.

9.3. Контроль сотрудников

Структура менеджмента в компании должна быть построена таким образом, чтобы каждый сотрудник и руководитель имел вышестоящего руководителя, перед которым он должен отчитываться о своей работе, а также, чтобы руководители могли контролировать равномерное и разумное распределение обязанностей между своими подчиненными. Последствия несоблюдения требований и неприемлемого поведения должны быть доведены до сведения сотрудников до того, как это произойдет. Руководители должны обладать соответствующими навыками, позволяющими обеспечить эффективный контроль и выявлять необычные действия на раннем этапе, пока ситуация не вышла из под контроля.

Ротация обязанностей (rotation of duties) позволяет сохранить здоровую и продуктивную работу подразделений. Один человек не должен оставаться на одной и той же должности длительное время, так как в конечном итоге он может получить слишком большой контроль над отдельным сегментом бизнеса. А такой уровень контроля может стать причиной мошенничества, искажения данных, ненадлежащего использования ресурсов.

Сотрудники, работающие в критичных для компании областях, обязательно должны периодически брать отпуск (это известно как **политика обязательного отпуска (mandatory vacation policy)**). Во время их отпуска другие сотрудники должны исполнять их обязанности, что позволит выявить возможные ошибки и мошеннические действия. Должны обязательно контролироваться и расследоваться два варианта сетевых аномалий: использование учетных записей сотрудников, отсутствующих на рабочем месте (например, находящихся в отпуске), а также прекращение определенных проблем при отсутствии кого-либо на рабочем месте.

Существует два варианта разделения обязанностей – **разделение знаний (split knowledge)** и **двойное управление (dual control)**. В обоих случаях для выполнения операции требуется одновременное участие двух или более сотрудников, имеющих соответствующие полномочия. В случае разделения знаний ни один из сотрудников не знает всех деталей выполняемой задачи (например, чтобы открыть банковское хранилище два ответственных сотрудника должны ввести свою часть кода в электронный кодовый замок). В случае двойного управления также требуются два уполномоченных сотрудника, но каждый из них

выполняет свою часть задачи (или миссии). Например, для запуска ракеты два офицера должны одновременно и независимо друг от друга вставить и повернуть ключ. При этом обеспечивается невозможность выполнения этих операций одним человеком.

9.4. Увольнение

Увольнение может происходить по различным причинам, и, соответственно, оно может приводить к различной реакции увольняемых. Компания должна иметь специальные процедуры увольнения, безусловно применяемые во всех случаях увольнения сотрудников и руководителей. Например:

- Сотрудник должен покинуть здание немедленно в сопровождении руководителя или охранника;
- Сотрудник должен сдать все идентификационные бейджи или ключи, пройти специальное интервью, вернуть имущество компании;
- Учетные записи и пароли сотрудника должны быть немедленно заблокированы или изменены.

Это кажется излишне жестким и бессердечным, но статистика говорит о том, что слишком большое число компаний пострадали от мести уволенных сотрудников.

10. Обучение (тренинги) по вопросам безопасности

Директивы руководства, имеющие отношение к безопасности, выражаются в виде политики безопасности, а также стандартов, процедур и руководств, разрабатываемых для поддержки этих директив. Однако эти директивы не будут эффективны, если никто не знает о них и не соблюдает. Чтобы безопасность была эффективной, все сотрудники без исключений, включая высшее руководство, должны быть осведомлены о важности обеспечения безопасности компании. Все сотрудники должны знать относящиеся к ним требования безопасности и понимать смысл обеспечения безопасности компании в целом.

Защитные меры и процедуры программы безопасности должны соответствовать характеру деятельности компании и обрабатываемым данным. Безопасность компании, которая продает прохладительные напитки, сильно отличается от безопасности компании, производящей ракеты. Разные типы компаний могут иметь очень сильно отличающуюся корпоративную культуру. Поэтому, чтобы программа обучения по вопросам безопасности была эффективной, она должна быть понятной, учитывающей характер деятельности и особенности корпоративной культуры компании.

Обучение должно быть всесторонним, программа обучения должна быть сформирована с учетом особенностей отдельных обучаемых групп и всей компании в целом. Целью обучения является достижение понимания каждым сотрудником важности безопасности, как для всей компании, так и для него лично. В процессе обучения должны, в частности, рассматриваться вопросы об обязанностях сотрудников, допустимом поведении, ответственности за несоблюдение требований (от уведомления до увольнения) и т.д.

10.1. Различные типы обучения (тренинга) по вопросам безопасности

При проведении обучения по вопросам безопасности, слушателей обычно разделяют на три категории: руководство, персонал и сотрудники ИТ-подразделений. Программы обучения должны быть подготовлены с учетом особенностей каждой категории, в том числе их обязанностей, ответственности и ожиданий.

Высшее руководство ценит краткость, ориентацию на корпоративные активы, финансовые выгоды и потери. Руководители должны понять, как повлияют угрозы и их последствия на стоимость акций компании, какие механизмы безопасности должны быть интегрированы в бизнес-процессы и подразделения, которые они курируют, а также в их собственную

деятельность. Руководители должны понимать важность этих задач, так как именно они будут обеспечивать поддержку безопасности в деятельности подконтрольных им подразделений.

Руководители среднего звена должны получить более детальные объяснения политик, процедур, стандартов и руководств, также они должны понять, как это проецируется на деятельность их подразделений. Руководители должны хорошо понимать, зачем безопасности нужна их поддержка и каков уровень их ответственности за безопасную работу подчиненных им сотрудников. Им должно быть показано, как последствия невыполнения требований безопасности сотрудниками их подразделений влияют на компанию в целом и как они, руководители, будут за это отвечать.

ИТ-подразделения должны получить различные презентации в соответствии с выполняемыми ими задачами. Они должны получить более детальную информацию, чтобы обсудить технические настройки, обработку инцидентов, понять различные типы недостатков безопасности, которые они должны выявлять и предотвращать.

Презентации, предназначенные для остального персонала, должны показать важность обеспечения безопасности для компании в целом и для каждого сотрудника в частности. Лучше всего, если сотрудники поймут, как небезопасные действия могут негативно отразиться на них самих, и какое участие они должны принимать в предотвращении таких действий. Презентации должны иметь много примеров приемлемых и неприемлемых действий. Примеры следует приводить из областей, связанных с повседневной работой сотрудников компании, например, таких, как работа с электронной почтой, сетью Интернет, работа с конфиденциальной информацией и т.п. Сотрудники должны понять, что требуется от них самих и чем им грозит несоблюдение требований безопасности. Желательно по результатам обучения получить подпись каждого сотрудника об ознакомлении с требованиями и понимании ответственности за их несоблюдение. Это повысит осознание сотрудниками важности вопросов безопасности, а в случае инцидентов не позволит им уйти от ответственности, заявив, что им никто не говорил о требованиях безопасности.

Каждая группа должна понять, кому сообщать о подозрительной деятельности и как действовать в таких ситуациях. Сотрудникам должно быть объяснено, что им самим не нужно пытаться бороться с атакующими или противодействовать мошенническим действиям. В случае выявления таких фактов, они должны уведомить об этом своего непосредственного руководителя, который должен определить, как действовать в этой ситуации.

Обучение (тренинги) по вопросам безопасности должно периодически повторяться. Оптимальная периодичность – раз в год. Основная задача данного обучения – добиться понимания сотрудниками важности обеспечения безопасности, изменить их поведение и отношение к безопасности.

Для повышения осведомленности по вопросам безопасности можно использовать множество различных методов. Например, помимо традиционного обучения (тренингов) можно использовать такие вещи, как баннеры, плакаты, пособия работника, которые будут напоминать сотрудникам об их обязанностях и правильных подходах к обеспечению безопасности. Также, важно доводить до сведения сотрудников информацию об инцидентах, рассказывать им о том, как им не стать жертвой вредоносного программного обеспечения, социальной инженерии и других опасностей.

10.2. Оценка результатов обучения

Обучение по вопросам безопасности – это один из видов защитных мер. Поэтому, как и для любых других защитных мер, его выполнение должно отслеживаться, а эффективность – оцениваться. Нет смысла тратить деньги на то, что не работает, и нет смысла улучшать что-то, что не нуждается в улучшении. После обучения, компания может раздать сотрудникам

анкеты, провести оценку знаний сотрудников, получить обратную связь и оценить, таким образом, эффективность программы обучения. Нужно также учитывать, что в компании наверняка найдется несколько сотрудников, которые будут сопротивляться этому, считая, что им навязывают нечто, совершенно им не нужное. С такими сотрудниками необходимо проводить отдельную работу, чтобы добиться правильного понимания ими значимости обеспечения безопасности.

Хорошим индикатором эффективности обучения является сравнение количества сообщений об инцидентах, связанных с безопасностью, «до» и «после» обучения. Если количество сообщений сотрудниками об инцидентах возросло, значит время на их обучение было потрачено не зря.

При проведении обучения рекомендуется повторять наиболее важные аспекты по несколько раз в разных формах и контекстах, приводить актуальные примеры, объяснять позитивным, простым для понимания языком. И самое главное – процесс обучения по вопросам безопасности должен полностью поддерживаться высшим руководством компании. Руководство должно распределить ресурсы для выполнения этой работы и обеспечить присутствие сотрудников.

При проведении дистанционного обучения, следует контролировать прохождение обучения каждым сотрудником. В должностные обязанности сотрудников должны быть внесены пункты о необходимости прохождения данного обучения.

10.3. Специализированное обучение по безопасности

Сегодня компании тратят огромные средства на устройства и технологии безопасности, однако часто забывают, что ими нужно уметь пользоваться. Без проведения специализированного обучения персонала, деньги могут быть потрачены зря, и компания останется незащищенной. Многие понимают, что человек – самое слабое звено в безопасности, однако не тратят усилий на обучение этих людей.

Различные роли требуют различных видов обучения (администрирование межсетевых экранов, управление рисками, обслуживание систем IDS и т.п.). Квалифицированный персонал – это один из важнейших компонентов безопасности компании.

	Повышение осведомленности	Тренинги	Обучение
Атрибут	"Что"	"Как"	"Почему"
Уровень	Информация	Знания	Глубокие знания
Цель обучения	Ознакомление	Навыки	Понимание
Пример методик обучения	Видео, почтовые рассылки, плакаты	Лекции и/или демонстрации, изучение конкретных примеров, практика	Семинары и дискуссии, чтение и изучение, исследования
Средства проверки	Ответы да/нет или выбор ответов из нескольких вариантов	Решение задачи (проблемы)	Тест
Время обучения	Короткое	Среднее	Длительное

Таблица 1-8. Аспекты повышения осведомленности, тренингов и обучения

Ссылки по теме:

- Free practice quizzes
- CISSP and SSCP Open Study Guides
Information Technology Security Training Requirements: A Role- and Performance-Based Model, Dorothea de Zafra et al., NIST Special Publication 800-16 (April 1998)
- CSRC Awareness, Training, and Education links

11. Резюме

Программа безопасности должна учитывать вопросы со стратегической, тактической и

оперативной точки зрения, как показано на рисунке 1-11. Управление безопасностью включает в себя административные и процедурные мероприятия, необходимые для поддержки и защиты информации, а также активов в рамках компании в целом. Кроме того, управление безопасностью включает в себя разработку и внедрение политик безопасности и их вспомогательных механизмов: процедур, стандартов, базисов и руководств. Также, оно включает в себя управление рисками, обучение по вопросам безопасности, выбор и внедрение эффективных контрмер. Деятельность, связанная с персоналом (прием на работу, увольнение, обучение и структура управления) и его функциями (ротация и разделение обязанностей), должна проводиться надлежащим образом в целях создания безопасных условий. Руководство должно уважать и соблюдать необходимые правовые и этические нормы.

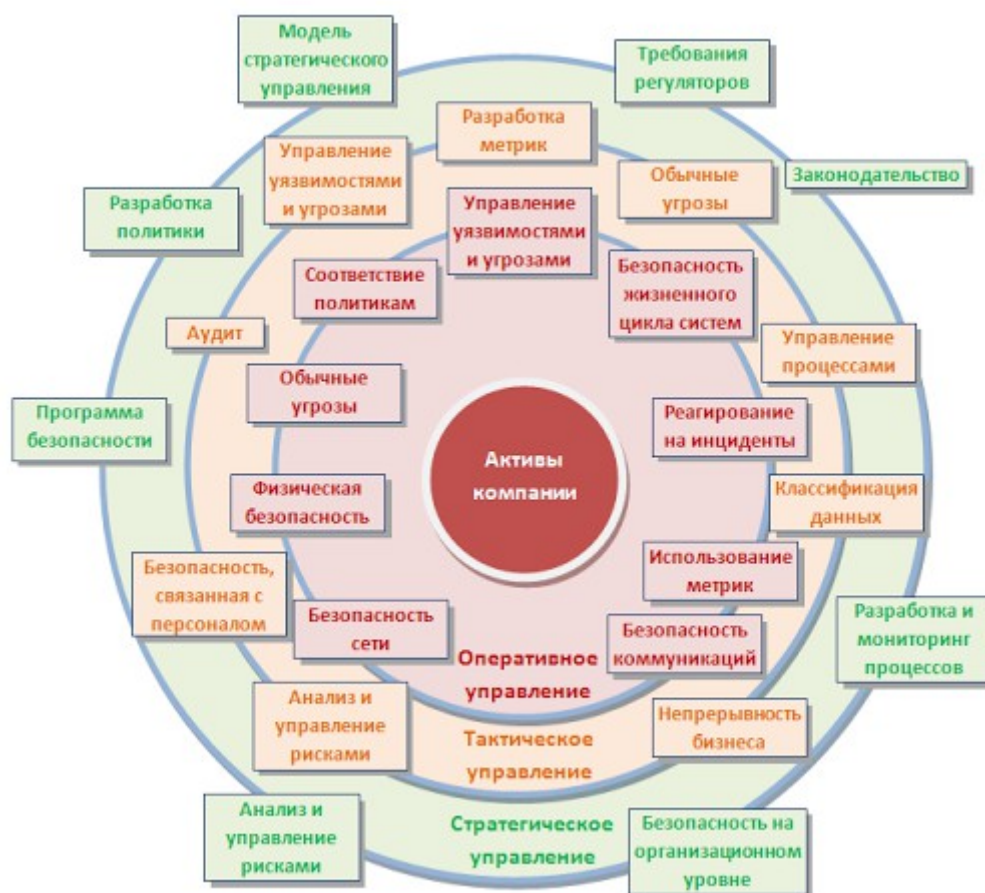


Рисунок 1-11. Полная программа безопасности состоит из множества элементов

Безопасность – это задача бизнеса, она должна рассматриваться именно в этом качестве. Она должна быть интегрирована в бизнес-цели и задачи компании, так как вопросы безопасности могут негативно сказаться на тех ресурсах, от которых зависит компания. Все большее и большее количество компаний, не уделявших должного внимания, поддержки и финансирования безопасности, несут колоссальные убытки. Хотя мы живем в прекрасном и удивительном мире, в нем может случиться всякое. Те, кто понимает это, могут не только выживать, но и процветать.

Тест

Вопросы экзамена CISSP являются концептуальными, поэтому они сформулированы соответствующим образом. Задачей кандидата является выбор наилучшего из всех представленных вариантов ответа. Среди вариантов ответа может не быть идеального ответа на поставленный вопрос - кандидат должен выбрать лучший ответ из имеющихся вариантов.

1. Кто является основным ответственным за определение уровня классификации информации?

- ☐ А. Руководитель среднего звена
- ☐ В. Высшее руководство

- ☐ C. Владелец
- ☐ D. Пользователь

2. Какая категория является наиболее рискованной для компании с точки зрения вероятного мошенничества и нарушения безопасности?

- ☐ A. Сотрудники
- ☐ B. Хакеры
- ☐ C. Атакующие
- ☐ D. Контрагенты (лица, работающие по договору)

3. Если различным группам пользователей с различным уровнем доступа требуется доступ к одной и той же информации, какое из указанных ниже действий следует предпринять руководству?

- ☐ A. Снизить уровень безопасности этой информации для обеспечения ее доступности и удобства использования
- ☐ B. Требовать подписания специального разрешения каждый раз, когда человеку требуется доступ к этой информации
- ☐ C. Улучшить контроль за безопасностью этой информации
- ☐ D. Снизить уровень классификации этой информации

4. Что самое главное должно продумать руководство при классификации данных?

- ☐ A. Типы сотрудников, контрагентов и клиентов, которые будут иметь доступ к данным
- ☐ B. Необходимый уровень доступности, целостности и конфиденциальности
- ☐ C. Оценить уровень риска и отменить контрмеры
- ☐ D. Управление доступом, которое должно защищать данные

5. Кто в конечном счете несет ответственность за гарантии того, что данные классифицированы и защищены?

- ☐ A. Владельцы данных
- ☐ B. Пользователи
- ☐ C. Администраторы
- ☐ D. Руководство

6. Что такое процедура?

- ☐ A. Правила использования программного и аппаратного обеспечения в компании
- ☐ B. Пошаговая инструкция по выполнению задачи
- ☐ C. Руководство по действиям в ситуациях, связанных с безопасностью, но не описанных в стандартах
- ☐ D. Обязательные действия

7. Какой фактор наиболее важен для того, чтобы быть уверенным в успешном обеспечении безопасности в компании?

- ☐ A. Поддержка высшего руководства
- ☐ B. Эффективные защитные меры и методы их внедрения
- ☐ C. Актуальные и адекватные политики и процедуры безопасности
- ☐ D. Проведение тренингов по безопасности для всех сотрудников

8. Когда целесообразно не предпринимать никаких действий в отношении выявленных рисков?

- ☐ A. Никогда. Для обеспечения хорошей безопасности нужно учитывать и снижать все риски
- ☐ B. Когда риски не могут быть приняты во внимание по политическим соображениям
- ☐ C. Когда необходимые защитные меры слишком сложны
- ☐ D. Когда стоимость контрмер превышает ценность актива и потенциальные потери

9. Что такое политики безопасности?

- ☐ A. Пошаговые инструкции по выполнению задач безопасности
- ☐ B. Общие руководящие требования по достижению определенного уровня безопасности
- ☐ C. Широкие, высокоуровневые заявления руководства
- ☐ D. Детализированные документы по обработке инцидентов безопасности

10. Какая из приведенных техник является самой важной при выборе конкретных защитных мер?

- ☐ A. Анализ рисков
- ☐ B. Анализ затрат / выгоды
- ☐ C. Результаты ALE
- ☐ D. Выявление уязвимостей и угроз, являющихся причиной риска

11. Что лучше всего описывает цель расчета ALE?

- ☐ A. Количественно оценить уровень безопасности среды
- ☐ B. Оценить возможные потери для каждой контрмеры
- ☐ C. Количественно оценить затраты / выгоды
- ☐ D. Оценить потенциальные потери от угрозы в год

12. Тактическое планирование – это:

- ☐ A. Среднесрочное планирование
- ☐ B. Долгосрочное планирование
- ☐ C. Ежедневное планирование
- ☐ D. Планирование на 6 месяцев

13. Что является определением воздействия (exposure) на безопасность?

- ☐ A. Нечто, приводящее к ущербу от угрозы
- ☐ B. Любая потенциальная опасность для информации или систем
- ☐ C. Любой недостаток или отсутствие информационной безопасности
- ☐ D. Потенциальные потери от угрозы

14. Эффективная программа безопасности требует сбалансированного применения:

- ☐ A. Технических и нетехнических методов
- ☐ B. Контрмер и защитных механизмов
- ☐ C. Физической безопасности и технических средств защиты
- ☐ D. Процедур безопасности и шифрования

15. Функциональность безопасности определяет ожидаемую работу механизмов безопасности, а гарантии определяют:

- ☐ A. Внедрение управления механизмами безопасности
- ☐ B. Классификацию данных после внедрения механизмов безопасности
- ☐ C. Уровень доверия, обеспечиваемый механизмом безопасности
- ☐ D. Соотношение затрат / выгод

16. Какое утверждение является правильным, если взглянуть на разницу в целях безопасности для коммерческой и военной организации?

- ☐ A. Только военные имеют настоящую безопасность
- ☐ B. Коммерческая компания обычно больше заботится о целостности и доступности данных, а военные – о конфиденциальности
- ☐ C. Военным требуется больший уровень безопасности, т.к. их риски существенно выше
- ☐ D. Коммерческая компания обычно больше заботится о доступности и конфиденциальности данных, а военные – о целостности

17. Как рассчитать остаточный риск?

- ☐ A. Угрозы x Риски x Ценность актива
- ☐ B. (Угрозы x Ценность актива x Уязвимости) x Риски
- ☐ C. $SLE \times Частота = ALE$
- ☐ D. (Угрозы x Уязвимости x Ценность актива) x Недостаток контроля

18. Что из перечисленного не является целью проведения анализа рисков?

- ☐ A. Делегирование полномочий
- ☐ B. Количественная оценка воздействия потенциальных угроз
- ☐ C. Выявление рисков
- ☐ D. Определение баланса между воздействием риска и стоимостью необходимых контрмер

19. Что из перечисленного не является задачей руководства в процессе внедрения и сопровождения безопасности?

- ☐ A. Поддержка
- ☐ B. Выполнение анализа рисков
- ☐ C. Определение цели и границ
- ☐ D. Делегирование полномочий

20. Почему при проведении анализа информационных рисков следует привлекать к этому специалистов из различных подразделений компании?

- ☐ A. Чтобы убедиться, что проводится справедливая оценка
- ☐ B. Это не требуется. Для анализа рисков следует привлекать небольшую группу специалистов, не являющихся сотрудниками компании, что позволит обеспечить беспристрастный и качественный анализ
- ☐ C. Поскольку люди в различных подразделениях лучше понимают риски в своих подразделениях и смогут предоставить максимально полную и достоверную информацию для анализа
- ☐ D. Поскольку люди в различных подразделениях сами являются одной из причин рисков, они должны быть ответственны за их оценку

21. Что является наилучшим описанием количественного анализа рисков?

- ☐ A. Анализ, основанный на сценариях, предназначенный для выявления различных угроз безопасности
- ☐ B. Метод, используемый для точной оценки потенциальных потерь, вероятности потерь и рисков
- ☐ C. Метод, сопоставляющий денежное значение с каждым компонентом оценки рисков
- ☐ D. Метод, основанный на суждениях и интуиции

22. Почему количественный анализ рисков в чистом виде не достижим?

- ☐ A. Он достижим и используется
- ☐ B. Он присваивает уровни критичности. Их сложно перевести в денежный вид.

- ☐ C. Это связано с точностью количественных элементов
- ☐ D. Количественные измерения должны применяться к качественным элементам

23. Если используются автоматизированные инструменты для анализа рисков, почему все равно требуется так много времени для проведения анализа?

- ☐ A. Много информации нужно собрать и ввести в программу
- ☐ B. Руководство должно одобрить создание группы
- ☐ C. Анализ рисков не может быть автоматизирован, что связано с самой природой оценки
- ☐ D. Множество людей должно одобрить данные

24. Какой из следующих законодательных терминов относится к компании или человеку, выполняющему необходимые действия, и используется для определения обязательств?

- ☐ A. Стандарты
- ☐ B. Должный процесс (Due process)
- ☐ C. Должная забота (Due care)
- ☐ D. Снижение обязательств

25. Что такое CobiT и как он относится к разработке систем информационной безопасности и программ безопасности?

- ☐ A. Список стандартов, процедур и политик для разработки программы безопасности
- ☐ B. Текущая версия ISO 17799
- ☐ C. Структура, которая была разработана для снижения внутреннего мошенничества в компаниях
- ☐ D. Открытый стандарт, определяющий цели контроля

26. Из каких четырех доменов состоит CobiT?

- ☐ A. Планирование и Организация, Приобретение и Внедрение, Эксплуатация и Сопровождение, Мониторинг и Оценка
- ☐ B. Планирование и Организация, Поддержка и Внедрение, Эксплуатация и Сопровождение, Мониторинг и Оценка
- ☐ C. Планирование и Организация, Приобретение и Внедрение, Сопровождение и Покупка, Мониторинг и Оценка
- ☐ D. Приобретение и Внедрение, Эксплуатация и Сопровождение, Мониторинг и Оценка

27. Что представляет собой стандарт ISO/IEC 27799?

- ☐ A. Стандарт по защите персональных данных о здоровье
- ☐ B. Новая версия BS 17799
- ☐ C. Определения для новой серии ISO 27000
- ☐ D. Новая версия NIST 800-60

28. CobiT был разработан на основе структуры COSO. Что является основными целями и задачами COSO?

- ☐ A. COSO – это подход к управлению рисками, который относится к контрольным объектам и бизнес-процессам
- ☐ B. COSO относится к стратегическому уровню, тогда как CobiT больше направлен на операционный уровень
- ☐ C. COSO учитывает корпоративную культуру и разработку политик
- ☐ D. COSO – это система отказоустойчивости

29. OCTAVE, NIST 800-30 и AS/NZS 4360 являются различными подходами к реализации управления рисками в компаниях. В чем заключаются различия между этими методами?

- ☐ A. NIST и OCTAVE являются корпоративными
- ☐ B. NIST и OCTAVE ориентирован на ИТ
- ☐ C. AS/NZS ориентирован на ИТ
- ☐ D. NIST и AS/NZS являются корпоративными

30. Какой из следующих методов анализа рисков пытается определить, где вероятнее всего произойдет сбой?

- ☐ A. Анализ связующего дерева
- ☐ B. AS/NZS
- ☐ C. NIST
- ☐ D. Анализ сбоев и дефектов

31. Что было разработано, чтобы помочь странам и их правительствам построить законодательство по защите персональных данных похожим образом?

- ☐ A. Безопасная OECD
- ☐ B. ISO/IEC
- ☐ C. OECD
- ☐ D. CPTED

Домен 02. Управление доступом.

1. Обзор управления доступом

Управление доступом – это механизм безопасности, который управляет процессом взаимодействия пользователей с системами и ресурсами, а также систем между собой. Этот механизм защищает системы и ресурсы от несанкционированного доступа и принимает участие в определении уровня авторизации после успешного прохождения процедуры аутентификации. Нельзя забывать о том, что кроме пользователей, в сети существуют и другие сущности, которым нужен доступ к сетевым ресурсам и информации. В процессе управления доступом необходимо знать и понимать определения субъекта и объекта.

Доступ – это поток информации между субъектом и объектом. **Субъект** – активная сущность, запрашивающая доступ к объекту или данным внутри объекта. Субъектом может быть пользователь, программа или процесс, использующий доступ к объекту для выполнения своей задачи. **Объект** – пассивная сущность, содержащая информацию. Объектом может быть компьютер, база данных, файл, компьютерная программа, директория или поле таблицы базы данных. Например, если вы просматриваете информацию в базе данных, вы являетесь активным субъектом, а база данных – пассивным объектом. Рисунок 2-1 иллюстрирует субъекты и объекты.

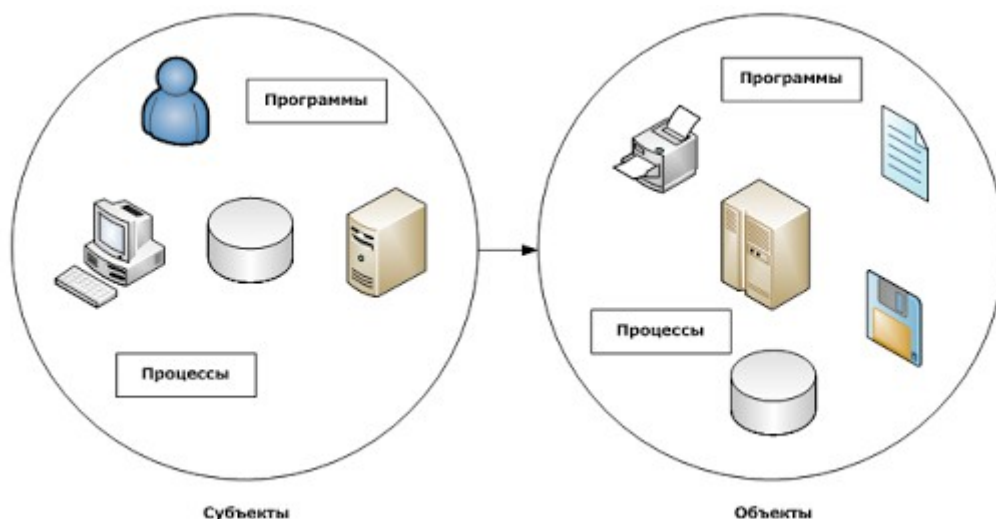


Рисунок 2-1. Субъект – это активная сущность, которая использует доступ к объекту, являющемуся пассивной сущностью

Управление доступом – это широкое понятие, включающее различные типы механизмов, выполняющих функции управления доступом для компьютерных систем, сетей и информации. Управление доступом крайне важно, так как является первой линией обороны в борьбе с несанкционированным доступом к системам и сетевым ресурсам. Когда у пользователя запрашивается имя и пароль для входа в компьютер, это управление доступом. После входа в компьютер, пользователь пытается получить доступ к файлам, которые имеют списки контроля доступа, содержащие перечни пользователей и групп, имеющих право использовать эти файлы. Это тоже управление доступом. Управление доступом позволяет компании управлять, ограничивать, контролировать и защищать доступность, целостность и конфиденциальность ресурсов.

2. Принципы безопасности

Существует три основных принципа безопасности для любых видов управления безопасностью: доступность, целостность и конфиденциальность. Более подробно эти принципы рассматривались в Домене 01, но там они рассматривались с точки зрения управления безопасностью, а сейчас мы будем рассматривать их с точки зрения технологий и

методик управления доступом. Каждый механизм защиты (или управления) реализует как минимум один из этих принципов. Специалист по безопасности должен понимать все возможные способы реализации этих принципов.

Доступность. Информация, системы и ресурсы должны быть доступны пользователям в нужное им время, так как это необходимо для выполнения ими своих обязанностей. Отсутствие доступа к информации может оказать существенное негативное воздействие на продуктивность работы пользователей. Следует применять механизмы обеспечения отказоустойчивости и восстановления для обеспечения непрерывной доступности ресурсов.

Информация имеет различные атрибуты, такие как точность, актуальность, оперативность и секретность. Для биржевых брокеров крайне важно иметь точную и своевременную информацию, чтобы они могли покупать и продавать ценные бумаги в нужное время и по правильной цене. Брокеру не нужно заботиться о конфиденциальности этой информации, его интересует только ее постоянная доступность. С другой стороны, компания, выпускающая безалкогольные напитки, зависит, в первую очередь, от сохранения в тайне рецептов приготовления этих напитков, и будет заботиться об этом, внедряя соответствующие механизмы безопасности.

Целостность. Информация должна быть точной, полной и защищенной от несанкционированных изменений. Механизмы безопасности, обеспечивающие целостность информации, должны уведомлять пользователей или администраторов о фактах незаконных изменений.

Например, если пользователь направляет в банк по системе Интернет-банкинга платежное поручение, банк должен убедиться в его целостности и в том, что никто не внес несанкционированных изменений в сумму, не изменил получателя платежа.

Конфиденциальность. Информация должна быть защищена от несанкционированного раскрытия неуполномоченным лицам, программам или процессам. Одна информация может быть более критична, чем другая информация, поэтому она требует более высокого уровня конфиденциальности. В связи с этим данные должны быть классифицированы. Следует применять механизмы управления, которые указывают, кто имеет доступ к данным и что может делать с ними, получив доступ. Эта деятельность должна контролироваться и постоянно отслеживаться.

Примером конфиденциальной информации могут быть медицинские записи, финансовые счета, исходные тексты программ, военные тактические планы.

Некоторые механизмы безопасности обеспечивают конфиденциальность средствами шифрования, управления логическим и физическим доступом, управления потоками трафика, использованием безопасных протоколов и т. п.

3. Идентификация, аутентификация, авторизация и подотчетность

Пользователю, чтобы получить доступ к ресурсу, нужно сначала подтвердить, что он тот, за кого себя выдает, имеет необходимые полномочия, а также права и привилегии для выполнения действий, которые он запросил. Только при успешном выполнении всех этих шагов пользователю должен предоставляться доступ к ресурсам. Кроме того, необходимо отслеживать действия пользователей, используя для этого средства ведения учета.

Идентификация – это метод проверки, подтверждающий, что субъект (пользователь, программа или процесс) – тот, за кого себя выдает. Идентификация может осуществляться, например, с использованием имени пользователя или номера счета. Для прохождения **аутентификации** субъект обычно должен предоставить вторую часть учетных данных, например, пароль, парольную фразу, криптографический ключ, PIN-код, биометрический атрибут или токен. Эти две части учетных данных сравниваются с предварительно сохраненной информацией о субъекте и, если они совпадают, аутентификация считается

успешной. Далее система проверяет матрицу контроля доступа или сравнивает метки безопасности для проверки, что субъект действительно может использовать ресурс и выполнять запрошенные действия с ним. Если система определяет, что субъект может получить доступ к ресурсу, она **авторизует** его.

Атаки соревнования. Крайне важно, чтобы процессы, выполняющие свои задачи с общими (совместно используемыми) ресурсами, действовали в правильной последовательности. Атаки соревнования (race conditions) возможны, когда два или более процессов совместно используют общие ресурсы. Например, если в программном обеспечении функции аутентификации и авторизации разделены, существует возможность для злоумышленника (например, вызванная уязвимостью в программе) произвести атаку соревнования, чтобы обеспечить выполнение шага авторизации до выполнения шага аутентификации, что может стать причиной получения злоумышленником несанкционированного доступа к ресурсу.

Хотя идентификация, аутентификация, авторизация и подотчетность тесно связаны между собой, каждый элемент имеет различные функции, которые реализуют определенные требования в процессе управления доступом. Пользователь может быть успешно идентифицирован и аутентифицирован для доступа к сети, но он может не иметь разрешения на доступ к файлам на файловом сервере. Либо наоборот, пользователю может быть разрешен доступ к файлам на файловом сервере, но пока он не прошел успешно процедуры идентификации и аутентификации, эти файлы ему недоступны. Рисунок 2-2 иллюстрирует четыре шага, которые необходимо пройти субъекту для получения доступа к объекту.

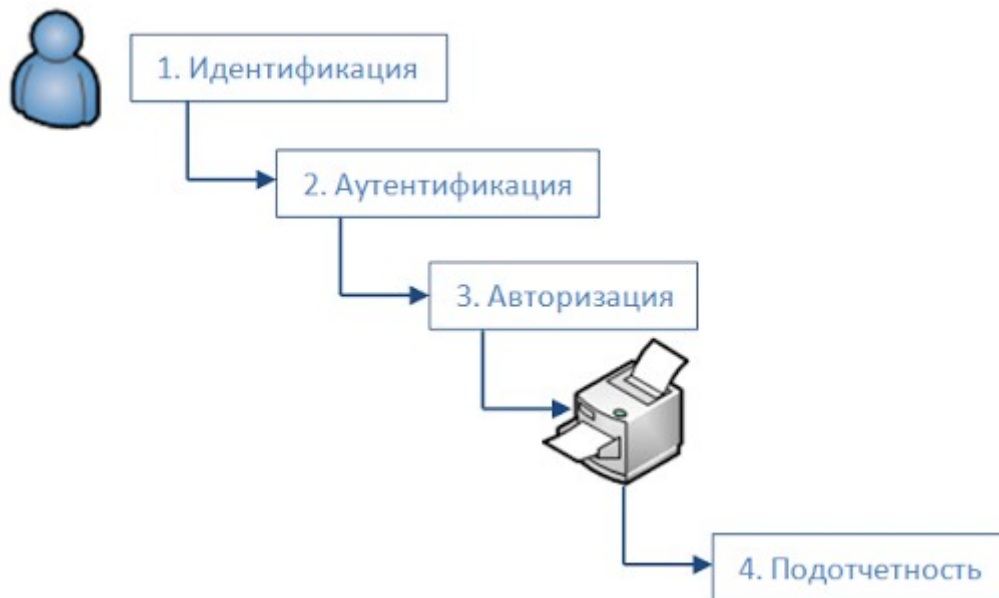


Рисунок 2-2. Для доступа субъекта к объекту должны быть пройдены четыре шага: идентификация, аутентификация, авторизация и подотчетность

Субъект должен нести ответственность за все действия, совершенные от его имени в системе или домене. Единственным способом обеспечения подотчетности является надлежащая идентификация субъекта и запись всех его действий.

Логическое управление доступом – это инструмент, используемый для идентификации, аутентификации, авторизации и подотчетности. Это реализуется в виде программных компонентов, выполняющих функции управления доступом к системам, программам, процессам и информации. Логическое управление доступом может быть встроено в операционную систему, приложения, дополнительные пакеты безопасности, базы данных или системы управления телекоммуникациями. Может быть оказана сложная синхронизация всех механизмов управления доступом, учитывая при этом все возможные уязвимости и не навредив производительности.

ПРИМЕЧАНИЕ. Слова "логическое" и "техническое" управление доступом взаимозаменяемы в контексте данной книги и экзамена CISSP.

В процессе аутентификации должна быть проверена личность человека. Аутентификация, как правило, включает в себя два этапа: ввод публичной информации (имя пользователя, идентификатор, номер счета и т.п.), а затем ввод секретной информации (постоянный пароль, смарт-карта, одноразовый пароль, PIN-код, электронно-цифровая подпись и т.п.). Ввод публичной информации – это идентификация, а ввод секретной информации – аутентификация. Каждый метод, используемый для идентификации и аутентификации, имеет свои плюсы и минусы. Следует проводить надлежащую оценку методов идентификации и аутентификации для выбора правильного механизма для имеющейся среды.

Ссылки по теме:

- FWPro Secure Coding Standards
- “What Are Race Conditions and Deadlocks?” Microsoft Knowledge Base Article 317723

3.1. Идентификация и аутентификация

После прохождения идентификации (посредством некоего идентификатора, например, логина), пользователь должен быть аутентифицирован. Это означает, что он должен доказать, что он является тем, кем представляется. Существует три основных метода, используемых для аутентификации – «что-то знать» (аутентификация по знанию), «что-то иметь» (аутентификация по владению), «кем-то быть» (аутентификация по характеристикам).

Проверка «один к одному» означает, что предъявленные учетные данные сверяются с одним правильным вариантом – система проверяет, является ли пользователь тем, за кого себя выдает. Для этого представленные им учетные данные сверяются с теми, которые хранятся в аутентификационной базе данных (например, при проверке логина и пароля). Если они совпадают, аутентификация считается успешной. **Проверка «один ко многим»** означает, что предъявленные учетные данные сверяются с множеством вариантов – система проверяет, кто этот человек (например, при проверке отпечатка пальца по базе данных).

Для аутентификации по знанию («что-то знать») могут использоваться, например, такие факторы, как пароль, PIN-код, девичья фамилия матери или комбинации этих факторов. Такой способ аутентификации обычно является самым простым для внедрения. Однако у него есть недостаток – другой человек может получить это «знание» и воспользоваться им для несанкционированного доступа к системе.

Для аутентификации по владению («что-то иметь») могут использоваться, например, ключ, магнитная карта, смарт-карта, бейдж. Этот вариант обычно применяется для управления доступом в здание и критичные помещения, но может использоваться и для аутентификации в системе. Недостаток этого метода состоит в том, что идентификатор может быть потерян или украден, что может стать причиной несанкционированного доступа.

Аутентификация по характеристикам («кем-то быть»), т.е. проверка чего-то, специфичного для человека, более интересна. Этот метод основан на проверке физических атрибутов, например, биометрии (более подробно об этом рассказано далее в разделе «Биометрия»).

Строгая аутентификация включает в себя любые два из этих трех методов: человек «что-то знает» и «что-то имеет» или «кем-то является». Это также называют **двухфакторной аутентификацией**. Использование биометрических систем само по себе не является строгой аутентификацией, так как это только один из трех методов. Для строгой аутентификации нужно добавить, например, ввод PIN-кода до проведения сканирования сетчатки глаза.

Требования к идентификации. При выпуске идентификаторов для пользователей, следует учитывать следующее:

- Каждый идентификатор должен быть уникален для обеспечения подотчетности;

- Должна использоваться стандартная схема имен;
- Идентификатор не должен указывать на должность или задачи пользователя;
- Идентификатор не должен совместно использоваться несколькими пользователями.

Обзор управления доступом. Основные концепции управления доступом:

- Идентификация
 - Субъекты предоставляют идентификационную информацию
 - Имя пользователя, идентификатор пользователя, номер счета
- Аутентификация
 - Проверка идентификационной информации
 - Парольная фраза, PIN-код, биометрия, одноразовый пароль, обычный пароль
- Авторизация
 - Использование критериев, определяющих операции, которые субъект может выполнять над объектом
 - «Я знаю, кто вы, что теперь мне разрешить вам делать?»
- Подотчетность
 - Ведение и мониторинг журналов регистрации событий для отслеживания действий субъектов над объектами

Идентификация – достаточно сложная концепция, которая имеет множество нюансов (например, один и тот же человек может иметь множество различных логинов в различных системах, что может вызвать значительные сложности при попытке централизации управления доступом). Определение идентификации в сфере безопасности имеет три ключевых аспекта: уникальность (uniqueness), неявность (nondescriptive) и публикация (issuance). Уникальность подразумевает, что каждый пользователь должен иметь уникальный идентификатор (для обеспечения подотчетности). Неявность означает, что никакая из частей учетных данных не должна указывать на цель учетной записи (например, не следует использовать логины типа «administrator», «CEO», «backup_operator» и т.п.). Публикация подразумевает возможность выпуска средств идентификации другим уполномоченным органом (например, идентификационные карты).

Управление идентификацией

Управление идентификацией (IdM – Identity Management) – это широкое понятие, заключающееся в использовании различных автоматизированных средств идентификации, аутентификации и авторизации пользователей.

Для специалиста по безопасности важно понимать не только сам термин IdM, но и технологии, на основе которых реализуется полноценное корпоративное решение IdM. IdM требует управления уникально идентифицированными сущностями, их атрибутами, учетными данными, правами. IdM позволяет компании организовать жизненный цикл цифровых идентификаторов (создание, поддержка, уничтожение) и надлежащим образом управлять этим жизненным циклом автоматизированными средствами. Корпоративная IdM должна учитывать потребности и масштабы бизнеса.

Рынок продуктов управления идентификацией сегодня процветает, так как эти продукты позволяют снизить административные расходы, повысить безопасность, обеспечить соответствие требованиям и повысить уровень сервиса в масштабах всей компании. Продолжающееся повышение сложности и разнообразия сетевых сред только повышает потребности в управлении тем, кто может получить доступ, к чему и когда. Компании используют различные типы приложений, сетевых операционных систем, баз данных, ERP-систем, CRM-систем – все они используются для выполнения различных задач бизнеса. У

компаний есть партнеры, консультанты, подрядчики, постоянные и временные сотрудники. Каждый из пользователей использует несколько различных видов систем для выполнения своих ежедневных задач, что делает систему управления доступом и обеспечение необходимого уровня безопасности различных типов данных весьма трудной задачей. Часто это приводит к неожиданным и невыявленным «дырам» в защите активов, дублированию и противоречию средств управления, несоответствию действующим требованиям. Целью технологий IdM является упрощение задач администрирования и наведение порядка в этом хаосе.

Ниже представлены наиболее частые вопросы, возникающие сегодня в компаниях, при управлении доступом к активам:

- К чему каждый пользователь должен иметь доступ?
- Кто дает разрешение на доступ и сам доступ?
- Как принимаются решения о доступе в соответствии с политиками?
- Остается ли доступ у уволенных сотрудников?
- Как поддерживать в порядке нашу динамичную и постоянно меняющуюся среду?
- Каков процесс отзыва прав доступа?
- Каким образом осуществляется централизованное управление правами доступа и их мониторинг?
- Почему сотрудники должны помнить по восемь паролей?
- Мы имеем пять различных операционных платформ. Как нам централизовать доступ, если каждая платформа (и приложение) требует своего набора учетных данных?
- Как мы управляем доступом наших сотрудников, клиентов, партнеров?
- Как мы можем убедиться, что мы соответствуем необходимому набору требований?

Традиционный процесс управления доступом, осуществляющийся вручную, с использованием службы каталогов, списков контроля доступа (ACL) и профилей стал неэффективным в современных условиях, поэтому он был заменен автоматизированными приложениями с богатой функциональностью, которые работают совместно друг с другом, создавая инфраструктуру управления идентификацией. Основными целями технологий IdM является оптимизация процессов управления идентификацией, аутентификацией, авторизацией и контролем субъектов на множестве систем в рамках всей компании. Внедрение IdM в крупной компании может являться сложнейшей задачей.

На рынке существует множество решений по управлению идентификацией. В рамках CISSP нужно иметь представление о следующих типах технологий:

- Каталоги
- Управление веб-доступом
- Управление паролями
- Функциональность единого входа (SSO - Single Sign-On)
- Управление учетными записями
- Обновление профилей

Каталоги

Большинство компаний имеют тот или иной каталог (directory), который содержит информацию о компании, ее сетевых ресурсах и пользователях. Большинство каталогов

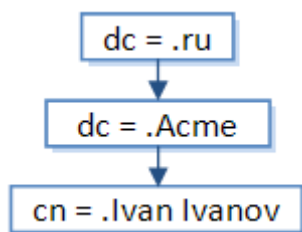
имеют формат иерархической базы данных, основаны на стандарте X.500 и имеют протокол доступа (например, LDAP - Lightweight Directory Access Protocol), позволяющий приложениям и субъектам взаимодействовать с ним. Приложения могут запросить информацию о конкретном пользователе, сделав LDAP-запрос в каталог, а пользователь аналогичным образом может запросить информацию о конкретных ресурсах.

Объекты в рамках каталога управляются с помощью **службы каталога** (directory service). Служба каталога позволяет администратору настраивать и управлять процессом идентификации, аутентификации, авторизации, управления доступом в сети. Объекты в каталоге имеют свои метки и идентифицируются в соответствии с пространством имен.

В среде Windows, когда вы входите в систему, вы регистрируетесь на контроллере домена (DC - Domain Controller), который хранит в своей базе данных иерархический каталог. Эта база данных используется службой каталога (AD - Active Directory), которая организует сетевые ресурсы и выполняет функции управления доступом пользователей. Как только пользователь успешно аутентифицируется на контроллере домена, ему станут доступны определенные сетевые ресурсы (сервер печати, файловый сервер, почтовый сервер и т.д.) в соответствии с настройками AD.

Как же службе каталогов удастся содержать в порядке все эти элементы? Она использует для этого **пространства имен** (namespace). Каждая служба каталога имеет способ идентификации и именования объектов, которыми она управляет. В базе данных, основанной на стандарте X.500 и доступной посредством LDAP, служба каталогов присваивает каждому объекту различимое имя (DN - Distinguished Name). Каждое DN представляет собой набор атрибутов, относящихся к соответствующему объекту, который хранится в каталоге как элемент. В следующем примере DN состоит из общего имени (cn - common name) и компонента домена (dc - domain component).

dn: cn=Ivan Ivanov,dc=Acme,dc=ru
cn: Ivan Ivanov



Это очень простой пример. Обычно компании имеют широкие деревья (каталоги), включающие в себя много уровней и объектов, представляющих различные департаменты, роли, пользователей и ресурсы.

Служба каталога управляет элементами и данными в каталоге и обеспечивает выполнение политики безопасности, выполняя функции управления доступом и аутентификацией. Например, когда вы регистрируетесь на контроллере домена, AD определяет, к каким ресурсам сети у вас должен быть доступ, а к каким – нет.

Организация всего этого хлама. В базе данных каталога, основанной на стандарте X.500, используются следующие правила организации объектов:

- Каталог имеет древовидную структуру организации элементов с использованием конфигурации «родительский-дочерний».
- Каждый элемент имеет уникальное имя, состоящее из атрибутов соответствующего объекта.
- Атрибуты, используемые в каталоге, определены схемой.
- Уникальные идентификаторы называемые различимыми именами (DN).

Схема описывает структуру каталога, используемые в нем имена и другие вещи (схема и

компоненты базы данных более глубоко описаны в Домене 09).

OU (organizational unit) – это организационные единицы. Они используются в качестве контейнеров для других OU, пользователей и ресурсов. Они обеспечивают организационную структуру «родительский-дочерний» (иногда ее называют «дерево-лист»).

Однако при использовании каталогов возникает и ряд проблем. Множество унаследованных устройств и приложений не могут управляться службой каталогов, так как в них не встроено необходимое клиентское программное обеспечение, они могут управляться только через свои собственные системы управления. Соответственно, в сети будут существовать субъекты, сервисы и ресурсы, отсутствующие в службе каталога и управляющиеся администратором в индивидуальном порядке.

Роль каталогов в Управлении идентификацией

Каталог, используемый в IdM, является специализированной базой данных, оптимизированной для операций чтения и поиска. Это основной компонент системы управления идентификацией, так как в нем хранится вся информация о ресурсах, атрибуты пользователей, профили авторизации, роли, политики управления доступом и многое другое. Приложения IdM для выполнения своих функций (авторизация, управление доступом, присвоение разрешений), обращаются за всей нужной информацией в каталог, являющийся централизованным хранилищем. Для получения всей необходимой информации из каталога достаточно сделать всего один запрос.

Много информации, хранящейся в каталоге IdM, разбросано по всей компании. Информация об атрибутах пользователей (статус сотрудника, должностная инструкция, подразделение и т.д.) обычно хранится в базе данных HR (отдела кадров), аутентификационная информация может быть на сервере Kerberos, информация о ролях и группах хранится в базе данных SQL, а информация о доступе к ресурсам размещена в AD на контроллере домена. Продукты управления идентификацией поступают очень хитро, создавая **мета-каталоги** или **виртуальные каталоги**, которые собирают необходимую информацию из множества различных источников и хранят ее в одном центральном каталоге. Это обеспечивает унифицированное представление всей цифровой идентификационной информации пользователей в рамках всей компании. Мета-каталог периодически синхронизируется со всеми хранилищами идентификационной информации, чтобы обеспечить актуальной информацией все приложения и компоненты IdM компании.

Разница между виртуальным каталогом и мета-каталогом состоит только в том, что мета-каталог физически хранит идентификационные данные в своем каталоге, а виртуальный каталог только ссылается на реальные места хранения идентификационных данных.

Рисунок 2-3 иллюстрирует центральный LDAP каталог, который используется службами IdM: управление доступом, инициализация, управление идентификацией. Когда одна из этих служб принимает запрос от пользователя или приложения, она извлекает необходимые данные из каталога, чтобы выполнить запрос. Если эти данные хранятся в различных местах, каталог метаданных, в свою очередь, извлекает данные из их источников и обновляет LDAP каталог.



Рисунок 2-3.Мета-каталог извлекает данные из других источников для внесения их в IdM каталог

Управление веб-доступом

Программное обеспечение управления веб-доступом (WAM – Web access management) управляет правами доступа пользователей к веб-активам компании при обращении посредством веб-браузера. Этот тип технологий постоянно совершенствуется по мере распространения и увеличения количества сервисов электронной коммерции, интернет-банкинга, поставщиков информационных услуг, веб-сервисов и т. д.

Рисунок 2-4 показывает основные компоненты и действия в процессе управления веб-доступом.



Рисунок 2-4.Простой пример управления веб-доступом

1. Пользователь отправляет учетные данные на веб-сервер.
2. Веб-сервер проверяет учетные данные пользователя.
3. Пользователь запрашивает доступ к ресурсу (объекту).
4. Веб-сервер на основе политики безопасности проверяет может ли пользователь выполнять эту операцию.
5. Веб-сервер разрешает доступ к запрошенному ресурсу.

Это простой пример. В более сложном случае пользователь может быть аутентифицирован различными способами (пароль, цифровой сертификат, токен и т.п.), ему могут быть доступны различные ресурсы и сервисы (перевод денежных средств, покупка услуг, обновление профиля и т.п.), а также необходимая инфраструктура. Инфраструктура, как правило, включает в себя ферму веб-серверов (состоящую из множества серверов), каталог, содержащий учетные записи и атрибуты пользователей, базу данных, пару межсетевых экранов, а также несколько маршрутизаторов. Все это расположено на различных уровнях многоуровневой архитектуры. Но сейчас давайте остановимся на более простом варианте.

Программное обеспечение WAM – это главные ворота между пользователями и корпоративными веб-ресурсами. Обычно оно представляет из себя плагин для веб-сервера, работающий в качестве фронтального процесса. Когда пользователь запрашивает доступ, программное обеспечение веб-сервера сначала направляет запрос в каталог, на сервер аутентификации и, возможно, в базу данных. Консоль WAM позволяет администратору настроить уровни доступа, требования аутентификации, шаги настройки учетной записи, выполнять операции технической поддержки.

Инструменты WAM, как правило, предоставляют также возможности SSO, позволяющие пользователю, единожды аутентифицировавшись на веб-сервере, иметь доступ к различным веб-приложениям и ресурсам без необходимости повторного прохождения аутентификации. Продукт, обеспечивающий возможности SSO в веб-среде, должен отслеживать состояние аутентификации пользователя и контекст безопасности при переходе пользователя с одного ресурса на другой.

Предположим, например, что Кэти вошла на свою персональную страницу системы интернет-банкинга. Все коммуникации при этом осуществляются по протоколу HTTP, который не контролирует состояние, т.е. после получения пользователем веб-страницы соединение закрывается и сервер забывает про пользователя. Многие веб-серверы работают в таком режиме, на них поступает огромное количество запросов и они просто предоставляют пользователям веб-страницы. Поддержка постоянного соединения с каждым обратившимся на веб-сервер пользователем потребует колоссальных ресурсов, это нецелесообразно. Поэтому постоянное соединение поддерживается только в случае необходимости.

Сначала, когда Кэти заходит на веб-сайт банка, она видит общедоступные данные, для просмотра которых не требуется аутентификация. При этом нет необходимости веб-серверу поддерживать постоянное соединение, поэтому он работает в неконтролируемом состоянии. Далее Кэти нажимает на пункт «Доступ к моей учетной записи», веб-сервер устанавливает безопасное соединение (SSL) с ее браузером и запрашивает ее учетные данные. После аутентификации Кэти, веб-сервер отправляет ей куки-файл (cookie – маленький текстовый файл), который указывает на то, что она уже аутентифицирована и соответствующий доступ ей разрешен. Когда Кэти переходит со своего депозитного счета на текущий счет, веб-сервер запрашивает у браузера Кэти ее куки-файл, чтобы проверить, что она имеет права доступа к новому ресурсу. Веб-сервер постоянно проверяет этот куки-файл на протяжении всего сеанса работы Кэти, чтобы убедиться, что никто не перехватил сессию и что веб-сервер постоянно взаимодействует именно с системой Кэти, а не с кем-нибудь другим.

Веб-сервер постоянно просит браузер Кэти доказать, что она была аутентифицирована. Браузер делает это, предоставляя информацию куки-файла (информация куки-файла может содержать пароль пользователя, номер учетной записи, уровень безопасности, информацию о браузере и/или информацию персонализации). Все время, пока Кэти аутентифицирована, программное обеспечение веб-сервера отслеживает каждый ее запрос, журналирует ее действия, делает изменения в ее контексте безопасности в соответствии с ее запросами. Контекст безопасности (security context) – это уровень авторизации, основанный на

разрешениях, полномочиях и правах доступа.

Когда Кэти завершает сеанс, куки-файл обычно стирается из памяти браузера, а веб-сервер закрывает соединение и не собирает больше информацию о состоянии сессии этого пользователя.

ПРИМЕЧАНИЕ. Куки-файл в формате текстового файла может быть сохранен на жестком диске пользователя (постоянно) или может храниться только в памяти (в течении сессии). Если в куки-файле содержится какая-либо критичная информация, его следует хранить только в памяти и стирать по окончании сессии.

Пока браузер направляет куки-файл веб-серверу, Кэти не нужно предоставлять свои учетные данные при переходе к другому ресурсу. Это и есть результат работы SSO. Вы только однажды предоставляете свои учетные данные, а дальше происходит постоянная проверка наличия у вас необходимого куки-файла, который позволяет вам переходить от одного ресурса к другому. Если вы завершили сессию с веб-сервером, но вам потребовалось взаимодействовать с ним снова, вы должны повторно пройти аутентификацию, после чего вашему браузеру будет выслан новый куки-файл и все начнется с начала.

ПРИМЕЧАНИЕ. Технологии SSO будут рассмотрены далее в этом домене.

Итак, продукты WAM позволяют администратору настроить и управлять доступом к внутренним ресурсам. Этот тип управления доступом обычно используется для управления внешними сущностями (пользователями), запрашивающими доступ. Продукт WAM может работать как на единственном веб-сервере, так и на ферме серверов.

Управление паролями

Мы рассмотрим требования к паролям, вопросы безопасности и лучшие практики далее в этом домене. Сейчас нам нужно понять, как работает управление паролями в среде IdM.

Сотрудники службы технической поддержки и администраторы знают сколько времени теряется на сброс паролей, когда пользователи забывают их. Основной проблемой чаще всего является количество различных паролей для доступа в различные системы, которые пользователи должны помнить. Для изменения пароля администратор должен напрямую подключиться к управляющему программному обеспечению соответствующей системы и изменить значение пароля. Казалось бы, это не так трудно, однако представьте, что в компании работают 4000 пользователей, используются 7 различных систем, 35 различных приложений, а ряд подразделений работает круглосуточно...

Разработаны различные типы безопасных автоматизированных технологий управления паролями, позволяющие вернуть пользователям возможность пользоваться средствами ИТ и упростить техническую поддержку. Основные подходы к управлению паролями перечислены ниже:

- **Синхронизация паролей** (password synchronization). Уменьшает количество различных паролей от различных систем, которые нужно помнить пользователям.
- **Система самообслуживания для сброса паролей** (self-service password reset). Уменьшает количество звонков в службу технической поддержки, позволяя пользователям самим сбрасывать свои пароли.
- **Сброс паролей с дополнительной помощью** (assisted password reset). Упрощает процесс сброса паролей службой технической поддержки. Могут использоваться различные механизмы аутентификации (биометрия, токены).

Синхронизация паролей

Если у пользователей много паролей, которые им нужно помнить и за которыми нужно следить, они будут записывать их на стикерах и прятать под клавиатуру или просто приклеивать к монитору. Это, конечно же, упрощает жизнь пользователя, но это не хорошо с

точки зрения безопасности.

Технология синхронизации паролей может сократить количество паролей, которые нужно помнить пользователям, и позволить им использовать только один пароль для доступа в несколько систем. Продукт, реализующий эту технологию, будет синхронизировать пароли между различными системами и приложениями, делая это совершенно прозрачно для пользователя. Если пользователю нужно будет помнить только один пароль, можно будет рассчитывать на то, что он будет соблюдать требования по стойкости и безопасности паролей. К тому же, это существенно уменьшит количество звонков в службу технической поддержки.

Единственным недостатком данного подхода является то, что для доступа к различным ресурсам используется только один пароль, и хакеру достаточно получить только один набор учетных данных для возможности несанкционированного доступа ко всем ресурсам. Чтобы усложнить задачу хакера, к паролям должны предъявляться требования по уровню сложности (12 символов, не словарное слово, три знака, буквы в верхнем и нижнем регистрах и т.д.) и регулярной смене. Однако при этом следует соблюдать приемлемый баланс между безопасностью и удобством для пользователей.

Система самообслуживания для сброса паролей

Существуют продукты, которые позволяют пользователям самостоятельно сбрасывать собственные пароли. Это не означает, что пользователи имеют какой-либо привилегированный доступ к системе, позволяющий им менять свои учетные данные. Вместо этого, в процессе регистрации учетной записи пользователя, у него запрашивается несколько персональных вопросов (например, год выпуска из школы, любимый учитель, любимый цвет и т.д.) и ответы на них. Если пользователь после этого забыл свой пароль, он должен предъявить другой аутентификационный механизм (смарт-карту, токен) и ответить на эти предварительно подготовленные вопросы для подтверждения своей личности. Если он все сделал правильно, он получит возможность изменить свой пароль.

Существуют и другие продукты, обеспечивающие возможность самостоятельного изменения паролей пользователями. Например, у забывшего пароль пользователя запрашиваются ответы на заранее определенные (в процессе регистрации учетной записи) вопросы, и, в случае правильного ответа, на его адрес электронной почты (также указанный при регистрации учетной записи) отправляется ссылка, перейдя по которой он сможет ввести для себя новый пароль.

ПРЕДУПРЕЖДЕНИЕ. Вопросы не должны быть такими, ответы на которые может легко узнать любой, либо информация по которым находится в публичном доступе, так как кто угодно может найти эту информацию и попытаться идентифицироваться в системе под чужим именем.

Сброс паролей с дополнительной помощью

Существуют продукты специально для сотрудников технической поддержки, которым нужно работать с пользователями, забывшими свои пароли. Ведь сотрудник службы технической поддержки не должен просто менять пароль для любого звонящего, не аутентифицировав его (иначе можно, используя техники социальной инженерии, позвонить в службу технической поддержки и, представившись другим человеком, получить доступ к системе под его учетной записью).

Эти продукты позволяют сотрудникам службы технической поддержки аутентифицировать звонящих перед сбросом их паролей. Обычно процесс аутентификации выполняется посредством вопросов и ответов, как было описано ранее. Сотрудник службы технической поддержки и звонящий должны быть идентифицированы и аутентифицированы средствами управления паролями до того, как пароль будет сброшен. После сброса пароля следует потребовать от пользователя незамедлительно сменить этот временный пароль, чтобы гарантировать, что только сам пользователь (а не сотрудник службы технической

поддержки) знает свой пароль. Целью продуктов сброса паролей с дополнительной помощью является снижение затрат на телефонную поддержку и обработки вызовов унифицированными и безопасными способами.

На рынке существуют различные продукты управления паролями, предоставляющие одну или несколько функций. Обычно такие продукты интегрируют в корпоративные решения IdM для оптимизации процессов идентификации, аутентификации и управления доступом.

Функциональность единого входа

Мы будем детально рассматривать технологии SSO далее в этом домене, но в данный момент нам нужно понять как продукты SSO используются в качестве решения IdM (или части большого корпоративного решения IdM).

Технологии SSO позволяют пользователю аутентифицироваться один раз и затем использовать различные ресурсы без необходимости повторной аутентификации. Программное обеспечение SSO распознает запросы учетных данных от сетевых систем и приложений и вводит в них необходимую информацию (например, имя и пароль) за пользователя.

Технологии SSO имеют ту же уязвимость, что и технологии синхронизации паролей – если атакующий узнает учетные данные пользователя, он сможет получить доступ ко всем ресурсам, к которым имеет доступ этот пользователь. Кроме того, решение SSO может являться «бутылочным горлышком» или единой точкой отказа – если сервер SSO отключится, пользователи не смогут получить доступ к сетевым ресурсам. Поэтому следует обеспечить некоторый уровень избыточности или использовать отказоустойчивые технологии.

Большинство сред не являются гомогенными с точки зрения устройств и приложений, что еще больше усложняет надлежащее внедрение корпоративного решения SSO.

Унаследованные системы часто требуют другого типа процесса аутентификации, отличающегося от того, который предоставляет система SSO. В настоящее время около 80% приложений и устройств потенциально способны работать с программным обеспечением SSO, а оставшиеся 20% требуют, чтобы пользователи аутентифицировались в них напрямую. В таких ситуациях ИТ-департамент может самостоятельно разработать решение для унаследованных систем на базе командных скриптов.

Другим недостатком SSO может являться высокая стоимость внедрения, в особенности в крупной среде, что может быть неприемлемым для многих компаний. Также, при использовании SSO возникает дополнительная уязвимость, связанная с тем, что все учетные данные пользователей, используемые для доступа к ресурсам компании, хранятся в одном месте. Если атакующий сможет проникнуть в эту сокровищницу, он сможет получить доступ куда захочет и сделать с активами компании что захочет.

Как всегда, безопасность, функциональность и стоимость должны быть надлежащим образом взвешены для определения наилучшего решения для компании.

Управление учетными записями

Управление учетными записями часто реализуется неэффективно и нерационально. Управление учетными записями включает в себя создание учетных записей пользователей во всех системах, изменение привилегий учетных записей (при необходимости) и вывод из эксплуатации учетных записей, которые больше не требуются. В большинстве компаний сотрудники ИТ-департамента создают учетные записи в различных системах вручную, и часто это приводит к тому, что пользователи получают чрезмерные права доступа и привилегии, а после увольнения их учетные записи не блокируются. Все это является следствием отсутствия технологий централизованного управления учетными записями.

Продукты управления учетными записями пытаются решить эти проблемы, позволяя

администраторам управлять учетными записями пользователей во множестве систем. Когда существует несколько каталогов, содержащих профили пользователей или информацию о доступе, программное обеспечение управления учетными записями позволяет реплицировать информацию между каталогами, обеспечивая наличие в каждом актуальной информации.

Теперь взглянем на настройку учетных записей. Во многих компаниях, когда нужно зарегистрировать нового сотрудника, сетевой администратор вручную создает и настраивает учетную запись (одну или несколько), предоставляет ей некоторые права доступа и привилегии. Но откуда сетевой администратор знает, к каким ресурсам следует иметь доступ новому пользователю и какие привилегии должны быть установлены для новой учетной записи? В большинстве случаев сетевой администратор может только догадываться об этом. Именно поэтому пользователи в конечном итоге имеют слишком много прав доступа к слишком большому количеству ресурсов. Чтобы решить эту проблему, целесообразно организовать специализированный процесс документооборота, позволяющий формировать запросы на регистрацию учетных записей для новых пользователей. Такой запрос обычно визируется руководителем сотрудника, для которого запрашивается доступ, после чего учетная запись автоматически создается и настраивается в системе, либо генерируется заявка для технического персонала на создание учетной записи с указанием необходимых прав доступа. Запросы на изменение прав доступа или аннулирование учетной записи делаются аналогичным образом – запрос отправляется руководителю (или тому, кому он делегировал обязанности по утверждению запросов), руководитель утверждает его, и на основании этого делаются изменения в учетной записи.

В продукты управления учетными записями, входящие в состав решений IdM, обычно встроен автоматизированный компонент документооборота. Он не только снижает вероятность ошибок при управлении учетными записями, но также обеспечивает журналирование и предоставляет возможность отслеживания каждого шага (включая утверждение учетной записи). Это позволяет обеспечить подотчетность и предоставляет журналы регистрации событий, которые можно использовать, если что-то пойдет не так. Кроме того это помогает убедиться, что права доступа предоставлены только в необходимом объеме, нет «осиротевших» учетных записей, остающихся активными, когда сотрудник покидает компанию.

ПРИМЕЧАНИЕ. Эти типы продуктов управления учетными записями обычно используются для сопровождения внутренних учетных записей. Управление веб-доступом используется в основном для внешних пользователей.

Как и продукты SSO, корпоративные продукты управления учетными записями обычно дороги и могут потребоваться годы, чтобы полностью внедрить их в масштабах крупной компании. Однако, требования регуляторов заставляют все большее и большее число компаний тратить деньги на такие типы решений.

Инициализация

Давайте немного повторим то, что мы уже знаем, так как далее мы будем основываться на этих концепциях.

Большинство решений IdM извлекают информацию о пользователях из кадровой базы данных (называемой ***авторитетным источником*** (authoritative source)), потому что в ней эта информация уже собрана, хранится в одном месте и постоянно обновляется по мере изменения состояния работников и подрядчиков.

Когда на работу принимается новый сотрудник, информация о нем вместе с именем его руководителя извлекается из кадровой базы данных и копируется в каталог. От имени руководителя сотрудника автоматически формируется и отправляется электронное сообщение с запросом предоставления новой учетной записи. После того, как руководитель утвердит этот запрос, учетная запись будет создана в нужной системе. Со временем, этот новый сотрудник обычно получает различные удостоверяющие его личность атрибуты,

которые используются для целей аутентификации и хранятся в разных системах в сети. На тот момент, когда пользователь запрашивает доступ к ресурсу, все его идентификационные данные уже скопированы из различных хранилищ и кадровой базы данных в этот централизованный каталог (иногда называемый **идентификационным репозиторием** (identity repository)). Это может быть мета-каталог или виртуальный каталог. Компоненты управления доступом системы IdM сравнивают запрос пользователя с политикой управления доступом IdM и убеждаются, что пользователь надлежащим образом идентифицирован и аутентифицирован перед разрешением доступа к ресурсу.

Когда сотрудник увольняется, информация об этом идет из кадровой базы данных в каталог. Автоматически генерируется электронное сообщение и отправляется руководителю для подтверждения необходимости блокировки (уничтожения) учетной записи. После подтверждения руководителя, программное обеспечение управления учетными записями блокирует (уничтожает) все учетные записи, принадлежащие увольняющемуся сотруднику.

Пользователям необходим доступ к ресурсам для выполнения своих должностных обязанностей. Но к каким именно и с какими правами? Этот вопрос в действительности очень сложен в нашем современном распределенном, гетерогенном и хаотичном окружении. Слишком большие права доступа к ресурсам компании повышают потенциальные риски мошенничества (и другие риски). Слишком маленькие права доступа не позволяют пользователям выполнять свои обязанности. Поэтому следует сделать все правильно.

Инициализацией пользователя (user provisioning) называется создание, сопровождение и деактивация пользовательских объектов и атрибутов, имеющих в одной или нескольких системах, каталогах или приложениях в соответствии с бизнес-процессами. Программное обеспечение инициализации пользователей может включать в себя один или несколько следующих компонентов: распространение изменений, документооборот системы самообслуживания, консолидированное администрирование пользователя, делегирование администрирования пользователя, федеративное управление изменениями.

Пользовательские объекты могут представлять собой сотрудников, подрядчиков, производителей, партнеров, клиентов и других получателей сервисов. Сервисы могут включать в себя электронную почту, доступ к базе данных, к файловому серверу или мейнфрейму и т. д.

Итак, мы создаем, поддерживаем и деактивируем учетные записи по мере необходимости, основываясь на потребностях бизнеса. При создании учетной записи создаются также определенные права доступа к активам компании. Это делается посредством инициализации, в рамках которой пользователям предоставляются (или отзываются) права доступа. На протяжении всего жизненного цикла при внесении изменений в идентификатор пользователя, права доступа, разрешения и привилегии следует четко понимать, что именно необходимо изменить, и использовать для этого автоматизированный и контролируемый процесс.

Сейчас вы должны понимать, как эти различные технологии работают вместе для оптимальной реализации IdM. Каталоги предназначены для хранения информации о пользователях и ресурсах. Каталог извлекает идентификационную информацию, хранящуюся в различных местах в сети, чтобы процессы IdM могли из одного места получить все данные, необходимые им для выполнения своих задач. Инструменты управления пользователями предоставляют автоматизированные средства управления идентификаторами пользователей на всем протяжении их (идентификаторов) жизни и обеспечивают инициализацию. Инструменты управления паролями применяются для того, чтобы производительность работы сотрудника не понижалась, если он забудет свой пароль. Технологии SSO требуются внутренним пользователям для того, чтобы аутентифицироваться только один раз для доступа к любым ресурсам компании.

Инструменты управления веб-доступом предоставляют средства SSO внешним

пользователям и управляют доступом к веб-ресурсам. Рисунок 2-5 представляет собой наглядный пример совместной работы многих из этих компонентов.

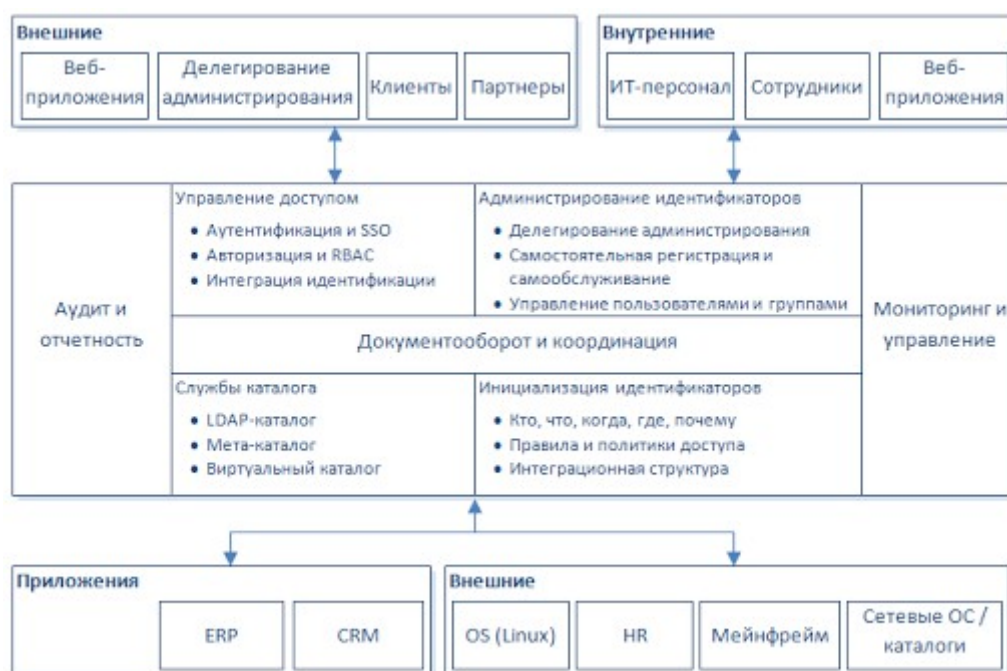


Рисунок 2-5.Компоненты управления идентификацией

Обновление профиля

Большинство компаний хранят значительно больше информации о пользователях, чем просто их имя и права доступа к данным. Эта информация может включать в себя адрес электронной почты, домашний адрес, номер телефона и т.п. Этот набор данных, связанных с личностью пользователя, называется профилем.

Профиль должен храниться централизованно для удобства управления. В корпоративном решении IdM применяются технологии обновления профилей, позволяющие администратору в автоматическом режиме создавать, изменять или удалять профили, когда это необходимо. Некоторые данные, содержащиеся в профилях пользователей, являются не критичными, и пользователи могут обновлять их самостоятельно (это называется *самообслуживанием*), например, изменить адрес своего проживания после переезда в новый дом. Также профили содержат критичные данные, которые должны быть недоступны для пользователей, например, права доступа к ресурсам.

Цифровая личность. Не многие знают о том, что цифровая личность (digital identity) состоит из атрибутов, прав и особенностей. Многие считают цифровой личностью просто идентификатор пользователя, который связан с конкретным лицом. Истина, как правило, значительно сложнее.

Цифровая личность пользователя может состоять из его атрибутов (департамент, роль в компании, график рабочего времени, допуск и т.д.), прав (доступные ему ресурсы, полномочия в компании и т.д.) и особенностей (биометрическая информация, рост, пол и т.д.).

Таким образом, если пользователь запрашивает доступ к базе данных, которая содержит критичную информацию о сотрудниках компании, системе IdM потребуется сначала извлечь необходимые данные о личности пользователя и его полномочиях, прежде чем она авторизует доступ. К примеру, если пользователь является старшим менеджером (атрибут), имеет допуск к секретной информации (атрибут), имеет доступ к базе данных (право), ему будет предоставлен доступ с правами на чтение и запись к определенным записям в базе данных с понедельника по пятницу, с 8 утра до 5 вечера (атрибут).

Каталог (или мета-каталог) системы IdM централизованно хранит всю идентификационную информацию, поэтому он очень важен.

Многие думают, что управление идентификацией работает только на этапе входа на контроллер

домена или сетевой сервер. На самом деле управление идентификацией участвует в целом ряде сложных процессов и технологий, работающих совместно.

Интеграция

Мир постоянно уменьшается по мере того, как технологии все больше сближают людей и компании. Часто, когда мы просто просматриваем один сайт, на самом деле мы взаимодействуем с несколькими различными компаниями, только не знаем об этом. Причина нашего незнания в том, что эти компании совместно используют нашу идентификационную и аутентификационную информацию. Они делают это с добрыми намерениями, стараясь упростить нашу жизнь и позволить, в частности, покупать товары без физического присутствия в магазине.

Например, человек хочет забронировать билеты на самолет и номер в гостинице. Если авиакомпания и гостиница используют федеративную систему управления идентификацией (т.е. имеют доверительные отношения между собой), они смогут совместно использовать идентификационную и (возможно) аутентификационную информацию. При этом, человеку, забронировавшему билет на самолет, сайт авиакомпании предложит забронировать номер в гостинице. Если он согласится, он будет автоматически переадресован на сайт гостиничной компании, который предоставит ему информацию о ближайших от аэропорта гостиницах. Чтобы забронировать номер в одной из этих гостиниц, ему не нужно будет повторно регистрироваться – сайт авиакомпании автоматически отправит информацию о нем на сайт гостиничной компании.

Федеративный идентификатор (federated identity) является перемещаемым, он связан с правами, которые можно использовать в пределах некоторых различных ИТ-систем и компаний. Федеративная идентификация основана на связях отдельных пользовательских идентификаторов в двух или более системах без необходимости синхронизации или консолидации информации каталогов. Федеративный идентификатор дает компаниям и их клиентам более удобный способ доступа к распределенным ресурсам и является ключевым компонентом электронной коммерции.

ПРИМЕЧАНИЕ. Федеративный идентификатор, как и все технологии IdM, обсуждаемые далее, обычно более сложны, чем представляется в этом тексте. Это всего лишь обзор «на один дюйм в глубину», позволяющий сдать экзамен CISSP. Для получения более глубоких сведений о технологиях IdM посетите веб-сайт автора www.logicalsecurity.com/IdentityManagement.

Кому нужно управление идентификацией? Следующий список является хорошим показателем для оценки потребности вашей компании в решениях по управлению идентификацией:

- Если пользователи имеют более шести комбинаций имен и паролей
- Если создание и настройка учетной записи для нового сотрудника занимает у вас более одного дня
- Если отзыв всех прав доступа и блокировка учетной записи увольняющегося сотрудника занимает у вас более одного дня
- Если доступ к критичным ресурсам не может быть ограничен
- Если доступ к критичным ресурсам не может мониториться и/или периодически проверяться.

В следующих разделах рассказывается про различные типы методов аутентификации, наиболее часто используемые и интегрированные во многие современные процессы и продукты управления идентификацией.

Управление доступом и языки разметки

HTML (HyperText Markup Language – гипертекстовый язык разметки) появился в начале 1990-х годов. Предшественниками HTML были **SGML** (стандартный обобщенный язык разметки) и **GML** (обобщенный язык разметки). В настоящее время HTML по-прежнему широко используется.

Язык разметки – это способ структурирования текста для управления его отображением. Когда вы настраиваете поля, размер шрифта и т.п. в текстовом редакторе, вы размечаете текст с помощью языка разметки текстового редактора. Если вы создаете веб-страницу, вы также используете какой-либо язык разметки. Вы можете управлять тем, как выглядит ваша страница и какие функции она предоставляет.

Более мощный язык разметки – ***XML*** (Extensible Markup Language – расширяемый язык разметки) – был разработан в качестве спецификации для создания различных языков разметки. Исходя из этой спецификации XML были разработаны более специфические стандарты, предоставляющие для отдельных отраслей необходимые им функции. Отдельные отрасли имеют различные потребности в использовании языков разметки, что может вызвать проблемы совместимости для компаний, которым требуется взаимодействовать друг с другом. Например, автомобильной компании нужно работать с данными о ценах, цветах, запасных частях, моделях и т.п. А компания, производящая автомобильные шины, работает с другими данными – этапах производства, спецификациях, типах синтетической резины, способах доставки для автомобильных компаний и т.п. Эти компании должны иметь возможность взаимодействия на уровне своих компьютеров и приложений. К примеру, автомобильная компания использует тег языка разметки <модель>, который определяет модель автомобиля, и шинная компания также использует тег <модель>, однако он определяет модель шин. Это приводит к возникновению проблемы совместимости. Чтобы иметь возможность взаимодействия, эти компании должны говорить на одном языке. Таким языком является XML, который должны использовать и понимать обе взаимодействующие компании. Поскольку компании используют различные типы данных, необходимые им для работы, каждая компания (либо отрасль) использует свой производный от XML стандарт, который лучше всего подходит ее потребностям.

ПРИМЕЧАНИЕ. XML используется для множества различных целей, а не только для построения веб-страниц и веб-сайтов.

Но какое все это имеет отношение к управлению доступом? Существует язык разметки, построенный на базе XML, предназначенный для обмена информацией о доступе пользователей к различным ресурсам и сервисам. Скажем, шинная компания предоставляет менеджерам автомобильной компании возможность для заказа шин. Менеджер автомобильной компании Боб регистрируется в программном обеспечении автомобильной компании и делает заказ на 40 комплектов шин. Но каким образом шинная компания узнает, что этот заказ действительно исходит от автомобильной компании и от уполномоченного на это менеджера? Программное обеспечение автомобильной компании может передавать программному обеспечению шинной компании идентификационную информацию о пользователе и группе. На основе этой информации программное обеспечение шинной компании может провести авторизацию данного запроса и реально позволит Бобу заказать 40 комплектов шин.

Языком разметки, который может обеспечить такую функциональность, является ***SPML*** (Service Provisioning Markup Language – обеспечивающий сервис язык разметки). Этот язык позволяет одной компании передавать запросы на обслуживание, а другой компании принимать их и обеспечивать (разрешать) доступ к своим сервисам. Так как и отправляющая, и принимающая компании следуют одному и тому же стандарту (XML), они могут взаимодействовать.

Если автомобильная и шинная компании имеют реализованную модель доверия и общие методы идентификации, авторизации и аутентификации, Боб может быть аутентифицирован и авторизован программным обеспечением автомобильной компании, которое передаст соответствующую информацию программному обеспечению шинной компании, и Бобу не нужно будет дважды проходить аутентификацию. Таким образом, когда Боб регистрируется в программном обеспечении автомобильной компании, он сразу же получает возможность заказа шин в шинной компании. Это означает, что автомобильная и шинная компании имеют

домены безопасности, которые доверяют друг другу (это доверие может быть либо взаимным, либо односторонним). Компания, которая отправляет авторизационные данные, называется *поставщиком подтверждений* (producer of assertion), а получающая их компания – *потребителем подтверждений* (consumer of assertion).

Компании не должны принимать решения об авторизации волей-неволей. Например, разработчик XML для шинной компания должен не просто принимать решение о том, что менеджеры могут выполнять одни функции, бухгалтеры – другие функциональности, а Сью может делать все, что захочет. Компания должна иметь политики безопасности для конкретных приложений, в которых указано, какие роли и отдельные пользователи могут выполнять конкретные функции. Эти решения должны приниматься не на уровне разработчика приложений. Автомобильная и шинная компании должны следовать одинаковым политикам безопасности, чтобы когда менеджер регистрируется в приложении автомобильной компании, обе компании одинаково понимают, что может делать эта роль. Это является целью **XACML** (eXtensible Access Control Markup Language – расширяемого языка разметки управления доступом). Политики безопасности приложений могут совместно использоваться различными приложениями, чтобы гарантировать, что все приложения следуют одним и тем же правилам безопасности.

ПРИМЕЧАНИЕ. Кто разрабатывает и отслеживает все эти стандартизованные языки? Это делает Организация по развитию структурированных информационных стандартов (OASIS – Organization for the Advancement of Structured Information Standards). Эта организация разрабатывает и поддерживает стандарты по различным аспектам взаимодействия через веб.

Итак, подытожим эти сведения. Компаниям нужен способ управления информацией внутри их приложений. XML является стандартом, обеспечивающим структуры метаданных, которые позволяют описывать данные. Компаниям необходимо иметь возможность передавать свою информацию, для чего им целесообразно использовать XML, являющийся международным стандартом и позволяющий избежать проблем совместимости.

Пользователям на стороне отправителя нужно иметь доступ к сервисам на стороне получателя, что обеспечивается с помощью SPML. Принимающая сторона должна убедиться, что пользователь, который делает запрос, надлежащим образом аутентифицирован отправляющей стороной, прежде чем она разрешит доступ к запрошенному сервису, что обеспечивается посредством SAML. Чтобы обеспечить, что отправляющая и принимающая компания следуют одним и тем же правилам безопасности, они должны следовать одинаковым политикам безопасности, что является функциональностью, обеспечиваемой XACML.

ПРИМЕЧАНИЕ. XML рассматривается в Домене 09.

Для получения дополнительной информации об этих языках разметки и их функциях, посетите следующие сайты:

- www.oasis-open.org/home/index.php
- www.w3.org/XML
- <http://saml.xml.org/>
- <http://identitymngn.sourceforge.net/>

Ссылки по теме:

- Identity Management

Биометрия

Биометрия идентифицирует человека, анализируя уникальные личные атрибуты или поведение, она является одним из самых эффективных и точных методов идентификации, так как такие атрибуты обычно нельзя изменить (не нанося физического вреда) и сложно

подделать. Биометрия является очень изощренной системой, поэтому она обычно дороже и сложнее, чем другие механизмы идентификации.

Биометрические системы обычно делятся на две категории: физиологическая («кто ты?») и поведенческая («что ты делаешь?»). Физиологические биометрические системы основаны на атрибутах, уникальных для каждого отдельного человека (например, сетчатка или радужная оболочка глаза, отпечаток пальца). Поведенческие биометрические системы основаны на индивидуальных характеристиках человека (например, динамика подписи).

Биометрические системы сканируют атрибут (поведение) человека, а затем сравнивают его с заранее записанным эталонным образцом. В работе таких систем могут возникать ошибки двух видов: ошибочное разрешение (false positive) и ошибочный отказ (false negative). Такие системы должны быть откалиброваны для обеспечения максимально возможной точности результатов.

Когда биометрическая система отказывает в доступе уполномоченному человеку, это называется *ошибкой 1 рода* (уровень ошибочных отказов). Когда разрешает доступ самозванцу, которому в доступе должно быть отказано, это называется *ошибкой 2 рода* (уровень ошибочных разрешений). Целью является минимизация уровней ошибок обоих видов, но нужно учитывать, что ошибки 2 рода более опасны, поэтому минимизация их уровня более приоритетна.

При сравнении различных биометрических систем используется множество различных переменных, но одна из них является самой важной – это **CER** (Crossover Error Rate - уровень пересечения вероятности ошибок). CER измеряется в процентах и представляет собой точку, в которой уровни ошибок 1 и 2 рода равны. Это показатель точности системы (чем он ниже, тем точнее система). CER является беспристрастной оценкой биометрической системы и помогает создать стандарты, в соответствии с которыми продукты различных производителей можно адекватно оценивать и сравнивать.

ПРИМЕЧАНИЕ. Иногда CER называют ERR (Equal Error Rate – уровень равной вероятности ошибок).

Реальные среды имеют свои собственные требования к уровню безопасности, которые указывают какое количество ошибок 1 и 2 рода будет считаться приемлемым. Например, военная организация, беспокоящаяся о конфиденциальности информации, будет готова принять определенное количество ошибок 1 рода, но никаких ошибочных разрешений (ошибок 2 рода) в ее системах быть не должно. Биометрические системы могут быть откалиброваны. Увеличением их чувствительности можно минимизировать количество ошибок 2 рода, однако следует учитывать, что это приведет к росту количества ошибок 1 рода.

Биометрия является одним из наиболее дорогостоящих методов идентификации людей, кроме того, существует ряд других препятствий, мешающих широкому распространению биометрических систем. Это и возможное неприятие таких систем пользователями, и значительное время для первоначальной регистрации пользователей, а также производительность (некоторые действия пользователям придется повторять по несколько раз). Часто люди просто не хотят, чтобы машина считывала сетчатку их глаз и геометрию их рук. Все это существенно замедляет широкое распространение биометрических систем в нашем обществе.

В процессе регистрации, пользователь предоставляет свои биометрические данные (отпечатки пальцев, голоса), а биометрический считыватель преобразует полученные данные в двоичный вид. В зависимости от системы, считыватель может создать хэш-функции биометрических данных или зашифровать сами биометрические данные, либо сделать и то и другое. Биометрические данные со считывателя отправляются (в виде эталонного файла) в аутентификационную базу данных, в которой для пользователя создается учетная запись.

Когда позднее пользователю потребуется пройти процедуру аутентификации, предъявленные им биометрические данные будут сравниваться с данными (эталонным файлом), хранящимися в этой базе данных. Если они совпадут, пользователь будет считаться идентифицированным и/или аутентифицированным.

Перед использованием биометрической системы, настраивается пороговый уровень совпадения введенных пользователем биометрических данных с эталоном, при достижении которого аутентификация будет считаться успешной. Требовать стопроцентного совпадения не имеет смысла, так как пользователи не смогут аутентифицироваться в системе за разумное время, хотя это и исключит ошибки 2 рода. Уровень совпадения зависит от уровня чувствительности биометрических систем, который дополнительно может быть снижен загрязнением считывателя, маслом на пальце пользователя или другими небольшими проблемами.

Скорость обработки. При выборе биометрических устройств для покупки, нужно учесть один важный параметр – время аутентификации пользователей. От момента считывания биометрических данных пользователя до момента получения разрешения или отказа не должно проходить более 5 - 10 секунд.

Ниже представлен обзор различных типов биометрических систем и те физиологические или поведенческие характеристики, которые они используют.

Отпечаток пальца

Отпечатки пальцев состоят из бороздок папиллярных линий, их раздвоений и соединений, а также более детальных характеристик, называемых минуциями. Отличия этих минуций дает каждому человеку уникальные отпечатки пальцев. Человек помещает свой палец на устройство, которое считывает его отпечаток и сравнивает с эталоном. Если они совпадают, личность человека считается подтвержденной.

ПРИМЕЧАНИЕ. Системы, считывающие отпечатки пальцев, сохраняют полную информацию отпечатка, которая занимает много места и требует значительных ресурсов для обработки. Поэтому технологии сканирования отпечатков пальцев извлекают только определенную часть информации отпечатка пальца, хранение которой требует меньше места, ускоряется поиск и сравнение с эталонной информацией в базе данных.

Сканирование ладони

Ладонь также содержит богатую информацию, многие из аспектов которой могут использоваться для идентификации человека. Ладонь имеет складки, бороздки, углубления, уникальные для каждого человека. Сканирование ладони включает в себя сканирование отпечатка каждого пальца. Человек помещает свою ладонь на биометрическое устройство, которое производит ее сканирование. Полученная при этом информация сравнивается с эталонным файлом.

Геометрия руки

Форма руки человека (форма, длина и ширина руки и пальцев) определяет геометрию руки, являющуюся уникальной особенностью, значительно отличающей одного человека от другого. Это используется биометрическими системами для идентификации. Человек помещает свою руку на устройство, которое имеет выемки для каждого пальца. Система сравнивает геометрию каждого пальца и руки в целом с информацией в эталонном файле.

Сканирование сетчатки глаза

Система сканирует рисунок кровеносных сосудов сетчатки на задней стенке глазного яблока. Это изображение абсолютно уникально у разных людей. Камера с помощью инфракрасных лучей подсвечивает сетчатку, получает отраженное изображение кровеносных сосудов и сравнивает его с эталонным файлом.

Сканирование радужной оболочки глаза

Радужная оболочка – это цветной круг, окаймляющий черный зрачок. Изображение радужной оболочки имеет уникальные узоры, трещины, цвета, кольца, короны, борозды. Каждая из этих уникальных характеристик снимается камерой и сравнивается с эталонным файлом. Из всех биометрических систем, сканирование радужной оболочки глаза является самым точным методом. Радужная оболочка не меняется с возрастом, что также снижает вероятность ошибок в процессе аутентификации.

Динамика подписи

Когда человек ставит подпись, он обычно делает это одним и тем же образом и за одно и то же время. В процессе подписи физические движения вырабатывают электрические сигналы, которые фиксируются биометрической системой. Эти сигналы обеспечивают уникальные характеристики, отличающие одного человека от другого. Динамика подписи содержит больше информации, чем просто изображение подписи. При проведении идентификации человека по динамике подписи, учитывается скорость подписи, давление, способ, которым человек держит перо. Все это обеспечивает более точную идентификацию.

Динамика работы на клавиатуре

Также как и при анализе динамики подписи, этот метод фиксирует электрические сигналы при наборе пользователем на клавиатуре определенной фразы. При этом биометрическая система фиксирует скорость и движения процесса ввода. Каждый человек имеет свой стиль и скорость, которые преобразуются в уникальные сигналы. Этот метод аутентификации эффективнее, чем проверка пароля. Повторить стиль печати человека гораздо сложнее, чем подобрать его пароль.

Штамп голоса

Звуки голоса и стиль речи людей имеют множество небольших отличий. Биометрическая система может создавать штамп голоса, который будет уникален для каждого человека, и сравнивать эту информацию с эталонным файлом. В процессе подготовки эталонного файла человека просят произнести несколько различных слов. При проверке система перемешивает эти слова (чтобы избежать попыток аудиозаписи и воспроизведения) и предлагает человеку повторить их.

Сканирование лица

Человеческое лицо содержит множество индивидуальных признаков (структура костей, форма носа, ширина глаз, размер лба, форма подбородка). Эта информация сканируется и сравнивается с эталонным файлом. Если информация совпадает, человек считается идентифицированным.

Топография кисти

Тогда как метод анализа геометрии руки учитывает размер и толщину руки и пальцев человека, при анализе топографии кисти рассматривается форма поверхности кисти, ее изгибы, рисунок кожи. Камера системы делает снимки кисти с разных ракурсов под разными углами и сравнивает с эталонным файлом. Получаемые при этом атрибуты не являются в достаточной степени уникальными, поэтому данный метод обычно применяется совместно с методом анализа геометрии руки.

Некоторые биометрические системы дополнительно проверяют пульсации и/или тепло тела, чтобы убедиться, что идентифицируемый человек жив. Это позволяет избежать ложной идентификации с помощью чужого пальца или глаза.

Как и любая другая система, биометрия имеет свои недостатки и проблемы, которые вызваны, в основном, тем, что работа биометрических систем зависит от конкретных уникальных характеристик живых людей. Многие из этих характеристик со временем меняются, а эталонная информация биометрических систем остается статичной. Например,

распознавание речи может быть затруднено, если человек зашел в теплое помещение с мороза; беременность может изменить структуру сетчатки; человек может потерять палец. В этом мире ни в чем нельзя быть полностью уверенным.

Ссылки по теме:

- Michigan State University Biometrics Research web site
- The Biometric Consortium home page

Пароли

Идентификатор пользователя в паре с постоянным (повторно используемым) паролем в настоящее время является самым распространенным способом идентификации и аутентификации. Пароль – это защищенная строка символов, используемая для аутентификации пользователя. Пароль является фактором «что-то знать» (аутентификация по знанию). Очень важно использовать надежные пароли и надлежащим образом управлять ими.

Управление паролями

Хотя пароли и являются самым распространенным механизмом аутентификации, они являются в то же время и самым слабым из доступных механизмов, т.к. пользователи не заботятся о безопасности, выбирают легко угадываемые пароли, сообщают их другим, пишут их на стикерах и клеят на монитор или прячут под клавиатуру. Большинству пользователей безопасность кажется неважной и неинтересной, они начинают интересоваться ей только тогда, когда кто-то взламывает их компьютер и копирует с него конфиденциальную информацию.

Если пароли надлежащим образом генерируются, обновляются и хранятся в секрете, они могут обеспечить эффективную безопасность. Для создания паролей пользователей целесообразно использовать генераторы паролей, которые должны быть предварительно настроены на генерацию несложных для запоминания, удобопроизносимых, но не словарных паролей.

Если пользователи сами выбирают себе пароли, следует использовать механизмы операционной системы, для установления требований сложности паролей, их неповторяемости и периодической смене. Это значительно усложнит задачу атакующего, который может попытаться угадать или подобрать пароль пользователя.

Для атаки на пароли атакующий может использовать несколько различных методик:

- **Электронный мониторинг.** Прослушивание сети с целью перехвата трафика процесса аутентификации, для повторной отправки этого трафика серверу аутентификации в другое время (*replay attack*).
- **Доступ к файлу с паролями.** Обычно выполняется непосредственно на сервере аутентификации. Парольный файл содержит множество паролей пользователей, его компрометация крайне опасна. Этот файл должен быть защищен с помощью механизмов управления доступом и криптографии.
- **Подбор паролей.** Выполняется специальными утилитами, которые перебирают все возможные комбинации букв, цифр и символов для поиска действительного пароля (*brute force attack*).
- **Атака по словарю.** Для этого используются специальные текстовые файлы, которые содержат тысячи слов, каждое из которых проверяется на совпадение с паролем пользователя (*dictionary attack*).
- **Социальная инженерия.** Атакующий обманным путем заставляет пользователя

сообщить ему учетные данные для доступа к определенным ресурсам.

- **Rainbow-таблицы.** Атакующий может воспользоваться предварительно подготовленными таблицами, содержащими значения хэшей всех возможных паролей.

ПРИМЕЧАНИЕ. Rainbow-таблицы содержат пароли, уже преобразованные в хэш-значения. Используя их, атакующий просто сравнивает перехваченное значение хэша пароля со значениями в таблице, и, в случае совпадения, сразу же получает сам пароль открытым текстом. Это может сократить время подбора пароля до нескольких секунд!

Существуют методики, обеспечивающие еще один уровень безопасности для паролей и процесса их использования. Например, после успешной регистрации сообщать пользователю дату и время его последнего успешного входа, место, откуда был произведен вход, были ли зафиксированы неудачные попытки входа. Это позволит пользователю выявить попытки входа в систему под его именем. Также, администратор может установить количество неудачных попыток регистрации, после которого учетная запись будет заблокирована на определенное время. Кроме того, следует сократить срок «жизни» паролей, чтобы пользователи их регулярно меняли (но так, чтобы это не представляло существенных сложностей для них), использовать средства аудита для отслеживания успешных и неудачных попыток регистрации, провести обучение пользователей по вопросам парольной защиты, объяснив им, почему следует защищать пароли, каким образом пароли могут быть похищены.

Проверка паролей

Многие компании проводят проверки выбранных пользователями паролей, используя утилиты, производящие подбор паролей по словарю или полным перебором. Это позволяет выявить слабые пароли. Однако следует учитывать, что такие утилиты могут использоваться как сотрудниками безопасности и администраторами, так и злоумышленниками, создавая дополнительную уязвимость.

Необходимо перед проведением проверок паролей сотрудников получить на это соответствующее разрешение руководства, чтобы не пришлось потом объяснять генеральному директору, что вы взломали его пароль в целях повышения безопасности компании.

Хэширование и шифрование паролей

В большинстве случаев, злоумышленник, даже перехватив пароль, должен приложить немалые усилия, чтобы узнать сам пароль, так как большинство систем хэшируют пароли (например, с использованием алгоритмов MD4, MD5), а не пересылают их открытым текстом. В Windows-системах пароли хранятся в виде хэш-значений в базе данных SAM (Security Account Management). Для дополнительной защиты администраторы могут использовать утилиту Syskey, которая может работать в трех режимах:

- **Режим 1.** Системный ключ генерируется, шифруется и хранится локально. Компьютер можно перезапускать и он будет нормально работать без вмешательства пользователя.
- **Режим 2.** Системный ключ генерируется, шифруется и хранится локально, но защищен паролем. Администратор должен вводить пароль разблокировки Syskey при каждом запуске компьютера, так как пароль локально не хранится.
- **Режим 3.** Системный ключ генерируется, шифруется и хранится на дискете или компакт-диске, без наличия которого запуск компьютера невозможен.

Unix-системы не используют реестр и базы данных SAM, в них пароли хранятся в файле, называемом shadow в виде хэш-значений. Кроме того, Unix-системы добавляют в процесс шифрования случайные значения (salts), усложняющие процесс взлома.

Устаревание паролей

Многие системы позволяют администратору установить срок действия паролей, что заставит пользователей регулярно менять их. Система может также хранить список последних 5 или 10 паролей (историю паролей), чтобы не позволить пользователям повторно использовать старые пароли.

Ограничение количества попыток регистрации

Может быть установлено пороговое значение количества неудачных попыток регистрации, при превышении которого учетная запись будет заблокирована на заранее определенное время (или до момента разблокировки администратором вручную). Это поможет противостоять атаке, использующей перебор различных вариантов учетных данных до нахождения работающей комбинации (имя пользователя и пароль).

Когнитивные пароли

Когнитивные пароли (cognitive passwords) – это пароли на основе фактов или мнений, используемые для аутентификации человека. В процессе регистрации пользователь отвечает на несколько личных вопросов (например, девичья фамилия матери, любимый цвет, имя собаки, год окончания школы). Затем, при аутентификации, пользователю задаются те же вопросы, запомнить ответы на которые гораздо проще, чем запомнить сложный пароль. Такой процесс аутентификации лучше всего применять в сервисах, которые пользователь использует не очень часто, например, для восстановления обычного пароля.

Одноразовые пароли

Одноразовые пароли (OTP - one-time password) называют также динамическими паролями. Они используются для аутентификации лишь один раз. После использования пароль становится недействительным и даже в случае его перехвата злоумышленником, он не может быть использован. Такие способы аутентификации применяются в среде, в которой необходим более высокий уровень безопасности, чем тот, который предоставляют обычные пароли. Одноразовые пароли генерируются токенами, которые бывают двух типов: синхронные и асинхронные.

Токены

Токен (или генератор паролей) обычно представляет собой миниатюрное устройство, имеющее жидкокристаллический дисплей и, возможно, клавиатуру. Токен аппаратно отделен от компьютера, к которому пользователь пытается получить доступ. Токен и служба аутентификации должны быть синхронизированы определенным образом, чтобы иметь возможность аутентифицировать пользователя. Токен предоставляет пользователю последовательность символов, которая должна быть введена в качестве пароля для входа в компьютер. Только токен и служба аутентификации знают, что означают эти символы. Так как токен и сервер аутентификации синхронизированы, токен может отображать именно тот пароль, который ожидает сервер аутентификации. После однократного использования такой пароль становится недействительным.

Синхронные токены

Синхронный токен синхронизируется со службой аутентификации, используя время или счетчик, в качестве ключевой части процесса аутентификации. Если синхронизация основана на *времени*, на токене и сервере аутентификации должно быть установлено одинаковое время на внутренних часах. На токене для создания одноразового пароля, показываемого пользователю, используется значение времени и секретный ключ. Пользователь вводит отображаемое на токене значение и свой идентификатор в компьютер, служба аутентификации расшифровывает введенное значение и сравнивает его с ожидаемым. Если они совпадают, аутентификация считается успешной и пользователю предоставляется доступ к компьютеру и ресурсам.

Если синхронизация основана на *счетчике*, пользователь должен начать процесс регистрации на компьютере и процесс создания одноразового пароля (нажав кнопку на токене). Это заставляет службу аутентификации и токен сформировать следующее аутентификационное значение. Это значение и секрет хэшируются и показываются пользователю, который вводит отображаемое на токене значение и свой идентификатор в компьютер.

В обоих вариантах используется общий (для токена и службы аутентификации) секретный ключ, используемый как для зашифрования, так и для расшифрования.

Асинхронные токены

Асинхронный токен использует схему запрос/ответ для аутентификации пользователя. В этом случае сервер аутентификации посылает пользователю запрос и случайное значение (называемое *nonce*). Пользователь вводит это значение в токен, который шифрует его и возвращает пользователю результат, который и является одноразовым паролем. Затем пользователь отправляет полученное значение и свой идентификатор серверу аутентификации. Если сервер аутентификации может расшифровывать это значение, и оно совпадает с первоначальным значением (*nonce*), аутентификация считается успешной.

ПРИМЕЧАНИЕ. Одноразовый пароль может также генерироваться программным обеспечением, в таком случае аппаратная часть (например, токен) не требуется. Это называется программным токеном и требует, чтобы служба аутентификации и приложение имели один и тот же базовый секрет, который используется для генерации одноразового пароля.

Оба типа токенов подвержены маскардингу, что может быть вызвано совместным использованием идентификатора пользователя и токена, либо потерей токена пользователем (для снижения этого риска, токен может требовать ввод PIN-кода перед генерацией одноразового пароля, что является примером двухфакторной аутентификации). Кроме того, на токене может быть повреждена батарея или может произойти другая неисправность, не позволяющая пользователю пройти аутентификацию. Однако токены не уязвимы для перехвата информации, sniffинга или угадывания пароля.

Ссылки по теме:

- RFC 2289 - A One-Time Password System
- RFC 2444 - The One-Time-Password SASL Mechanism
- One-Time Password
- RSA SecureID Authentication home page

Криптографические ключи

Другим способом идентификации пользователя является использование закрытого ключа для генерации электронной цифровой подписи (ЭЦП, digital signature). Пароль является наислабейшим способом аутентификации, так как он может быть перехвачен при передаче через сеть. ЭЦП в качестве способа аутентификации может применяться в средах, где требуется более высокий уровень безопасности, чем тот, который обеспечивают пароли. Секретным ключом должен владеть только один человек. ЭЦП – это технология, которая использует закрытый ключ для зашифрования значения хэша (дайджеста сообщения). Действие по зашифрованию значения хэша на секретном ключе называется *подписанием сообщения ЭЦП*. ЭЦП присоединяется к сообщению, чтобы подтвердить, что сообщение исходит от определенного источника и что оно не было изменено в процессе передачи. Открытый ключ может быть общедоступным и он не может быть скомпрометирован, как закрытый ключ. Подробнее о закрытых и открытых ключах, ЭЦП и инфраструктуре открытых ключей (PKI) мы будем говорить в домене «Криптография», однако на данный момент вам нужно понимать, что закрытый ключ и ЭЦП являются механизмами, которые

можно использовать для аутентификации пользователя.

Парольные фразы

Парольная фраза (passphrase) – это длинная последовательность символов, которая, в некоторых случаях, заменяет пароль при аутентификации. Пользователь вводит парольную фразу в приложение, а оно преобразует ее в **виртуальный пароль** необходимой приложению длины и формата. Например, пользователь хочет аутентифицироваться в приложении и вводит парольную фразу, скажем, «СоюзНерушимыйРеспубликСвободных». Приложение преобразует эту фразу в реальный пароль, который в действительности используется для аутентификации. Чаще всего парольную фразу пользователь генерирует так же, как и обычный пароль при первом входе в компьютер. Парольная фраза лучше защищена, чем пароль, так как она длиннее и ее сложнее получить атакующему. К тому же пользователю проще запомнить парольную фразу, чем пароль.

Карты памяти

Основным отличием карт памяти от смарт-карт являются их возможности по обработке информации. **Карта памяти (memory card)** может хранить информацию, но не может ее обрабатывать. **Смарт-карта (smart card)** помимо хранения информации имеет необходимое аппаратное и программное обеспечение для обработки информации. Карта памяти может хранить информацию для аутентификации пользователя таким образом, что для успешной аутентификации пользователю достаточно только ввести свой идентификатор или PIN-код и предоставить карту памяти. Если введенные пользователем реквизиты совпадут с данными на карте памяти, пользователь будет успешно аутентифицирован. Если пользователь вводит PIN-код, это является примером двухфакторной аутентификации («что-то знать» – PIN-код и «что-то иметь» – саму карту). Карта памяти может также хранить идентификационные данные, которые извлекаются из нее считывателем и передаются вместе с PIN-кодом серверу аутентификации.

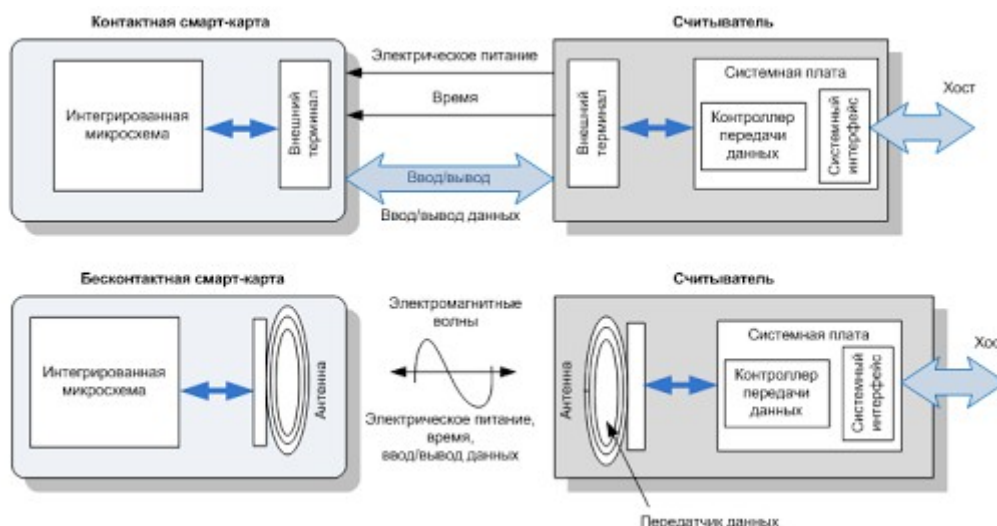
Примером карты памяти является магнитная карта, которая используется для входа в здание. Человек вводит PIN-код и «прокатывает» карту по считывателю. Если информация верна, на считывателе загорается зеленая лампочка и человек может открыть дверь и войти в здание. Другим примером является банковская карта с магнитной полосой.

Карта памяти может использоваться с компьютерами, однако она требует применения специального считывателя для обработки информации. Это увеличивает стоимость данного решения (в особенности, если считыватель требуется на каждый компьютер компании), также стоимость увеличивает процесс изготовления («прошивки») карт памяти. Использование карты памяти обеспечивает более защищенный метод аутентификации, чем использование паролей, так как злоумышленнику необходимо получить и саму карту, и знать корректный PIN-код. Администраторы и руководство компании должны оценить стоимость и преимущества организации аутентификации с помощью карт памяти, чтобы выбрать наиболее оптимальный для компании механизм аутентификации.

Смарт-карты

Смарт-карта имеет возможности обработки хранящейся в ней информации, так как она обладает микропроцессором и интегральными схемами, встроенными в саму карту. Смарт-карта может обеспечить двухфакторную аутентификацию, требуя от пользователя ввести PIN-код («что-то знать»), чтобы разблокировать смарт-карту («что-то иметь»). Существует два основных типа смарт-карт: контактные и бесконтактные. **Контактная** смарт-карта имеет золотые контактные площадки на лицевой стороне. Когда такая карта полностью вставлена в считыватель, эти контактные площадки соприкасаются с соответствующими электрическими контактами в считывателе, которые предназначены для электрического питания компонентов смарт-карты и передачи необходимой для аутентификации информации. **Бесконтактная** смарт-карта имеет антенну, которая проложена по периметру

карты. Когда карта попадает в электромагнитное поле считывателя, антенна в карте вырабатывает достаточно энергии для электрического питания внутренних компонентов карты. Результаты обработки смарт-картой аутентификационных данных могут транслироваться посредством той же антенны. Аутентификация может выполняться с использованием одноразового пароля, применения метода запрос/ответ или с помощью закрытого ключа при использовании в среде PKI.



ПРИМЕЧАНИЕ. Существует два типа бесконтактных смарт-карт: гибридные и комбинированные. Гибридная карта имеет два чипа, что позволяет ей работать в двух форматах – контактном и бесконтактном. Комбинированная карта имеет один чип, который может работать с контактным или бесконтактным считывателем.

Информацию из смарт-карты до ввода правильного PIN-кода считать невозможно, так как она хранится в зашифрованном виде. Этот факт, а также сложность самой смарт-карты не позволяет провести «обратный инжиниринг» и вмешаться в работу карты. Кроме того, смарт-карта может быть запрограммирована на выявление любых попыток вмешательства в свою работу. При выявлении таких попыток, хранящаяся в ней информация может автоматически уничтожаться.

Препятствием для использования смарт-карт является слишком высокая стоимость считывателей и расходы на генерацию карт, как и в случае с картами памяти, хотя стоимость карт памяти ниже. Смарт-карты сами по себе дороже карт памяти, так как они имеют дополнительные интегральные схемы и микропроцессор. По сути, смарт-карта – это разновидность компьютера, поэтому она подвержена многим угрозам и рискам, присущим компьютерам.

Смарт-карты имеют много различных возможностей. Они могут хранить персональную информацию способами, устойчивыми к взлому. Они позволяют изолировать критичные с точки зрения безопасности расчеты. Они могут применяться в качестве портативных и безопасных устройств для хранения ключей криптографических систем. Наличие в них памяти и интегральных схем, позволяет организовать работу алгоритмов шифрования непосредственно в самой карте и использовать это для безопасной аутентификации, используемой в масштабах всей компании.

Атаки на смарт-карты

Смарт-карты более устойчивы к взлому, чем карты памяти, однако с годами и для них были найдены успешные способы атак. Например, существует возможность нарушить правильную работу карты с целью вскрытия ключей шифрования, путем изменения параметров внешней среды (напряжения питания, температуры, тактовой частоты). Атакующий может увидеть при этом как корректный результат работы функции шифрования, так и ошибочный, а анализ разницы между ними позволяет провести «обратный инжиниринг» процесса

шифрования и попытаться вскрыть ключ шифрования. Такой тип атаки называется *генерацией ошибок*.

Атаки с использованием побочных каналов не вмешиваются в работу карты, они используются для получения критичной информации о работе компонентов карты. При этом атакующий смотрит, как работает карта, как она реагирует в различных ситуациях. Примером атаки с использованием побочных каналов может являться *дифференциальный анализ мощности карты* (оценка излучаемой энергии, связанной с работой карты), *электромагнитный анализ* (оценка частот излучения), анализ времени выполнения отдельных процессов в карте. Такие атаки часто применяются для сбора данных. Атакующий ведет мониторинг и перехватывает аналоговые характеристики всех источников питания, соединительных интерфейсов, а также иные электромагнитные излучения, вырабатываемые процессором карты в процессе выполнения обычных операций. Также атакующий может собрать информацию о времени, которое требуется карте для выполнения различных своих функций. На основании собранных данных атакующий может сделать много выводов и в отдельных случаях получить закрытый ключ, критичную информацию или ключ шифрования, сохраненный на карте.

В настоящее время, недостатки совместимости смарт-карт являются большой проблемой. Хотя производители заявляют, что их продукты «полностью совместимы с ISO/IEC 14443», часто они разрабатываются на основе собственных технологий и методов производителя.

Совместимость. Стандарт ISO/IEC 14443 описывает следующие элементы стандартизации смарт-карт:

- ISO/IEC 14443-1 Физические характеристики
- ISO/IEC 14443-3 Инициализация и предотвращение коллизий
- ISO/IEC 14443-4 Протокол передачи

Атаки на программное обеспечение также не вмешиваются в работу карты. Смарт-карты, как и другие устройства, имеют программное обеспечение, позволяющее им обрабатывать данные. Однако любое программное обеспечение может содержать недостатки и уязвимости, которые можно использовать. И программное обеспечение смарт-карт – не исключение. Основной целью атаки на программное обеспечение смарт-карты является ввод в карту специально сформированных команд, позволяющих атакующему извлечь информацию об учетной записи, достаточную для проведения мошеннических операций. Многие из этих видов атак могут производиться с помощью устройств, замаскированных под обычные легитимные устройства (например, считыватели), используемые для работы со смарт-картами.

Можно также попытаться провести микроисследование карты, удалив с помощью ультразвуковой вибрации защитный материал с электронных компонентов карты. Сделав это, можно получить прямой доступ к информации, хранящейся в постоянной памяти (ROM) карты.

Ссылки по теме:

- NIST Smart Card Standards and Research web page
- Smart Card Alliance home page
- “Smart Cards: A Primer,” by Rinaldo Di Giorgio, JavaWorld (Dec. 1997)
- “What Is a Smart Card?” HowStuffWorks.com

3.2. Авторизация

Хотя процессы аутентификации и авторизации совершенно различны, вместе они составляют двухступенчатый процесс, определяющий, может ли пользователь получить доступ к определенному ресурсу. На первом шаге этого процесса, аутентификации,

пользователь должен доказать системе, что он является именно тем, кем представляется, т.е. разрешенным пользователем системы. После успешной аутентификации, система должна выяснить, уполномочен ли пользователь на доступ к определенному ресурсу и какие действия с этим ресурсом он может выполнять.

Авторизация является одним из основных компонентов любой операционной системы, аналогичную функциональность могут предоставлять различные приложения, дополнительные пакеты безопасности и сами ресурсы. Когда пользователь, после прохождения процедуры аутентификации, хочет обратиться, например, к таблице на файловом сервере, этот сервер проверяет, имеет ли пользователь доступ к этой таблице. Сервер также проверяет, может ли пользователь изменить, удалить, переместить или скопировать файл с таблицей. Решение о предоставлении пользователю доступа к ресурсу основывается на критериях доступа, которые являются основой авторизации.

Критерии доступа

Мы подошли к основам управления доступом. Субъект может иметь очень детализированные права доступа к объекту или ресурсу. Это хорошо для сетевых администраторов и специалистов по безопасности, т.к. они хотят иметь максимальный контроль над ресурсами, а высокая детализация настройки прав доступа позволяет предоставить пользователям в точности те права, которые им необходимы. Однако существуют системы и приложения, в которых отсутствуют детальные настройки прав доступа, что вынуждает администратора предоставлять всем пользователям полные права доступа. Это не обеспечивает какой-либо защиты.

Предоставление прав доступа субъектам должно быть основано на уровне доверия этому субъекту компанией и на принципе «необходимо знать» (need-to-know). Только тот факт, что компания полностью доверяет пользователю, еще не означает, что ему «необходимо знать» содержимое всех ресурсов компании. И, наоборот, если пользователю «необходимо знать» определенные файлы, которые нужны ему для работы, это не означает, что компания доверяет ему доступ ко всем остальным файлам. Эти вопросы должны быть определены и включены в критерии доступа (access criteria). Различные критерии доступа могут определяться ролями, группами, местонахождением, временем и типами транзакций.

Использование **ролей** является эффективным способом распределения прав доступа на основе должностных обязанностей или функций пользователей. Если, например, в компании существует должность, в обязанности которой входит аудит транзакций и лог-файлов, соответствующая роль должна иметь доступ только на чтение и только к определенным файлам, содержащим необходимую информацию. Этой роли не нужны права на редактирование или удаление этих файлов.

Использование **групп** – другой эффективный способ распределения прав доступа. Если некоторым пользователям нужны одинаковые права доступа к информации и ресурсам, целесообразно поместить их учетные записи в одну группу, поскольку значительно проще управлять правами доступа и разрешениями группы, чем каждого пользователя в отдельности. Например, если права доступа к принтеру предоставлены только группе «Бухгалтерия», то когда пользователь попытается печатать на нем, система в первую очередь проверит, является ли он членом этой группы. Это один из способов обеспечения управления доступом посредством механизма логического управления доступом.

Физическое или логическое местонахождение также может быть использовано для ограничения доступа к ресурсам. Например, доступ к некоторым файлам может быть предоставлен только пользователям, получившим локальный доступ к компьютеру. В частности, можно таким способом ограничить доступ к конфигурационным файлам некоторых серверов, чтобы снизить риски изменения настроек неуполномоченными пользователями – настройки таких серверов могут быть изменены только авторизованными

пользователями непосредственно с локальной консоли сервера, а не через удаленный доступ по сети. Также ограничения могут быть введены на основе сетевых адресов. Например, администратор может ограничить удаленный доступ к системе выявления вторжений и разрешить подключение к ней только с определенных адресов компьютеров в сети.

Время дня – это другой механизм, который может использоваться для управления доступом. Например, можно установить, что доступ к файлам с платежной информацией запрещен с 20-00 до 04-00. Или можно настроить запрет на выполнение банковских транзакций в нерабочие для банка дни. Также существуют возможности установления ограничений по доступу в зависимости от времени создания ресурса – например, пользователю может быть предоставлен доступ только к тем файлам, дата создания которых превышает дату его приема на работу. При этом доступ к файлам, созданным ранее, для этого пользователя будет запрещен.

Ограничения по **типу транзакций** можно использовать для того, чтобы разрешить доступ только к тем данным и командам, которые необходимы в процессе выполнения определенной транзакции. Например, клиент, при использовании системы интернет-банкинга, может провести операцию на сумму до \$2000. Если ему нужно провести операцию на большую сумму, ему нужен специальный код администратора. Еще один пример – администратор базы данных может создать базу данных для Отдела кадров, но он не может читать некоторые конфиденциальные таблицы или поля в этой базе данных. Все это примеры ограничений по типу транзакций, предназначенные для управления доступом к данным и ресурсам.

Отсутствие доступа «по умолчанию»

Механизмы управления доступом следует настроить таким образом, чтобы «по умолчанию» доступ ни к каким ресурсам не предоставлялся («no access»). Это позволит избежать многих незаметных «дыр» в безопасности. В операционных системах и приложениях существуют многочисленные варианты прав доступа, которые могут быть предоставлены пользователям и группам, например, «чтение», «запись», «удаление», «полный доступ», «нет доступа». Пока пользователю не предоставлены права индивидуально или пока он не включен в какую-либо специальную группу, у него не должно быть никаких прав доступа к ресурсам, т.е. пользователю должен быть запрещен доступ ко всем ресурсам, кроме тех, к которым он явно разрешен. Другими словами, все управление доступом должно быть основано на концепции старта с нулевым доступом и добавления прав по мере необходимости в соответствии с принципом «необходимо знать».

Например, большинство списков контроля доступа (ACL) на маршрутизаторах и межсетевых экранах с пакетной фильтрацией не предоставляют доступа «по умолчанию».

«Расползание прав доступа». Сотрудники работают в компании долгое время и переходят из одного подразделения в другое, получая при этом все больше и больше прав доступа и полномочий. Часто это называют «расползанием прав доступа» (authorization creep). Это может быть большим риском для компании, т.к. в результате слишком многие пользователи имеют слишком привилегированный доступ к активам компании. Администраторам гораздо проще предоставить пользователям излишние права доступа, чем недостаточные, т.к. в этом случае пользователи не возвращаются с претензиями и не просят администратора еще поработать над их профилем. К тому же часто очень трудно определить точный перечень прав доступа, необходимых конкретному человеку. Именно поэтому управление пользователями и инициализация пользователей стали больше превалировать в современных продуктах управления идентификацией, и по этой же причине компании переходят к реализации ролевого управления доступом.

Одной из важных задач является обеспечение минимума привилегий для учетных записей пользователей, чтобы убедиться, что компания не подвергает сама себя излишнему риску. Для этого следует провести полный пересмотр имеющихся прав доступа и привилегий пользователей. Целесообразно совместить это с процессами, вызванными введением новых требований регуляторов. Например, одним из требований SOX является пересмотр руководителями прав доступа их сотрудников на ежегодной основе.

Принцип «необходимо знать»

Принцип *«необходимо знать»* (need-to-know) похож на принцип *наименьших привилегий* (least-privilege). Принцип «необходимо знать» говорит о том, что человек должен иметь доступ только к той информации, которая ему совершенно необходима для выполнения своих должностных обязанностей. Предоставление излишних полномочий ведет к излишним проблемам и, вероятно, злоупотреблениям. Руководство должно решить, что пользователь должен знать или какие права доступа ему необходимы, а администратор должен настроить механизмы управления доступом таким образом, чтобы пользователь имел минимальный набор необходимых ему прав доступа, достаточный для выполнения пользователем своих обязанностей, и не более того.

Например, руководство решило, что Дэну, выполняющему обязанности по распечатке информации, необходимо знать, где хранятся файлы, подлежащие распечатке, и иметь возможность распечатывать их. Это и есть критерий «необходимо знать» для Дэна.

Администратор мог бы дать Дэну полный доступ ко всем файлам, которые ему нужно печатать, однако это было бы нарушением принципа минимальных привилегий.

Администратору следует ограничить доступ Дэна только правами на чтение и печать соответствующих файлов, не больше и не меньше. Иначе кто будет отвечать за то, что Дэн случайно удалит все файлы с файлового сервера? Конечно, администратор.

Важно понимать, что определение требований по обеспечению безопасности пользователями и необходимых пользователям прав доступа не должно исходить от администратора, это задача руководства и владельцев. Администратор безопасности лишь настраивает механизмы безопасности для реализации этих требований.

Единый вход

Часто сотрудникам в течение рабочего дня требуется доступ к различным компьютерам, серверам, базам данных и другим ресурсам для выполнения своих задач. Для этого им зачастую требуется помнить несколько различных идентификаторов и паролей. В идеале, пользователь должен только один раз ввести свой идентификатор и пароль для получения возможности доступа ко всем ресурсам в сети, с которыми он работает. Однако в реальности это очень трудно реализовать.

В связи с распространением клиент-серверных технологий, сети мигрируют с централизованно управляемой среды в гетерогенную, распределенную среду.

Распространение открытых систем и разнообразие приложений, платформ, операционных систем ведет к тому, что пользователю нужно помнить несколько наборов учетных данных для получения доступа и использования различных сетевых ресурсов. Хотя различные идентификаторы и пароли теоретически повышают уровень безопасности, на практике они чаще ведут к нарушениям безопасности (потому что пользователи не могут запомнить много паролей, пишут их на стикерах и приклеивают к монитору) и существенно повышают нагрузку на персонал, занимающийся управлением и поддержкой сети. Сотрудниками служб технической поддержки теряется очень много времени и ресурсов на сброс забытых пользователями паролей. Забытые пароли часто сказываются на продуктивности работы не только самого пользователя, но и на выполнении бизнес-процессов, в реализации которых этот пользователь участвует. Системные администраторы вынуждены управлять множеством учетных записей пользователя на различных платформах, координируя эту деятельность для обеспечения целостности политики безопасности. С возрастанием сложности этих работ, растет число недостатков в управлении правами доступа, а вместе с ними растет и число уязвимостей. Таким образом, дополнительные пароли чаще всего не ведут к дополнительной безопасности.

Повышение стоимости поддержки различных сред, соображения безопасности, пожелания пользователей – все вместе они привели к идее создания возможностей единого входа (SSO –

single sign-on). SSO позволяет пользователям вводить свои учетные данные только один раз для получения возможности доступа ко всем ресурсам, что уменьшает время, которое пользователи тратят на процесс аутентификации. Также это позволяет администраторам рационализировать использование учетных записей и лучше управлять распределением прав доступа. SSO повышает безопасность, снижая вероятность того, что пользователь будет хранить свой пароль под клавиатурой – ведь теперь ему нужно запомнить гораздо меньше паролей (в идеале – один). Помимо этого, на повышение безопасности оказывает влияние и тот факт, что администраторы больше времени могут тратить на управление учетными записями и правами доступа пользователей, а не на сброс забытых ими паролей. Используя SSO, администратору проще приостановить или заблокировать учетную запись пользователя, т.к. нет необходимости делать это в каждой системе.

Однако в реальности у технологий SSO есть и ряд проблем. Основной проблемой SSO являются вопросы совместимости с различными системами – ведь она должна работать на любой платформе, в любом приложении и с любым ресурсом, используя для этого одни и те же учетные данные, в одинаковом формате. Другой проблемой SSO является тот факт, что в случае компрометации всего одной учетной записи, злоумышленник может получить доступ ко всем ресурсам, к которым имеет доступ скомпрометированная учетная запись. Однако использование пользователем только одного пароля позволяет рассчитывать на то, что он будет использовать сложный пароль, который злоумышленнику будет непросто подобрать.

Существуют различные виды технологий SSO. Каждый вид имеет свои преимущества и недостатки. На самом деле, встретить реальную среду SSO можно не часто, гораздо чаще встречаются группы серверов и ресурсов, которые принимают одни и те же учетные данные, что только увеличивает объем работы для администраторов.

Kerberos

Kerberos (Цербер) – это имя трехголовой собаки в греческой мифологии, которая охраняла вход в подземный мир. Этим именем названа технология безопасности, которая реализует функции аутентификации и защищает активы компании. Kerberos – это протокол аутентификации, разработанный в середине 1980-х годов, как часть Проекта Афина MIT. Он работает на базе модели клиент-сервер и основан на криптографии с симметричным ключом. Этот протокол годами использовался в системах Unix, а в настоящее время применяется как стандартный метод аутентификации в современных серверных операционных системах Microsoft Windows. Операционные системы Apple Mac OS X, Sun Solaris и Red Hat Enterprise Linux также используют аутентификацию Kerberos. Все чаще появляются коммерческие продукты, поддерживающие Kerberos.

Kerberos – это пример системы SSO для распределенных сред и стандарт «де-факто» для гетерогенных сетей. Kerberos объединяет широкий круг технологий безопасности, предоставляя компаниям больше гибкости и масштабируемости при создании архитектуры безопасности. Он имеет четыре элемента, необходимые для корпоративного управления доступом: масштабируемость, прозрачность, надежность и безопасность. Однако эта открытая архитектура имеет некоторые проблемы совместимости. В основном они связаны с тем, что производители пользуются свободой настройки протокола, и каждый из них настраивает его по-своему, что и вызывает эти проблемы.

Kerberos использует криптографию с симметричным ключом и обеспечивает «сквозную» (end-to-end) безопасность. Хотя Kerberos использует пароли для аутентификации, он спроектирован таким образом, чтобы исключить необходимость передачи паролей по сети. Большинство реализаций Kerberos работает с общими секретными ключами.

Основные компоненты Kerberos

Центр распространения ключей (KDC – Key Distribution Center) – это самый важный компонент в среде Kerberos. Он хранит секретные ключи всех пользователей и служб. KDC

реализует службу аутентификации и функции распространения ключей. Клиенты и службы доверяют целостности KDC, это доверие является основой безопасности Kerberos.

KDC предоставляет сервисы безопасности **системным объектам** (principal), которые могут быть пользователями, приложениями или сетевыми службами. KDC должен иметь учетную запись для каждого системного объекта и общий секретный ключ с каждым системным объектом. Для пользователей пароли преобразуются в значение секретного ключа.

Секретный ключ используется для передачи критичной информации между системным объектом и KDC, а также для аутентификации пользователя.

Служба предоставления билетов (TGS – Ticket Granting Service) на KDC генерирует и передает системному объекту (например, пользователю) билет (ticket), когда ему нужно аутентифицироваться на другом системном объекте (например, сервере печати).

Предположим, что Эмили нужно воспользоваться сервером печати. Для этого она должна доказать серверу печати, что она та, за кого себя выдает, и что она уполномочена использовать службу печати. Эмили запрашивает билет в TGS. TGS выдает Эмили билет, который она отправляет серверу печати. Если сервер печати принимает этот билет, Эмили разрешается использовать сервер печати.

KDC предоставляет сервисы безопасности набору системных объектов. Этот набор называется **областью** (realm) в Kerberos. KDC – это доверенный сервер аутентификации для всех пользователей, приложений и служб в рамках области. Один KDC может отвечать за одну или несколько областей. Области позволяют администратору логически группировать ресурсы и пользователей.

Итак, теперь мы знаем, что системным объектам (пользователям и службам) необходимы сервисы KDC для аутентификации друг друга, что KDC имеет базу данных, заполненную информацией о каждом системном объекте в рамках области, что KDC хранит и доставляет криптографические ключи и билеты, а также что билеты используются системными объектами, чтобы аутентифицировать друг друга. Но как работает весь этот процесс?

Процесс аутентификации Kerberos

Пользователь и KDC имеют один общий секретный (симметричный) ключ, а служба и KDC имеют другой общий секретный ключ. Пользователь и необходимая ему служба на первоначальном этапе не имеют общего симметричного ключа. Пользователь доверяет KDC, т.к. разделяет с ним свой секретный ключ. Они могут зашифровывать и расшифровывать данные, передаваемые друг другу, имея, таким образом, защищенный коммуникационный канал. После того, как пользователь аутентифицируется в службе, они также получают общий симметричный (сеансовый) ключ, который позволяет им зашифровывать и расшифровывать информацию, которой им нужно обмениваться. Именно таким образом Kerberos обеспечивает защиту передаваемых данных.

Выполняются следующие шаги:

1. Эмили приходит на работу в 8 часов и вводит свое имя и пароль на своей рабочей станции.
2. Программное обеспечение Kerberos на рабочей станции Эмили отправляет ее имя службе аутентификации (AS – Authentication Service) на KDC, которая в свою очередь отправляет ей билет для получения билета (TGT – Ticket Granting Ticket) зашифрованный на пароле Эмили (секретном ключе, хранящемся в KDC).
3. Если Эмили ввела правильный пароль, она сможет расшифровать TGT и получить доступ к рабочему столу своей локальной рабочей станции.
4. Когда Эмили потребуется отправить задание на сервер печати, ее система отправит TGT службе предоставления билетов (TGS), запущенной на KDC. Это позволит Эмили доказать, что она была аутентифицирована и позволит ей запросить доступ к

серверу печати.

5. TGS создает и отправляет второй билет Эмили, который она будет использовать для аутентификации на сервере печати. Этот второй билет содержит два экземпляра одного и того же сеансового ключа, один из которых зашифрован на секретном ключе Эмили, а второй – на секретном ключе сервера печати. Также, в этот второй билет входит **аутентификатор**, содержащий идентификационную информацию Эмили, IP-адрес ее системы, порядковый номер и штамп времени.
6. Система Эмили получает второй билет, расшифровывает его и извлекает сеансовый ключ, добавляет в него второй аутентификатор, содержащий идентификационную информацию, и отправляет билет серверу печати.
7. Сервер печати получает билет, расшифровывает его, извлекает сеансовый ключ, а затем расшифровывает и извлекает два аутентификатора. Если сервер печати может расшифровать и извлечь сеансовый ключ, он может быть уверен, что билет создал именно KDC, т.к. только KDC имеет секретный ключ этого сервера печати. Если информация из аутентификаторов (один из которых помещен в билет KDC, а другой – пользователем) совпадает, сервер печати может быть уверен, что билет получен от корректного системного объекта (пользователя).
8. После завершения этого процесса, Эмили считается успешно аутентифицированной на сервере печати, и сервер печатает ее документ.

Это очень простой пример, но он дает основную идею организации работы Kerberos при взаимодействии с сетевыми ресурсами. Посмотрите на рисунок 2-6, на котором схематически изображен этот процесс.



Рисунок 2-6. Для использования запрашиваемого ресурса, пользователь должен сначала получить билет от KDC

Служба аутентификации является частью KDC, аутентифицирующей системные объекты. TGS также является частью KDC, она создает билеты и отправляет их системным объектам. TGT позволяет пользователю не вводить свой пароль каждый раз, когда ему нужно взаимодействовать с другим системным объектом. После того, как пользователь ввел пароль, TGT временно сохраняется на его системе и пользователь может в любое время взаимодействовать с любым системным объектом, повторно используя этот TGT.

Необходимо понимать различия между сеансовым и секретным ключами. Секретный ключ является статическим, а сеансовый ключ генерируется при создании сеанса и уничтожается после его окончания. Секретный ключ совместно используется KDC и системным объектом, а сеансовый ключ – двумя системными объектами.

Если реализация Kerberos настроена на использование **аутентификатора**, пользователь

отправляет серверу печати свою идентификационную информацию, штамп времени (timestamp) и порядковый номер (sequence number), шифруя всю эту информацию на общем сеансовом ключе. Сервер печати расшифровывает полученную информацию и сравнивает ее с идентификационными данными запрашивающего пользователя, полученными от KDC. Если данные совпадают, сервер печати принимает задание от пользователя. При этом штамп времени помогает противодействовать атаке повтора (replay attack). Сервер печати сравнивает штамп времени в полученном пакете со своим собственным внутренним временем, если время существенно различается, это может означать, что сетевой пакет был перехвачен атакующим, а затем отправлен повторно с целью получения несанкционированного доступа от имени легитимного пользователя. Также сервер печати проверяет порядковый номер, чтобы убедиться, что этот пакет не был получен ранее. Это другая защитная мера, позволяющая противостоять атаке повтора.

Основная причина использования Kerberos заключается в том, что системные объекты недостаточно доверяют друг другу для прямого взаимодействия. В нашем примере, сервер печати не намерен печатать чьи угодно задания, без проведения предварительной аутентификации пользователя. Однако нет системных объектов, которые доверяют друг другу напрямую, они доверяют только KDC. KDC создает билеты, ручаясь таким образом за определенный системный объект, которому нужно взаимодействовать.

Такой же тип модели доверия используется в среде PKI (более подробная информация о PKI представлена в Домене 6). В среде PKI пользователи также не доверяют друг другу напрямую, но они доверяют удостоверяющему центру, который ручается за личность пользователей с помощью цифровых сертификатов.

Kerberos является примером технологии SSO – пользователь вводит свой идентификатор и пароль только один раз. Билеты имеют ограниченное время жизни, которое настраивается администратором. Чаще всего время жизни TGT составляет от 8 до 10 часов, чтобы, когда пользователь придет на работу на следующий день, он представил свои учетные данные снова.

Недостатки Kerberos

Ниже представлено несколько потенциальных недостатков Kerberos:

- KDC может являться единой точкой отказа (single point of failure). Если KDC недоступен, никто не сможет получить доступ к ресурсам. Поэтому KDC требует избыточности.
- KDC должен быть способен одновременно обрабатывать большое количество запросов. Поэтому он должен быть масштабируемым.
- Секретный ключ временно хранится на рабочей станции пользователя, что может привести к его компрометации.
- Сеансовый ключ хранится на рабочей станции пользователя в открытом виде в кэше или таблице ключей, что может привести к его компрометации.
- Kerberos уязвим к подбору паролей, он не сможет узнать, что атакующим производится подбор пароля по словарю.
- Kerberos не защищает сетевой трафик, если не включено шифрование.
- Если ключ очень короткий, он может быть уязвим к атаке «грубой силы» (brute force attack).
- Kerberos требует, чтобы системные часы на всех клиентах и серверах были синхронизированы.

Kerberos должен быть прозрачным (работать в фоновом режиме и не требовать от

пользователя понимать это), масштабируемым (работать в большой, гетерогенной среде), надежным (использовать распределенную серверную архитектуру, чтобы гарантировать отсутствие единой точки отказа) и безопасным (обеспечивать аутентификацию и конфиденциальность).

Kerberos и атака подбора пароля. Один только факт, что в среде применяется Kerberos, не говорит о том, что эта среда уязвима к атакам подбора пароля. Сама операционная система должна обеспечивать соответствующую защиту и отслеживать попытки регистрации. Протокол Kerberos не имеет такой функциональности, поэтому должны использоваться другие компоненты для противодействия этому типу атак.

Ссылки по теме:

- Kerberos FAQ
- MIT Kerberos Papers and Documentation web page

SESAME

Проект **SESAME** (Secure European System for Applications in a Multi-vendor Environment) – это технология SSO, расширяющая функционал Kerberos и устраняющая его недостатки. В отличие от Kerberos, SESAME использует симметричные и асимметричные технологии криптографии для аутентификации субъектов на сетевых ресурсах.

Kerberos использует билеты для аутентификации субъектов и объектов, а SESAME использует для этого сертификаты атрибутов привилегий (PAC – Privileged Attribute Certificate), которые содержат идентификационные данные субъекта, его возможности доступа к объекту, период времени доступа и время жизни PAC. PAC подписывается ЭЦП, которую объект может проверить с помощью доверенного сервера аутентификации, называемого сервером атрибутов привилегий (PAS – Privileged Attribute Server). PAS выполняет роль, похожую на роль KDC в Kerberos. После успешной аутентификации пользователя в службе аутентификации (AS), он предоставляет токен, полученный от PAS. Затем PAS создает PAC для пользователя, который пользователь будет предоставлять ресурсам, доступ к которым ему потребуется. На рисунке 2-7 показана схема работы основных элементов SESAME.



Рисунок 2-7. SESAME очень похож на Kerberos

Ссылки по теме:

- SESAME in a Nutshell
- SESAME links

Домены безопасности

Термин **домен** (domain) часто связывают с Microsoft, когда люди слышат этот термин, они представляют себе группу компьютеров и устройств в сетевом сегменте, управляемых сервером, на котором запущено программное обеспечение Microsoft, называемом контроллером домена. В действительности, домен – это просто набор ресурсов, доступных субъекту. Помните, что субъект может быть пользователем, процессом или приложением. В рамках операционной системы, процесс имеет домен, который является набором системных ресурсов, доступных процессу для выполнения им своих задач. Этими ресурсами могут быть сегменты памяти, пространство на жестком диске, службы операционной системы и другие процессы. В сетевой среде, домен является набором доступных физических и логических ресурсов, которыми могут быть маршрутизаторы, файловые серверы, службы FTP, веб-серверы и т. д.

Термин **домен безопасности** (security domain) основывается на определении домена, добавляя к нему факт, что ресурсы в рамках этой логической структуры (домена) работают с одной и той же политикой безопасности и управляются одной группой. Таким образом, администратор может поместить компьютеры, учетные записи и сетевые ресурсы сотрудников бухгалтерии в Домен 1, а компьютеры, учетные записи и сетевые ресурсы руководства в Домен 2. Все эти элементы попадут в эти два контейнера, поскольку они (элементы) не только выполняют однотипные задачи, но также, что более важно, имеют один и тот же уровень доверия. Общий уровень доверия позволяет управлять этими элементами одной (отдельной) политикой безопасности.

Отдельные домены разделяются логическими границами, такими как межсетевые экраны, с ACL, службы каталогов, принимающие решения о предоставлении доступа, объекты, имеющие собственные ACL, которые указывают, какие пользователи могут работать с ними. Все эти механизмы безопасности являются примером компонентов, обеспечивающих реализацию политики безопасности для каждого домена.

Домены могут быть спроектированы в виде иерархической структуры, определяющей взаимоотношения между различными доменами и способы взаимодействия субъектов, находящихся в различных доменах. На рисунке 2-8 показан пример иерархии сетевых доменов. Их коммуникационные каналы управляются агентами безопасности (списками контроля доступа межсетевых экранов и маршрутизаторов, службами каталогов) и отдельными доменами, изолированными с помощью различных масок подсетей.

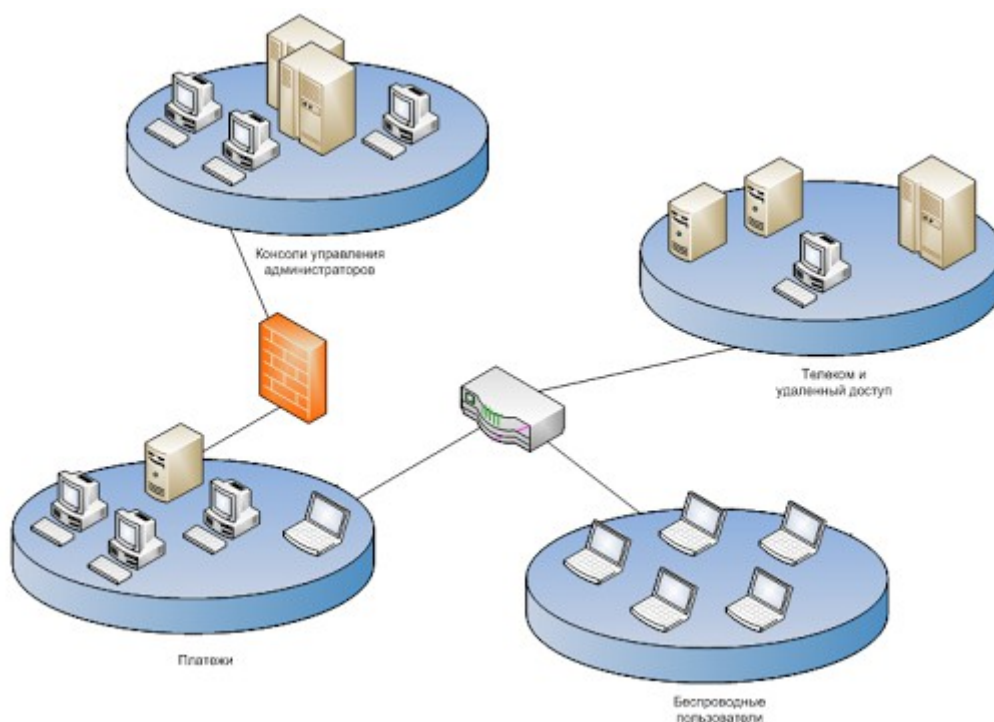


Рисунок 2-8.Сетевые домены используются для разделения различных сетевых сегментов

Помните, что домены не обязательно связаны с сетевыми устройствами и сегментацией, они могут также применяться к пользователям и процессам. На рисунке 2-9 показано, как с пользователями и процессами могут быть связаны более детальные домены на основе уровней доверия. Группа 1 имеет высокий уровень доверия и может использовать доступ как к домену своего уровня доверия (Домен 1), так и к домену более низкого уровня доверия (Домен 2). Пользователь 1, который имеет низкий уровень доверия, может использовать только домен своего уровня доверия и не выше. Система реализует эти домены с помощью привилегий доступа и прав на уровне файловой системы и ядра операционной системы.

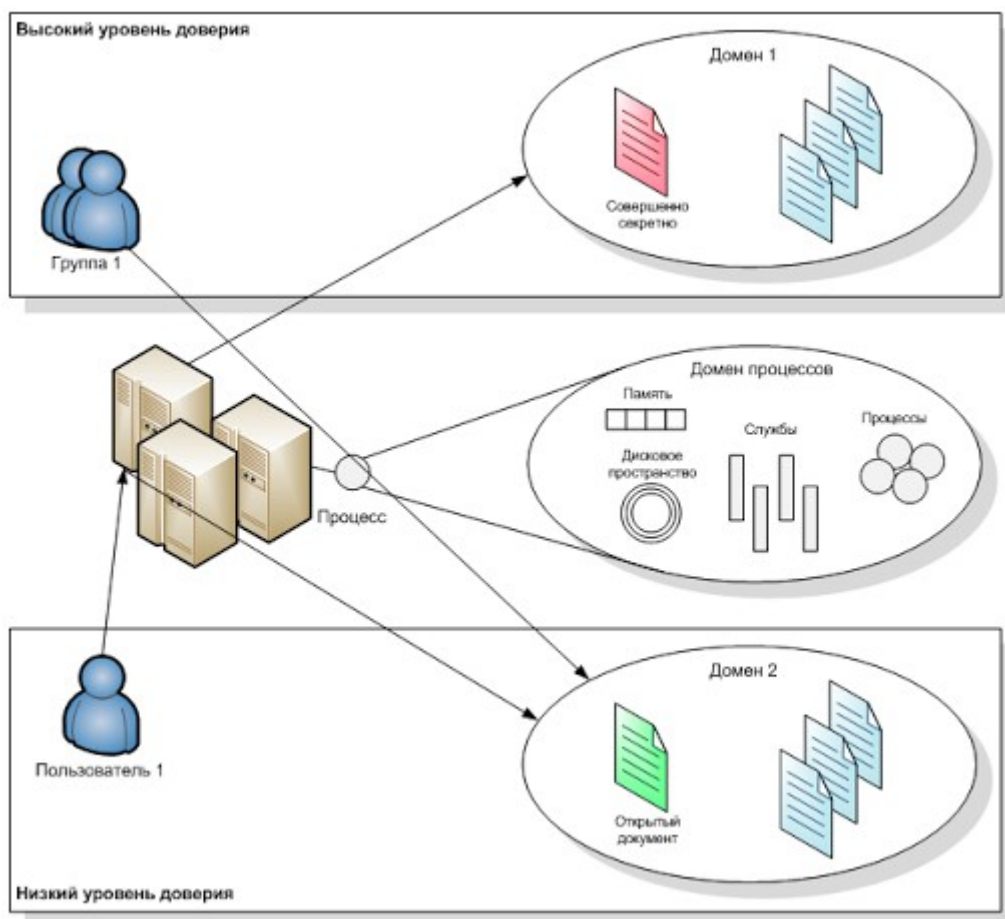


Рисунок 2-9. Субъекты могут использовать доступ к доменам на основе уровней доверия

Почему домены включены в раздел «Единый вход»? Потому что различные виды существующих в настоящее время технологий, использующихся для определения и реализации этих доменов и связанных с ними политик безопасности, направлены на предоставление пользователям (субъектам) возможности проходить аутентификацию только один раз и получать при этом возможность доступа к различным доменам без необходимости повторного ввода учетных данных. Примерами таких технологий являются: контроллеры домена в среде Windows, продукты ERM, Microsoft Passport и различные продукты, предоставляющие функциональность SSO.

Ссылки по теме:

- *Underlining Technical Models for Information Technology Security, Recommendations of the National Institute of Standards and Technology*, by Gary Stoneburner, NIST Special Publication 800-33 (Dec. 2001)
- “New Thinking About Information Technology Security,” by Marshall Abrams, PhD and Michael Joyce (first published in *Computers & Security*, Vol. 14, No. 1, pp. 57–68)

Службы каталогов

Служба каталога – это механизм, который идентифицирует ресурсы (принтеры, файловые серверы, контроллеры доменов и периферийные устройства) в сети. Сетевая служба каталога (network directory service) содержит информацию об этих ресурсах и субъектах, которым нужен доступ к ним, и выполняет действия по управлению доступом. Если служба каталога работает на основе базы данных, построенной в соответствии со стандартом X.500, она работает в иерархической схеме, которая описывает атрибуты ресурсов, такие как имя, логическое и физическое размещение, субъектов, которые могут использовать эти ресурсы, и операции, которые субъекты могут выполнять с ними.

В базах данных, построенных в соответствии со стандартом X.500, запросы на доступ выполняются пользователями и другими системами посредством протокола LDAP. База данных такого типа обеспечивает иерархическую структуру для организации объектов (субъектов и ресурсов). Служба каталога присваивает уникальные имена каждому объекту и, при необходимости, добавляет соответствующие ему атрибуты. Служба каталога реализует политику безопасности (настроенную администратором) для управления взаимодействием субъектов и объектов.

Сетевые службы каталогов предоставляют пользователям доступ к сетевым ресурсам прозрачно, т.е. пользователям не нужно знать конкретное местоположение ресурсов или шаги, которые необходимо выполнить для доступа к ним. Сетевые службы каталогов выполняют эти задачи для пользователей в фоновом режиме. Примерами служб каталогов являются LDAP (Lightweight Directory Access Protocol), Microsoft AD (Active Directory), Novell NDS (NetWare Directory Service).

ПРИМЕЧАНИЕ. Службы каталогов также рассматривались в разделе «Управление идентификацией» ранее в этом Домене.

Тонкие клиенты

Бездисковые компьютеры (diskless computer) и тонкие клиенты (thin clients) не могут хранить много информации, т.к. они не имеют встроенных устройств хранения информации и необходимых ресурсов. Этот вид клиент-серверной технологии позволяет пользователям регистрироваться на центральном сервере для использования компьютера и доступа к сетевым ресурсам. Когда пользователь включает компьютер, он выполняет короткий список команд и затем обращается на сервер, чтобы загрузить операционную систему или работать в терминальном режиме с прикладным программным обеспечением, работающим на сервере. Это позволяет реализовать строгий контроль доступа, т.к. компьютер пользователя не может сделать ничего самостоятельно, пока не аутентифицируется на центральном сервере и не получит с сервера операционную систему, профиль и функционал. Технология тонких клиентов предоставляет собой другой тип технологии SSO для пользователей, позволяя им аутентифицироваться только на центральном сервере или мейнфрейме, который затем предоставляет им доступ ко всем необходимым и разрешенным им ресурсам.

Помимо предоставления решения SSO, технология тонких клиентов имеет ряд других преимуществ. Компания может сэкономить, покупая тонкие клиенты вместо более дорогих полноценных компьютеров. Все приложения выполняются на центральном сервере, на нем же выполняется обработка и хранение данных. При этом тонкий клиент обеспечивает только отображение графической оболочки и передает на центральный сервер нажатия клавиш и перемещения «мыши». Тот факт, что все программное обеспечение находится в одном месте, а не распределено по всей сети, позволяет упростить администрирование, централизованно управлять доступом, упростить процесс обновлений, стандартизовать конфигурации. Также это упрощает защиту от вредоносного программного обеспечения и утечек конфиденциальной информации, т.к. на многих тонких клиентах отсутствуют приводы для работы с компакт-дисками и порты USB.

ПРИМЕЧАНИЕ. Эта технология пришла от централизованной модели, использовавшейся мейнфреймами и простыми терминалами (dumb terminal). Она реализуется с помощью службы терминалов Windows, программного обеспечения Citrix, сервис-ориентированной архитектуры (SOA – Service Oriented Architecture) и т.д.

Примеры технологий SSO:

- **Kerberos.** Протокол аутентификации, использующий KDC и билеты, который основан на криптографии с симметричным ключом.
- **SESAME.** Протокол аутентификации, использующий PAS и PAC, который основан на симметричной и асимметричной криптографии.
- **Домены безопасности.** Работа ресурсов в рамках одной политики безопасности и

управляемых одной группой.

- **Тонкие клиенты.** Терминалы, которые используют центральный сервер для управления доступом, обработки и хранения данных.

4. Модели управления доступом

Модель управления доступом – это структура, которая определяет порядок доступа субъектов к объектам. Для реализации правил и целей этой модели используются технологии управления доступом и механизмы безопасности. Существует три основных модели управления доступом: дискреционная, мандатная и недискреционная (также называемая ролевой). Каждая модель использует различные методы для управления доступом субъектов к объектам, каждая имеет свои преимущества и ограничения. Выбор оптимальной модели управления доступом следует производить на основе целей бизнеса и целей безопасности компании, а также на основе ее культуры и стиля управления бизнесом. Некоторые компании используют только одну модель, другие комбинируют их для получения необходимого уровня защиты.

Эти модели встроены в ядро различных операционных систем и во многих случаях поддерживаются приложениями. Каждая операционная система имеет ядро безопасности, которое реализует концепцию монитора обращений (reference monitor), которая зависит от встроенной в систему модели управления доступом. Для каждой попытки доступа, перед тем, как субъект сможет начать взаимодействовать с объектом, ядро безопасности проверяет правила модели управления доступом, чтобы определить, является ли запрос допустимым. В следующих разделах будет рассказано про эти различные модели доступа, поддерживающие их технологии и о том, когда их следует применять.

4.1. Дискреционное управление доступом

Если пользователь создает файл, он является владельцем этого файла. Идентификатор этого пользователя размещается в заголовке файла. Владение может быть также предоставлено определенному человеку. Система, которая использует **дискреционное (избирательное) управление доступом** (DAC – Discretionary Access Control) позволяет владельцу ресурса определять, какие субъекты могут использовать этот ресурс. Эта модель называется дискреционной (избирательной), т.к. управление доступом основано на решениях владельца. Часто руководители подразделений являются владельцами данных в рамках своих подразделений. Будучи владельцами, они могут решать, кому следует, а кому не следует иметь доступ к этим данным.

В модели DAC ограничения доступа основываются на авторизации пользователя. Это означает, что владельцы могут определять, какой тип доступа может быть разрешен к их объектам. Если компания использует модель DAC, сетевой администратор может разрешить владельцам ресурсов управлять доступом пользователей к своим ресурсам. Чаще всего модель DAC реализуется посредством списков контроля доступа (ACL), содержимое которых определено владельцами. Работа ACL реализуется средствами операционной системы. Это может позволить пользователям использовать информацию динамически вместо более статичного мандатного или ролевого управления доступом. Большинство операционных систем основаны на модели DAC (например, системы Windows, Linux, Macintosh и большинство систем *nix).

DAC может быть применен как к древовидной структуре директорий, так и к файлам, которые в них содержатся. Мир персональных компьютеров использует такие разрешения доступа, как «Нет доступа» (No Access), «Чтение» (Read), «Запись» (Write), «Выполнение» (Execute), «Удаление» (Delete), «Изменение» (Change), «Полный доступ» (Full Control). К примеру, атрибут «Чтение» позволяет читать файл, но не вносить в него изменения; атрибут «Изменение» позволяет читать, записывать, выполнять и удалять файл, но не менять его ACL или владельца файла; атрибут «Полный доступ» позволяет производить любые

действия с файлом, разрешениями на доступ к нему и владением им.

Посредством дискреционной модели, например, Сэм может предоставить совместный доступ к диску D на своем компьютере Дэвиду, а Дэвид может скопировать с него все MP3 Сэма. При этом Сэм может заблокировать доступ к своему диску D для своего начальника, чтобы тот не знал, что Сэм тратит время и ресурсы на скачивание музыки и ее раздачу своим друзьям.

Ссылки по теме:

- A Guide To Understanding Discretionary Access Control in Trusted Systems
- Mandatory Versus Discretionary Access Control
- Discretionary Access Control

Управление доступом на основе идентификатора. Системы DAC предоставляют, либо отказывают в доступе на основе идентификации субъекта. Идентификация может быть как на уровне пользователя, так и на уровне его членства в группе. Так, например, владелец данных может предоставить доступ к своему файлу Бобу (идентификатор пользователя) и группе «Бухгалтерия» (идентификатор группы).

4.2. Мандатное управление доступом

В модели **мандатного управления доступом** (MAC – Mandatory Access Control) пользователи и владельцы данных не могут самостоятельно определять, кто может иметь доступ к файлам. Окончательное решение принимает операционная система, и это решение может не совпадать с желаниями пользователя. Эта модель является более структурированной и жесткой, она основана на системе меток безопасности (security label). Пользователи получают уровни допуска (секретно, совершенно секретно, конфиденциально и т.д.), таким же способом классифицируются данные. Допуски и классы данных сохраняются в метках безопасности и являются границами для субъектов и объектов. Когда система принимает решение в процессе выполнения запроса на доступ к объекту, она основывается на уровне допуска субъекта, классификации объекта и политике безопасности системы. Правила доступа субъектов к объектам разрабатываются офицером безопасности, настраиваются администратором, реализуются операционной системой и поддерживаются технологиями безопасности.

Метки безопасности прикрепляются ко всем объектам, каждый файл, директория, устройство имеют свою метку безопасности, содержащую информацию о классе их информации. Например, если пользователь имеет уровень допуска «Секретно», а запрашивает информацию класса «Совершенно секретно», он получит отказ, поскольку его допуск не равен (и не выше) классификации.

ПРИМЕЧАНИЕ. Термины «метка безопасности» (security label), «метка критичности» и «метка чувствительности» (sensitivity labels) являются взаимозаменяемыми.

Каждый субъект и объект всегда должен иметь связанную с ним метку с атрибутами, поскольку это является частью критериев принятия решения операционной системой.

Эта модель применяется в среде, в которой классификация информации и конфиденциальность чрезвычайно важны, например, в военных организациях. На базе этой модели разработаны специализированные версии Unix-систем, например, SE Linux, Trusted Solaris. Компании не могут просто переключаться между использованием DAC и MAC. Им потребуется специально приобрести для этого операционную систему, спроектированную и реализующую правила MAC. Системы DAC не понимают меток безопасности, классификации, уровней допуска и поэтому не могут применяться в организациях, которым нужна такая структура управления доступом.

Ссылки по теме:

- Enterprise Security Policy
- Mandatory Access Control
- SELinux

Метки критичности

Когда используется модель MAC, каждый субъект и объект должен иметь метку критичности, также называемую меткой безопасности. Эта метка указывает на классификацию и различные категории. Классификация указывает на уровень критичности, а с помощью категорий реализуется принцип «необходимо знать» (need-to-know). Метка критичности показана на рисунке 2-10. Если кто-то имеет доступ к «совершенно секретной» информации, это вовсе не означает, что ему «необходимо знать» всю информацию с грифом «совершенно секретно».



Рисунок 2-10.Метка критичности состоит из классификации и категорий

Классификация использует иерархическую структуру, в которой один уровень является более доверенным, чем другой. Категории не используют иерархическую структуру, они представляют собой отдельные виды информации в рамках системы. Категории могут соответствовать структуре подразделений компании, проектам или уровням должностей. Классификация проводится по степени конфиденциальности информации, она зависит от среды, в которой работает компания.

ПРИМЕЧАНИЕ. В реализациях MAC, система принимает решение о возможности доступа, сравнивая уровень допуска субъекта и уровень «необходимо знать» с меткой безопасности. В DAC система сравнивает идентификатор субъекта со списком ACL ресурса.

Программные и аппаратные охранные средства (guard) позволяют обмениваться данными между доверенными (высокий уровень гарантий) и менее доверенными (низкий уровень гарантий) системами и средами, выполняя функции посредника между ними. Например, вы работаете на системе MAC (работающей в выделенной модели безопасности на уровне «Секретно») и вам нужно взаимодействовать с базой данных MAC (работающей в многоуровневом режиме безопасности, достигающем уровня «Совершенно секретно»). Эти две системы могут обеспечивать различные уровни защиты, и, если менее доверенная система будет напрямую взаимодействовать с более доверенной, в системе безопасности появятся уязвимости и повысятся риски компрометации. Программные охранные средства позволяют взаимодействовать системам, работающим на разных уровнях безопасности. Различные их виды могут применяться для выполнения фильтрации, обработки запросов, блокировки и обезличивания данных. Также существуют аппаратные охранные средства, представляющие собой устройства с двумя сетевыми картами, подключенными к двум разным системам, которым нужно взаимодействовать между собой. Охранные средства могут использоваться для соединения различных систем MAC или сетей, работающих в разных режимах безопасности и на разных уровнях безопасности. В большинстве случаев, менее доверенные системы могут отправлять сообщения более доверенным системам, но в обратном направлении они могут принимать только подтверждения о доставке.

4.3. Ролевое управление доступом

Модель *ролевого управления доступом* (RBAC – Role-based Access Control), также называемая недискреционным управлением доступом (Nondiscretionary Access Control), использует централизованно администрируемый набор контролей, предназначенных для определения порядка взаимодействия субъекта с объектом. Этот тип модели разрешает доступ к ресурсам, основываясь на роли пользователя в компании. Это называют недискреционным подходом, поскольку назначение пользователю роли является неизбежным. Это означает, что если вам в компании назначена роль «Подрядчик», вы ничего с этим сделать не можете. Вы не определяете самостоятельно, какая роль вам будет назначена.

Более традиционное администрирование прав доступа основано на модели DAC, в которой управление доступом происходит на уровне объекта с использованием ACL. Этот подход более сложен, т.к. администратор должен перевести организационную политику компании в разрешения при настройке ACL. С ростом количества объектов и пользователей в компании у многих пользователей появляются (или остаются после изменения обязанностей) права доступа к некоторым ресурсам, которые им не требуются для работы. Это нарушает принцип минимальных привилегий и увеличивает риски компании. Подход RBAC позволяет избежать этого, так как разрешения управляются на уровне должностных ролей. В модели RBAC роль определяется в терминах операций и задач, которые она выполняет, тогда как в модели DAC описывается, какие субъекты могут иметь доступ к каким объектам.

Например, нам нужна роль аналитика. Мы разрабатываем эту роль, позволяя ей иметь доступ ко всем видам продукции и данным тестирования, а также, что более важно, определяем задачи и операции, которые эта роль может выполнять с этими данными. Когда пользователь, которому присвоена эта роль «Аналитик», делает запрос на доступ к новым результатам тестирования на файловом сервере, операционная система в фоновом режиме просматривает уровень доступа роли, прежде чем эта операция будет разрешена.

ПРИМЕЧАНИЕ. Введение ролей показывает разницу между правами, назначенными явно и неявно. В явном виде права и разрешения назначаются непосредственно конкретному пользователю. В неявном виде они назначаются роли или группе, а пользователь просто наследует эти полномочия.

Модель RBAC лучше всего подходит для компаний с большой «текучестью» кадров. Если увольняется сотрудник, которому была назначена определенная роль, то новому сотруднику, который займет его место, просто назначается та же роль. Таким образом, администратору не нужно постоянно вносить изменения в ACL отдельных объектов. Ему достаточно создать определенные роли, настроить необходимые этим ролям права и разрешения, а затем назначить пользователям эти роли.

Ядро RBAC

Этот компонент должен быть интегрирован в каждую реализацию RBAC, т.к. это основа данной модели. Пользователи, роли, разрешения, операции и сессии определяются на основании политики безопасности.

- Имеются отношения «многие-ко-многим» между отдельными пользователями и привилегиями.
- Сессия является соответствием между пользователем и подмножеством назначенных ему ролей.
- Применяется традиционное, но более надежное управление доступом на основе групп.

Пользователи могут быть включены во многие группы, каждая из которых имеет различные привилегии. Когда пользователь входит в систему (это называется сессией или сеансом), то

ему сразу же становятся доступны все роли и группы, которые ему были назначены.

Это надежная модель, поскольку она может включать другие компоненты в процесс принятия решений о возможности доступа, а не просто основываться на учетных данных. Например, система RBAC может учитывать время дня, местоположение роли, день недели и т.д.

Иерархический RBAC

Этот компонент позволяет администратору создать организационную модель RBAC на основе организационной структуры и функциональных разграничений, требующихся в конкретной среде. Это очень полезно, поскольку в компаниях уже есть иерархические структуры персонала. Чаще всего, чем выше вы находитесь в организационной иерархии компании, тем больше прав доступа вы имеете.

1. Ролевые отношения определяют членство пользователя в группах и наследование привилегий. Например, роль «Медсестра» может получить доступ к одному набору файлов, а роль «Лаборант» – к другому. Роль «Доктор» наследует разрешения и права доступа из обеих этих ролей и имеет дополнительные права, назначенные непосредственно роли «Доктор». Таким образом, иерархия накапливает права и разрешения различных ролей.
2. Отражает организационную структуру и функциональные разграничения.
3. Существует два типа иерархий:
 - Ограниченные иерархии. Доступен только один уровень иерархии (например, Роль 1 наследует права Роли 2, но не других ролей).
 - Обычные иерархии. Доступно много уровней иерархии (например, Роль 1 наследует права Роли 2 и Роли 3).

Иерархии позволяют структурировать роли, естественным образом отражая разграничение полномочий и обязанностей в компании. Иерархии ролей определяют порядок наследования между ролями. Эта модель позволяет организовать разделение обязанностей (separation of duties).

- **Статическое разделение обязанностей (SSD – Static Separation of Duty) посредством RBAC.** Это может использоваться для предотвращения мошенничества, предоставляя постоянный ограниченный набор привилегий (например, пользователь не может быть одновременно членом групп «Кассир» и «Контролер»).
- **Динамическое разделение обязанностей (DSD – Dynamic Separation of Duties) посредством RBAC.** Это может использоваться для предотвращения мошенничества, предоставляя ограничение набора возможных привилегий в рамках одной сессии (например, пользователь не может в рамках одной сессии использовать права «Кассира» и «Контролера», хотя он может быть одновременно членом обеих этих групп). Это немного сложнее. Рассмотрим пример: пользователь является одновременно членом групп «Кассир» и «Контролер», однако, если он входит в систему в качестве кассира, ему недоступны права контролера и наоборот.

Управление доступом в модели RBAC может происходить следующими способами:

- **Не-RBAC.** Назначение прав пользователям производится напрямую в приложениях, роли не используются.
- **Ограниченное RBAC.** Пользователям назначены несколько ролей, а также отдельные права в приложениях, которые не имеют функциональности ролевого управления доступом.
- **Гибридное RBAC.** Пользователям назначены роли, связанные с различными приложениями. Этим ролям назначены только выбранные права.

- **Полное RBAC.** Пользователям назначены корпоративные роли.

Ссылки по теме:

- Role-Based Access Control
- Role-Based Access Control Case Studies
- Role-Based Access Control on the Web
- “A Role-Based Access Control Model and Reference Implementation Within a Corporate Intranet,” by John Barkley, D. Richard Kuhn, and David Ferraiolo (first published in ACM Transactions on Information and System Security, Vol. 2, No. 1, Feb. 1999, pp. 34–64)

RBAC, MAC, DAC. Много путаницы возникает в отношении того, является ли RBAC разновидностью модели DAC или MAC. Различные источники содержат разную информацию на этот счет, но фактически RBAC является самостоятельной моделью. В 1960-х – 1970-х годах американские военные и NSA проводили много исследований модели MAC. Появившаяся в то же время модель DAC имеет свои корни в академических и коммерческих лабораториях. Модель RBAC, которая начала набирать популярность в 1990-е годы, может использоваться в комбинации с системами MAC и DAC. Получить наиболее актуальную информацию о модели RBAC можно по адресу: <http://csrc.nist.gov/rbac>, где размещены документы с описанием стандарта RBAC и независимой модели, с целью прояснить прояснения этой постоянной путаницы.

Модели управления доступом. Важно понимать основные характеристики трех моделей управления доступом:

- **DAC** – владельцы данных решают, кто имеет доступ к ресурсам. Политика безопасности реализуется с помощью ACL.
- **MAC** – политика безопасности реализуется операционной системой посредством меток безопасности.
- **RBAC** – решения о предоставлении доступа принимаются системой на основании ролей и/или должностей субъектов.

ПРИМЕЧАНИЕ. В наше время, конфиденциальность многих типов данных нуждается в защите, поэтому многие компании имеют в штате офицеров безопасности конфиденциальных данных и разрабатывают политики конфиденциальности. Современные модели управления доступом (MAC, DAC, RBAC) сами по себе не обеспечивают защиту критичных данных, вместо этого они ограничивают функции, которые пользователи могут выполнять с этими данными. Например, всем менеджерам может быть предоставлен доступ к папке, содержащей конфиденциальную информацию, но более детальные права доступа могут указывать, что они могут читать информацию о домашнем адресе клиента, но не видят его паспортные данные. Это называется ролевым управлением доступом с учетом конфиденциальности (Privacy Aware Role Based Access Control).

5. Техники и технологии управления доступом

После того как компания решит, какой тип модели управления доступом она собирается использовать, она должна определить и усовершенствовать свои техники и технологии для поддержки этой модели. В следующих разделах описываются различные существующие технологии, поддерживающие различные модели управления доступом.

5.1. Управление доступом на основе правил

Управление доступом на основе правил (Rule-based Access Control) использует определенные правила, указывающие на то, что субъект может и что не может делать с объектом. Это основано на простых правилах (типа, «если X – то Y»), которые могут использоваться для обеспечения более детального управления доступом к ресурсам. Чтобы субъект получил доступ к объекту, должен быть выполнен набор предустановленных правил. Эти правила могут быть простыми и прямолинейными (например, «если идентификатор пользователя соответствует аналогичному идентификатору в предъявленном им цифровом сертификате, пользователю разрешается доступ»), либо может использоваться целый ряд сложных правил, которые должны быть выполнены, чтобы пользователь получил доступ к

объекту (например, «если пользователь входит в систему с понедельника по пятницу и между 9:00 и 18:00, если допуск безопасности пользователя равен или превышает класс объекта, если пользователь имеет необходимую категорию «должен знать», тогда пользователю разрешается доступ»). Управление доступом на основе правил не обязательно основано на идентификации. Например, правило, что вложение в электронном сообщении не должно превышать 5MB, действует на всех пользователей, независимо от их идентификатора. Однако это же правило можно связать с конкретными пользователями, индивидуально установив для них максимальный объем вложения, но это гораздо менее удобно и более трудоемко. Использование правил, затрагивающих всех пользователей, позволяет упростить управление доступом.

Управление доступом на основе правил позволяет разработчикам подробно описать конкретные ситуации, в которых субъект может (или не может) получить доступ к объекту, а также определить, что субъект сможет делать с объектом, получив к нему доступ. Традиционно управление доступом на основе правил используется в системах, использующих модель MAC, в качестве механизма ее реализации. Однако в настоящее время он используется и в других системах и приложениях. Примером могут быть системы контентной фильтрации, кроме того управление доступом на основе правил часто используется в межсетевых экранах и маршрутизаторах.

Ссылки по теме:

- “Creating a Policy-Aware Web: Discretionary, Rule-based Access for the World Wide Web,” Tim Berners-Lee, et al.
- Rule-Based Access Control for Web Services

5.2. Ограниченный пользовательский интерфейс

Ограниченный пользовательский интерфейс (constrained user interface) ограничивает возможности доступа пользователей к отдельным функциям, информации или отдельным системным ресурсам. Существует три основных типа ограниченных интерфейсов: меню и оболочки (shell), представления (view) баз данных и физически ограниченные интерфейсы.

При использовании ограниченного меню и оболочки пользователи видят только те команды, которые они могут использовать. Например, если администратор хочет, чтобы пользователи могли запускать только одну программу, именно одна эта программа должна отображаться в меню выбора. Это ограничивает доступную пользователям функциональность. Оболочка – это разновидность виртуальной среды системы, ее пользовательский интерфейс, командный интерпретатор. Ограниченная оболочка содержит только те команды, которые администратор хочет сделать доступными пользователям.

Часто администраторы баз данных настраивают базы данных так, чтобы пользователи не видели непосредственно поля, содержащие конфиденциальную информацию. Доступ пользователей к данным, содержащимся в базах данных, ограничивается с помощью представлений. Например, если администратор базы данных хочет, чтобы руководители видели график рабочего времени своих сотрудников, но не информацию об их заработной плате, они просто запрещают доступ к полям с информацией о заработной плате для данного типа пользователей. Аналогично, когда сотрудники бухгалтерии, занимающиеся начислением заработной платы, будут просматривать ту же базу данных, им, наоборот, должна быть доступна информация о заработной плате, но не данные о графике рабочего времени. Этот пример проиллюстрирован на рисунке 2-11.

Иванов И.	50 000 р.	9:00 – 18:00
Петров П.	65 000 р.	9:00 – 18:00
Сидоров С.	20 000 р.	9:00 – 14:00

Кадровая база данных

Иванов И.	9:00 – 18:00
Петров П.	9:00 – 18:00
Сидоров С.	9:00 – 14:00

Представление для руководителей

Иванов И.	50 000 р.
Петров П.	65 000 р.
Сидоров С.	20 000 р.

Представление для бухгалтерии

Рисунок 2-11. Различия между представлениями базы данных и самой базой данных

Физически ограниченный интерфейс может быть реализован посредством предоставления пользователю специальной клавиатуры, имеющей ограниченный набор клавиш, или ограниченного набора экранных кнопок на сенсорном экране (примером может быть интерфейс банкомата или терминала для приема платежей).

5.3. Матрица контроля доступа

Матрица контроля доступа (access control matrix) – это таблица субъектов и объектов, содержащая информацию о том, какие действия конкретные субъекты могут делать с конкретными объектами. В таблице 2-1 показан пример матрицы контроля доступа.

Пользователь	Файл 1	Файл 2	Файл 3
Диана	Чтение и Выполнение	Чтение, Запись и Выполнение	Нет доступа
Кэти	Чтение и Выполнение	Чтение	Нет доступа
Крис	Чтение, Запись и Выполнение	Чтение и Выполнение	Чтение
Джон	Чтение и Выполнение	Нет доступа	Чтение и Запись

Таблица 2-1. Пример матрицы контроля доступа

Этот тип управления доступом обычно используется в качестве атрибутов в моделях DAC. Права доступа могут быть напрямую назначены субъектам (разрешения) или объектам (ACL).

Таблицы разрешений

Таблицы разрешений (capability tables) указывают права доступа определенного субъекта к определенным объектам. Таблицы разрешений отличаются от ACL: таблицы разрешений являются ограничением для субъектов, а ACL – для объектов. Разрешения соответствуют строке субъекта в матрице контроля доступа. В таблице 2-1 разрешениями Дианы являются: Файл 1 (Чтение и Выполнение), Файл 2 (Чтение, Запись и Выполнение) и Файл 3 (Нет доступа). Пример системы, основанной на таблицах разрешений – Kerberos. В этой среде пользователь получает билет со своей таблицей разрешений. Билет является границами доступа пользователя и указывает, к каким объектам пользователь может иметь доступ и насколько времени. Управление доступом основано на этом билете (или таблице разрешений). На рисунке 2-12 показаны различия между таблицей разрешений и ACL. Разрешения могут реализовываться в форме токена, билета или ключа. Когда субъект предоставляет свой компонент разрешений, операционная система (или приложение) просматривает права доступа и операции, описанные в нем, и разрешает субъекту выполнять только соответствующие функции. Компонент разрешений – это структура данных, содержащая уникальный идентификатор объекта и права доступа субъекта к объекту. Объектом может быть файл, массив, сегмент памяти или порт. Каждый пользователь, процесс, приложение имеет свой список разрешений.

Матрица контроля доступа				
Субъект	Файл 1	Файл 2	Файл 3	Файл 4
Ларри	Чтение	Чтение, Запись	Чтение	Чтение, Запись
Дэвид	Полный доступ	Нет доступа	Полный доступ	Чтение
Мо	Чтение, Запись	Нет доступа	Чтение	Полный доступ
Боб	Полный доступ	Полный доступ	Нет доступа	Нет доступа
Разрешения			ACL	

Разрешения = строка в матрице
ACL = столбец в матрице

Рисунок 2-12. Таблица разрешений ограничивает субъект, а ACL ограничивает объект

Списки контроля доступа

Списки контроля доступа (ACL – access control list) используются во многих операционных системах, приложениях и маршрутизаторах. Это списки субъектов, которым разрешен доступ к определенному объекту, с указанием уровня разрешенного доступа.

Разграничение доступа может выполняться на уровне пользователей или на уровне групп. ACL являются отображением значений матрицы контроля доступа на отдельный объект. Тогда как разрешения являются строкой в матрице контроля доступа, ACL соответствует столбцу в этой матрице.

5.4. Контентно-зависимое управление доступом

В системе *контентно-зависимого управления доступом* (content-dependent access control) доступ к объекту определяется на основании содержимого самого объекта. Например, содержимое поля базы данных может указывать, какие пользователи могут иметь к нему доступ. Контентно-зависимая фильтрация используется в корпоративных фильтрах электронной почты, которые ищут в тексте сообщения определенные строки (например, «конфиденциально», «номер паспорта», «совершенно секретно» и другие словосочетания, которые компания считает подозрительными). Также компании используют это для контроля доступа в Интернет, аналогичным образом отслеживая в трафике определенные слова, например, с целью выявления сотрудников, играющих в азартные игры или занимающихся поиском работы.

5.5. Контекстно-зависимое управление доступом

Контекстно-зависимое управление доступом (context-dependent access control) отличается от контентно-зависимого, при контекстно-зависимом управлении доступом решение о возможности доступа принимается не на основе критичности данных, а на основе контекста собранной информации. Система, использующая контекстно-зависимое управление доступом, сначала «анализирует ситуацию», а затем принимает решение о возможности доступа. Например, ряд межсетевых экранов может принимать контекстно-зависимое решение, собрав информацию о состоянии пакета, перед тем, как пропустить его в сеть. Межсетевой экран с контролем состояния (stateful firewall) «знает» необходимые шаги для коммуникаций по определенным протоколам и проверяет, что они были соблюдены. Например, при использовании соединения TCP, отправитель отправляет пакет SYN, получатель отправляет SYN/ACK, и затем отправитель направляет пакет ACK (подтверждение). Межсетевой экран с контролем состояния понимает эти шаги и не пропускает пакеты, нарушающие эту последовательность. Если, к примеру, такой межсетевой экран получает SYN/ACK, однако перед ним не было соответствующего (в рамках этого соединения) пакета SYN, межсетевой экран понимает, что это неправильно и уничтожает этот пакет. Это пример контекстно-зависимого управления доступом – в нем межсетевой экран учитывает контекст при принятии решения о возможности доступа.

Техники управления доступом. Для поддержки модели управления доступом используются техники управления доступом.

- **Матрица контроля доступа.** Таблица субъектов и объектов, которая описывает возможности их взаимодействия.
- **Список контроля доступа (ACL).** Ограничивает права доступа к объекту, указывая, какие субъекты могут получить доступ к нему.
- **Таблица разрешений.** Ограничивает права доступа субъекта, указывая, к каким объектам он может получить доступ.
- **Доступ на основе контента.** Решения о возможности доступа принимаются на основе критичности данных, а не только на основе идентификатора субъекта.
- **Доступ на основе контекста.** Решения о возможности доступа принимаются в зависимости от ситуации, а не только на основе идентификатора субъекта или критичности данных.
- **Ограниченный интерфейс.** Ограничения пользовательской среды в системе, посредством чего ограничивается доступ к объектам.
- **Доступ на основе правил.** Ограничивает доступ субъектов по заранее определенным

правилам.

6. Администрирование доступа

Итак, после того как компания разработала политику безопасности, а также поддерживающие ее процедуры, стандарты и руководства (см. Домен 01), она должна выбрать тип модели управления доступом: DAC, MAC или ролевое. Затем компания должна выбрать и внедрить различные технологии и техники управления доступом, среди которых могут быть матрицы контроля доступа, ограниченные интерфейсы, контекстно- или контентно-зависимое управление, либо управление на основе правил. Если среда не требует высокого уровня безопасности, компании следует выбрать дискреционную или ролевую модель. Модель DAC позволяет владельцам данных управлять доступом пользователей к своим ресурсам, поэтому компания должна полностью осознавать возможные последствия, выбирая эту модель. Если в компании большая «текучесть» кадров и/или требуется более централизованный метод управления доступом, целесообразно выбрать ролевую модель. Если среда требует более высокого уровня безопасности, и в ней только администратор должен иметь возможность предоставления доступа к ресурсам, тогда наилучшим выбором будет модель MAC. Теперь осталось определить, как компания будет администрировать выбранную модель управления доступом. Существует два основных варианта администрирования управления доступом: централизованный и децентрализованный. При принятии решения необходимо понимать оба подхода, чтобы выбрать из них именно тот, который позволит обеспечить необходимый уровень безопасности.

6.1. Централизованное администрирование управления доступом

При централизованном администрировании доступа (centralized access control administration) один субъект (человек или подразделение) следит за доступом ко всем корпоративным ресурсам. Этот субъект (администратор безопасности) настраивает механизмы, которые реализуют управление доступом, выполняет изменения пользовательских профилей, отзывает права доступа при необходимости, полностью блокирует доступ пользователя в случае его увольнения. Этот тип администрирования предоставляет последовательные и унифицированные методы управления правами доступа пользователей. Он обеспечивает строгий контроль данных, т.к. только один человек (или подразделение) имеет необходимые права для изменения профилей доступа и разрешений, однако это довольно медленный способ, поскольку все изменения должны быть выполнены одним человеком (подразделением).

В следующих разделах будут приведены несколько примеров технологий централизованного удаленного управления доступом. Применяющиеся для этих целей протоколы аутентификации называют AAA-протоколами (аутентификация, авторизация и аудит).

В зависимости от протокола, существуют различные способы аутентификации пользователей в клиент-серверной архитектуре. Традиционные протоколы аутентификации: PAP (password authentication protocol), CHAP (challenge handshake authentication protocol) и новый метод EAP (extensible authentication protocol). Каждый из этих протоколов будет рассмотрен в Домене 05.

RADIUS

RADIUS (remote authentication dial-in user service) – это сетевой протокол, который обеспечивает клиент-серверную аутентификацию, авторизацию и аудит удаленных пользователей. Для подключения удаленных пользователей, в сети должны быть установлены серверы доступа и средства для удаленного подключения (например, модемный пул, DSL, ISDN или линия T1). Сервер доступа запрашивает у пользователя учетные данные для входа и передает их на сервер RADIUS, на котором хранятся имена пользователей и пароли. При этом удаленный пользователь является клиентом сервера доступа, а сервер доступа является клиентом RADIUS-сервера.

В настоящее время большинство интернет-провайдеров используют RADIUS для аутентификации клиентов, перед тем, как разрешить им доступ к сети Интернет. Сервер доступа и программное обеспечение клиента «договариваются» о протоколе аутентификации, который они будут использовать (PAP, CHAP или EAP), посредством процедуры «рукопожатия» (handshake), после чего клиент передает серверу доступа свое имя пользователя и пароль. Это взаимодействие происходит через соединение PPP. Взаимодействие сервера доступа с RADIUS-сервером осуществляется по протоколу RADIUS. Сервер доступа уведомляет RADIUS-сервер о фактах начала и окончания сеанса в целях предоставленных тарификации услуг (биллинга).

RADIUS также используется в корпоративных средах, чтобы обеспечить доступ к корпоративной сети сотрудникам во время командировок или из дома. RADIUS позволяет компаниям хранить профили пользователей в централизованной базе данных. После успешной аутентификации, внешнему пользователю присваивается предварительно настроенный профиль, определяющий к каким ресурсам он может получить доступ, а к каким – нет. Эта технология позволяет компании иметь единую управляемую точку входа, что обеспечивает стандартизацию с точки зрения безопасности и предоставляет простой способ контроля использования удаленного доступа и сбора сетевой статистики.

RADIUS был разработан компанией Livingston Enterprises для своей серии серверов доступа, но затем был опубликован в виде стандартов RFC 2865 и RFC 2866. Это открытый протокол, который может использовать любой производитель в своих продуктах. Конфигурации и учетные данные пользователей могут храниться на серверах LDAP, в различных базах данных или текстовых файлах. Рисунок 2-13 показывает некоторые примеры возможных реализаций RADIUS.

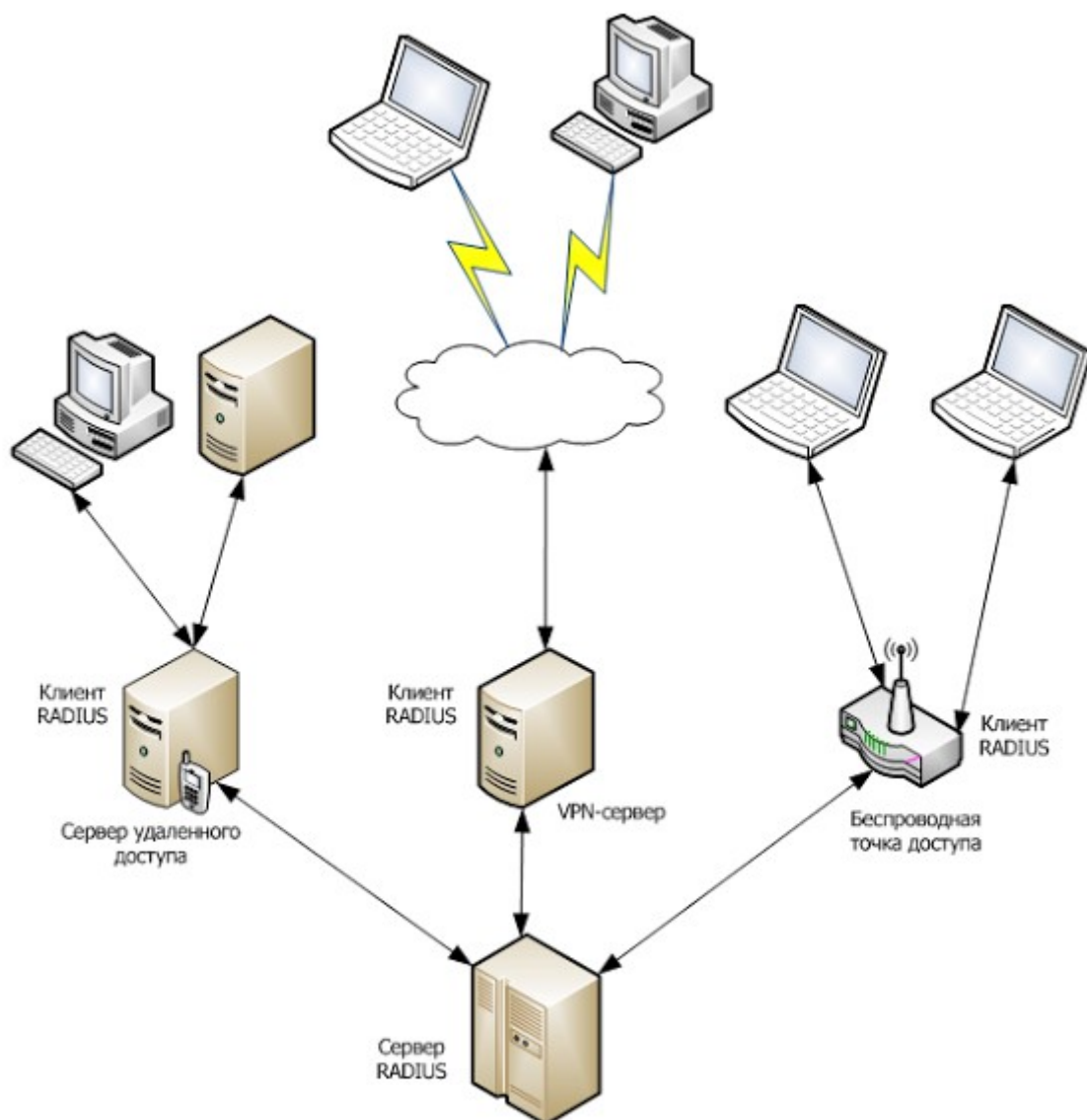


Рисунок 2-13. В различных средах инфраструктура RADIUS может быть реализована по-разному

TACACS

TACACS (Terminal Access Controller Access Control System) прошел через три поколения: TACACS, Extended TACACS (XTACACS) и TACACS+. TACACS объединяет процессы аутентификации и авторизации, XTACACS разделяет процессы аутентификации, авторизации и аудита, а TACACS+ – это XTACACS с расширенной двухфакторной аутентификацией пользователей. TACACS использует постоянные пароли для аутентификации, а TACACS+ позволяет использовать динамические (одноразовые) пароли, обеспечивая более высокий уровень безопасности.

ПРИМЕЧАНИЕ. TACACS+ на самом деле не является новым поколением TACACS и XTACACS. Это новый протокол, который обеспечивает похожую функциональность и использует ту же схему имен. И поскольку это совершенно другой протокол, он не совместим с TACACS или XTACACS.

TACACS+ реализует в основном ту же функциональность, что и RADIUS с некоторыми отличиями. Во-первых, TACACS+ использует в качестве транспорта протокол TCP вместо протокола UDP в RADIUS, и поэтому ему не требуется дополнительный код для обнаружения и исправления ошибок передачи сетевых пакетов. Во-вторых, TACACS+ шифрует всю информацию (включая имя пользователя, данные учета и авторизации), а RADIUS шифрует только пароль пользователя, позволяя злоумышленнику перехватить важную информацию. В-третьих, TACACS+ использует настоящую архитектуру AAA,

обеспечивая дополнительную гибкость при настройке процесса аутентификации удаленных пользователей, тогда как RADIUS объединяет функциональность аутентификации и авторизации.

Например, перед сетевым администратором Томом была поставлена задача организации удаленного доступа для пользователей, и он должен выбрать между RADIUS и TACACS+. Если в имеющейся среде аутентификация всех локальных пользователей осуществляется через контроллер домена с помощью Kerberos, Том может аналогичным образом настроить процесс аутентификации удаленных пользователей, как показано на Рисунке 2-14. Вместо того чтобы одновременно поддерживать базу данных удаленных пользователей на сервере удаленного доступа и базу данных локальных пользователей в Active Directory, Том может настроить работу через одну базу данных. Разделение функциональности аутентификации, авторизации и учета позволяет сделать это. TACACS+ также позволяет сетевому администратору настроить более детальные профили пользователей, указывая реальные команды, которые пользователи могут выполнять.

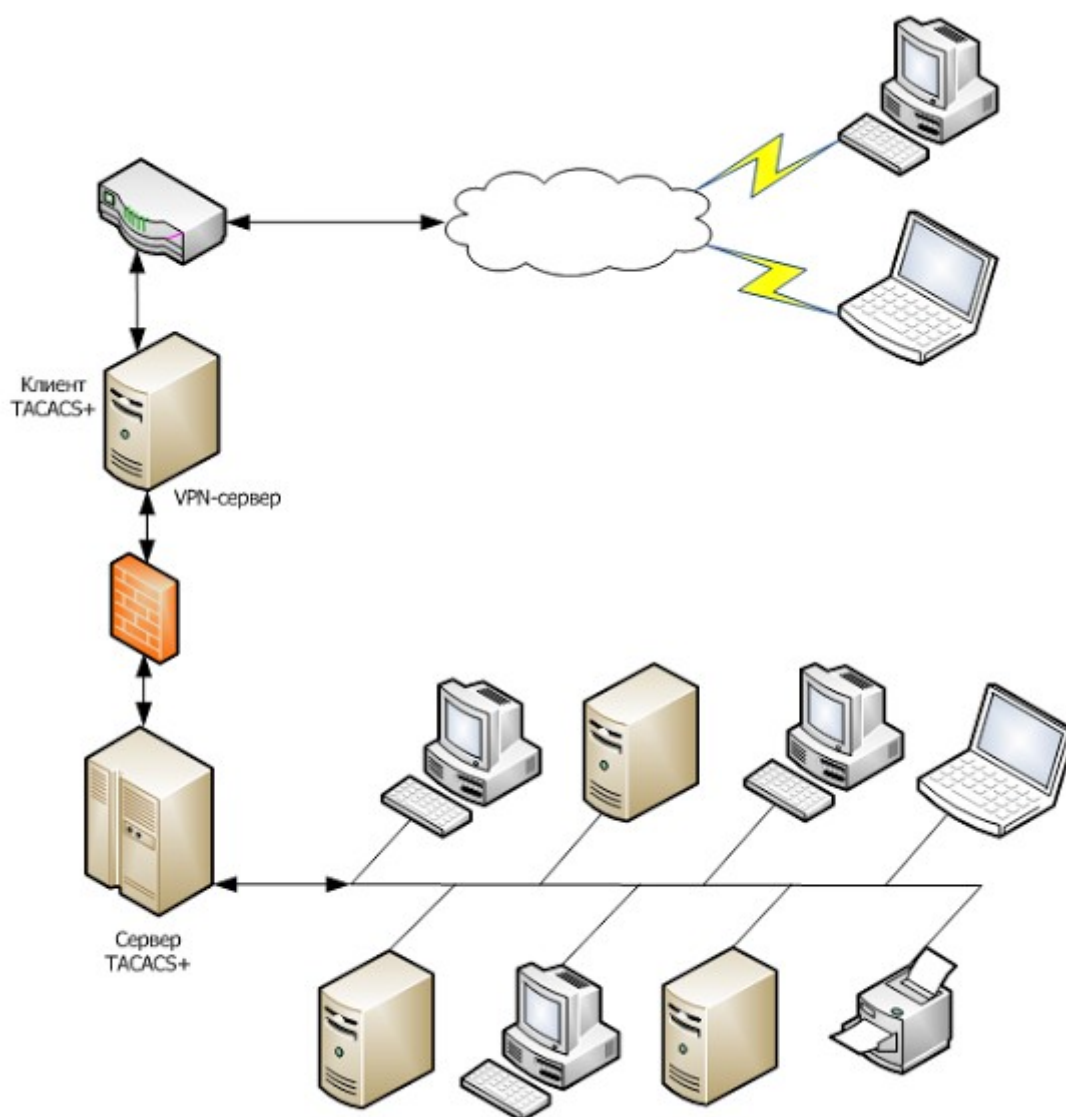


Рисунок 2-14. TACACS+ работает в рамках клиент-серверной модели

Помните, что RADIUS и TACACS+ являются протоколами, а протоколы являются не более чем согласованным способом взаимодействия. Когда RADIUS-клиент связывается с RADIUS-сервером, он делает это посредством протокола RADIUS, который на самом деле представляет собой некий набор полей, в которые записываются некоторые значения. Эти

поля называют парами атрибут-значение (AVP – attribute-value pair). В качестве аналогии, предположим, что я посылаю вам пустой бланк, на котором есть несколько пустых строк, каждая из которых имеет название: фамилия, имя, цвет волос, размер обуви. Вы заполняете эти строки соответствующими значениями и отправляете заполненный бланк обратно мне. В общем виде, именно так работают протоколы: отправляющая система просто заполняет поля необходимой для принимающей системы информацией, которая затем извлекает ее и обрабатывает.

TACACS+ имеет больше AVP, что обеспечивает более детальный контроль того, что могут и что не могут делать пользователи, а также позволяет сетевому администратору определять списки ACL, фильтры, привилегии пользователей и многое другое. Таблица 2-2 показывает различия между RADIUS и TACACS+.

	RADIUS	TACACS+
Передача пакетов	UDP	TCP
Шифрование пакетов	Шифруется только пароль при передаче от RADIUS-клиента к серверу.	Шифруется весь трафик между клиентом и сервером.
Поддержка AAA	Объединяет сервисы аутентификации и авторизации.	Использует архитектуру AAA, разделяя аутентификацию, авторизацию и аудит.
Поддержка протоколов	Работает через соединение PPP.	Поддерживает другие протоколы, такие как AppleTalk, NetBIOS и IPX.
Ответы	Использует единственный запрос-ответ при аутентификации пользователя, который используется для всех действий AAA.	Использует множество запросов-ответов для каждого процесса AAA. Каждое действие AAA должно быть аутентифицировано.

Таблица 2-2. Отдельные различия между двумя AAA-протоколами

Таким образом, RADIUS является подходящим протоколом, если используется упрощенная аутентификация по имени пользователя и паролю, и системе нужно только решить, разрешать доступ пользователю или нет (как, например, для работы интернет-провайдера). TACACS+ лучше подходит для сред, в которых требуется более надежная аутентификация и жесткий контроль более сложных действий по авторизации (как, например, в корпоративных сетях).

Сторожевые таймеры (Watchdog). Сторожевые таймеры, как правило, используются для выявления программных ошибок, таких как некорректно завершившийся или «зависший» процесс. Сторожевая функция посылает периодические запросы, чтобы определить, отвечает ли на них сервис. Если она не получает ответа от сервиса, он может быть остановлен или сброшен. Сторожевая схема предотвращает взаимные блокировки программного обеспечения, бесконечных циклов и проблем с приоритетами процессов. Такая функциональность может использоваться в протоколах AAA для выявления необходимости повторной отправки пакета, либо закрытия и повторного открытия соединения, с которым возникли какие-либо проблемы.

Diameter

Diameter – это протокол, разработанный на базе функциональности RADIUS, устраняющий многие из его ограничений. Diameter – это еще один AAA-протокол, предоставляющий такую же функциональность, как RADIUS и TACACS+, но являющийся более гибким и отвечающий современным требованиям. Одно время все удаленные подключения осуществлялись по протоколам PPP и SLIP, а аутентификация пользователей производилась через PAP или CHAP. Но сегодня технологии стали значительно сложнее, появилось множество различных устройств и протоколов, между которыми можно выбирать. Сегодня мы хотим, чтобы наши беспроводные устройства и смартфоны могли аутентифицироваться в нашей сети, мы используем протоколы роуминга, мобильные IP, PPP через Ethernet, голос через IP (VoIP) и т.п. Традиционные AAA-протоколы не могут работать со всем этим. Поэтому был разработан новый AAA-протокол Diameter, который решает эти проблемы, а также многие другие.

Протокол Diameter состоит из двух частей. Первая часть – это базовый протокол, который

обеспечивает безопасное взаимодействие между участниками Diameter, определение характеристик и соглашение о версии. Вторая является расширением, которое настроено над базовым протоколом. Эта часть позволяет различным технологиям использовать Diameter для аутентификации.

Мобильный IP. Эта технология позволяет пользователю перемещаться из одной сети в другую и при этом продолжать использовать один и тот же IP-адрес. Это усовершенствование протокола IP, которое позволяет пользователю иметь свой домашний IP-адрес, связанный с его домашней сетью, и обслуживающий адрес (care-of address). Обслуживающий адрес меняется при перемещении из одной сети в другую. Весь трафик, адресованный на домашний IP-адрес пользователя, перенаправляется на его обслуживающий адрес.

До появления концепции Diameter, в IETF были отдельные рабочие группы, которые определяли порядок работы протоколов VoIP (голос через IP), FoIP (факс через IP), Мобильный IP, а также протоколов удаленной аутентификации. Внедрение их по отдельности в любой сети легко может привести к целому ряду сложностей, включая проблемы совместимости. Это требует от клиентов разворачивать и настраивать несколько серверов с различными политиками и увеличивает стоимость каждого нового дополнительного сервиса. Diameter предоставляет базовый протокол, который определяет формат заголовков, параметры безопасности, команды и AVP. Этот базовый протокол позволяет связать расширения с другими сервисами, такими как VoIP, FoIP, Мобильный IP, аутентификацию беспроводных устройств и мобильных телефонов. Таким образом, Diameter может использоваться в качестве AAA-протокола во всех этих случаях.

В качестве аналогии, рассмотрим ситуацию, в которой десяти сотрудникам одной компании нужно попасть на работу. Они могут ездить на работу на своих автомобилях по своим маршрутам, но это потребует дополнительную территорию для организации стоянки, а также найма охранника, чтобы он стоял около ворот и пропускал только машины сотрудников компании. С другой стороны, все они могут воспользоваться автобусом компании. Автобус в данном случае является общим элементом (как базовый протокол), позволяющим всем сотрудникам (различным сервисам) попасть в одно и то же место – на работу (сетевую среду). Diameter обеспечивает общий AAA и структуру безопасности, в рамках которой могут работать различные службы, как показано на Рисунке 2-15.



Рисунок 2-15. Diameter предоставляет архитектуру AAA для различных сервисов

ПРИМЕЧАНИЕ. ROAMOPS (Roaming Operations) позволяет пользователям PPP получить доступ к сети Интернет без необходимости звонить своему домашнему интернет-провайдеру. Интернет-провайдер, который имеет роуминговое соглашение, выполняет перекрестную аутентификацию клиентов, что позволяет пользователю позвонить любому интернет-провайдеру по месту своего присутствия и получить доступ в Интернет.

RADIUS и TACACS+ являются клиент-серверными протоколами, т.е. серверная часть не может по своей инициативе отправлять команды клиентской части. Diameter – это одноранговый (peer-based) протокол, что позволяет любой стороне инициировать взаимодействие. Функциональность позволяет серверу Diameter отправлять сообщения серверу доступа, чтобы запросить у пользователя другой набор учетных данных, если он пытается получить доступ к защищаемому ресурсу.

Diameter не является полностью обратно совместимым с RADIUS, но он предоставляет возможности для перехода с RADIUS. Diameter использует UDP и AVP, и обеспечивает поддержку работы через прокси-сервер. По сравнению с RADIUS, он лучше выявляет и

исправляет ошибки, обладает повышенной отказоустойчивостью. Diameter также предоставляет сквозную (end-to-end) безопасность посредством использования IPSec или TLS, которые недоступны в RADIUS.

Diameter может предоставлять функциональность AAA другим протоколам и сервисам, поскольку имеет широкий набор AVP. RADIUS имеет 2^8 AVP, тогда как Diameter – 2^{32} . Как было сказано ранее в этом домене, AVP похожи на пустые строки на бланке, которые описывают как стороны могут взаимодействовать друг с другом. Поэтому, большее количество AVP позволяет обеспечить больше функциональности и сервисов для взаимодействия систем. Diameter предоставляет следующую функциональность AAA:

- **Аутентификация**
 - PAP, CHAP, EAP
 - Защита аутентификационной информации между конечными точками
 - Защита от атак повтора (replay attack)
- **Авторизация**
 - Перенаправление, безопасные прокси, ретрансляция (relay), посредничество (broker)
 - Согласование состояния
 - Отключение по собственной инициативе
 - Переавторизация по запросу
- **Учет**
 - Отчетность, учет ROAMOPS (roaming operations), мониторинг событий.

Diameter – это относительно новый протокол. Вероятно, в ближайшее время он не сможет захватить весь мир, сначала он будет использоваться в средах, которым изначально были нужны его сервисы, а затем постепенно проникать в корпоративные сети, будучи реализованным все в большем и большем количестве доступных продуктов. RADIUS длительное время использовался практически повсеместно, он хорошо служит своим целям и сейчас, поэтому не следует ожидать его скорого выведения из эксплуатации.

Ссылки по теме:

- RFC 3588 – Diameter Base Protocol
- RFC 2869 – RADIUS Extensions
- RFC 2865 – Remote Authentication Dial In User Service (RADIUS)
- RFC 2975 – Introduction to Accounting Management

6.2. Децентрализованное администрирование управления доступом

Метод *децентрализованного администрирования управления доступом* (decentralized access control administration) передает управление доступом людям, которые непосредственно связаны с ресурсами и лучше понимают, кто должен и кто не должен иметь доступ к определенным файлам, данным и ресурсам. Обычно это функциональные руководители, которые предоставляют права доступа сотрудникам. Компании следует выбрать децентрализованную модель, если ее руководители имеют хорошее представление о том, какие пользователи к каким ресурсам должны иметь доступ, а также в компании отсутствуют требования о необходимости использования централизованной модели.

Управление доступом в децентрализованной модели может происходить быстрее, поскольку

этим занимается больше людей в компании. Однако при этом может возникнуть конфликт интересов. Поскольку не один человек (подразделение) управляет всеми правами доступа, различные руководители и подразделения могут выполнять функции по управлению доступом и обеспечению безопасности различными способами. Это не позволит обеспечить унификацию и справедливость в рамках всей компании. Одни руководители будут слишком заняты своими ежедневными задачами и легко позволят кому угодно получить полный доступ ко всем системам своего подразделения. Другие подразделения, напротив, будут применять строгие и детальные методы управления, предоставляя сотрудникам только тот уровень доступа, который необходим им для выполнения своих задач. Кроме того, в некоторых случаях функции управления доступом будут накладываться друг на друга, что может стать причиной того, что некоторые нежелательные действия не будут запрещены и заблокированы. Если Майк, входящий в группу бухгалтерии, был заподозрен в несанкционированном изменении информации о заработной плате персонала, руководитель бухгалтерии может ограничить его доступ к соответствующим файлам правами «только чтение». Однако руководитель бухгалтерии не делает этого, и Майк продолжает иметь к этим файлам полный доступ в рамках сетевой группы, членом которой он является. Таким образом, метод децентрализованного администрирования не обеспечивает целостного управления и достаточного уровня согласованности в процессе защиты компании. Например, если Шон уволен с работы за просмотр порнографии на своем рабочем компьютере, руководитель соответствующей группы, в которую входит Шон, может не заблокировать его учетную запись, и у Шона после увольнения останутся права доступа. Это может привести к серьезным проблемам у компании, если Шон захочет отомстить.

7. Методы управления доступом

Управление доступом может обеспечиваться на различных уровнях сети и отдельных систем. Одни средства управления доступом являются компонентами ядра операционной системы или встроены в приложения и устройства. Другие могут поставляться в виде отдельных пакетов, разработанных сторонними производителями. Различные средства управления доступом предоставляют различную функциональность, поэтому они должны работать совместно, чтобы обеспечивать необходимый уровень защиты, при котором плохие парни будут оставаться снаружи, а хорошие – внутри.

Большинство компаний не хочет, чтобы посторонние бесконтрольно разгуливали по их офисам, садились за компьютеры сотрудников, получали доступ к сетевым ресурсам. Также компании не хотят, чтобы каждый сотрудник имел доступ ко всей информации компании, например, записям кадрового учета, сведениям о заработной плате, коммерческой тайне. Компании хотят иметь определенную уверенность в том, что для сотрудников, которые имеют доступ к критичной информации, установлены ограничения, не позволяющие им удалить финансовые, налоговые и иные данные, повреждение которых является риском для компании. Некоторые виды управления доступом, предотвращающие такие события, рассматриваются далее.

7.1. Уровни управления доступом

Управление доступом состоит из трех широких категорий: административного, технического и физического. Каждая категория имеет различные механизмы управления доступом, которые могут выполняться автоматически или вручную. Все эти механизмы должны работать согласованно друг с другом для защиты инфраструктуры и данных.

Каждая категория управления доступом содержит несколько компонентов:

- **Административный уровень**
 - Политики и процедуры
 - Управление персоналом

- Структура надзора
- Тренинги и повышение осведомленности по вопросам безопасности
- Тестирование
- **Физический уровень**
 - Сегментация сети
 - Защита периметра
 - Управление компьютерами
 - Отделение рабочих областей
 - Резервное копирование данных
 - Прокладка кабелей
 - Контролируемые зоны
- **Технический уровень**
 - Доступ к системам
 - Сетевая архитектура
 - Доступ к сети
 - Шифрование и протоколы
 - Аудит

Следующие разделы разъясняют каждую из этих категорий и компонентов, показывают, как они связаны с управлением доступом.

7.2. Административный уровень

Высшее руководство должно принять решение в отношении роли безопасности в компании, а также ее целей и задач. От этого решения зависит реализация всех механизмов, обеспечивающих и поддерживающих безопасность. Обычно руководство предоставляет «скелет» инфраструктуры безопасности и определяет, кто будет «наполнять» его.

Первая часть строительства фундамента безопасности в компании – это политика безопасности. Создание политики безопасности входит в обязанности руководства, после чего оно должно делегировать разработку поддерживающих ее процедур, стандартов и руководств, указывающих какое управление персоналом должно применяться, как должно проходить тестирование, которое гарантирует, что все части работают для достижения целей безопасности компании. Эти элементы являются административным уровнем управления, они работают на самом верху иерархии модели управления доступом. (Административный уровень управления рассматривался в Домене 01, но здесь показывается его взаимосвязь с логическими и физическими уровнями управления доступом).

Политики и процедуры

Политика безопасности – это высокоуровневый документ, отражающий намерения руководства в отношении безопасности компании, указывающий, какие действия являются допустимыми, какой уровень риска считается приемлемым. Политика безопасности основывается на требованиях законодательства, регуляторов, а также бизнес-целях самой компании. Эта политика задает направление для каждого сотрудника и подразделения, указывающее как политика должна быть внедрена, как она должна соблюдаться и какие меры предусмотрены за несоответствие ей. Процедуры, руководства и стандарты предоставляют дополнительные детали, необходимые для поддержки и соблюдения

политики безопасности компании.

Управление персоналом

Управление персоналом определяет, как сотрудники должны взаимодействовать с механизмами безопасности и какие предусмотрены меры воздействия за несоблюдение требований безопасности. Сюда включаются меры безопасности при приеме, увольнении, переводе и повышении сотрудников. Следует разработать специальные процедуры для каждой из этих ситуаций, для чего целесообразно привлекать подразделение по работе с кадрами, а также юридическое подразделение. Разделение критичных обязанностей и ротация обязанностей также являются средствами управления персоналом, которые должны быть продиктованы руководством.

Структура надзора

Высшее руководство должно создать структуру надзора, в которой каждый сотрудник отчетывается перед вышестоящим руководителем, а этот руководитель несет ответственность за выполнение данной функции. Это направлено на обеспечение ответственности руководителей за своих подчиненных и за выполняемые ими действия, на повышение интереса со стороны руководителей к деятельности подчиненных им сотрудников. Если сотрудника поймали на попытке взлома сервера, на котором хранится информация о кредитных картах клиентов, сотрудник и его руководитель, перед которым он отчетывается (supervisor), будут совместно нести за это ответственность. Это административная мера помогает бороться с мошенничеством и обеспечивать надлежащий контроль.

Тренинги и повышение осведомленности по вопросам безопасности

Во многих компаниях руководство крайне неохотно тратит время, деньги и другие ресурсы на те вещи, которые не влияют на рентабельность. Именно поэтому проведение тренингов и повышение осведомленности по вопросам безопасности традиционно имеет низкий приоритет, однако последнее время у компаний возникает все больше и больше проблем, связанных с компьютерной безопасностью, и руководство начинает понимать ценность таких тренингов и повышения осведомленности.

Безопасность компании зависит от технологий и людей, а люди обычно являются «слабым звеном» – именно их действия (или бездействие) чаще всего приводят к появлению уязвимостей и нарушению безопасности компании. Если пользователи понимают, как правильно использовать ресурсы, знают, зачем нужны средства управления доступом и к чему может привести их отсутствие, осведомлены о последствиях ненадлежащего использования ресурсов, компания может существенно снизить количество инцидентов безопасности.

Тестирование

Все защитные меры, механизмы и процедуры должны периодически проверяться, чтобы убедиться, что они надлежащим образом поддерживают политику безопасности и выполняют установленные для них задачи. Тестированием может быть отработка реакции на физические атаки или повреждения сети, проведение теста на проникновение через межсетевой экран и сетевой периметр для выявления уязвимостей, проверка знаний сотрудников, пересмотр процедур и стандартов для убеждения в том, что они соответствуют произошедшим изменениям в бизнесе и технологиях.

Поскольку изменения происходят постоянно, а окружение эволюционирует, процедуры и практики безопасности должны непрерывно тестироваться для уверенности в том, что они соответствуют ожиданиям руководства и остаются актуальными при каждом изменении инфраструктуры. Руководство обязано убедиться, что процесс тестирования выполняется надлежащим образом.

7.3. Физический уровень

Мы будем детально рассматривать физическую безопасность далее в Домене 04, но сейчас важно понять, что определенные физические меры безопасности должны поддерживать административные и технические (логические) меры и работать совместно с ними.

Примерами физических мер безопасности являются: охрана, проверяющая пропуска на входе в здание, ограждения внешнего периметра здания, защита серверной комнаты от посторонних и от факторов внешней среды (влажность, температура), обеспечение только уполномоченным лицам доступа в определенные помещения и работы с конфиденциальной информацией.

Сегментация сети

Сегментация сети может быть реализована физическими и логическими средствами.

Например, сеть может быть спроектирована таким образом, что все компьютеры AS400 и базы данных не только отделены в отдельный сетевой сегмент на логическом уровне, но и физически находятся в отдельном помещении, для входа в которое необходимо иметь специальную магнитную карточку, выдаваемую лишь сотрудникам с соответствующим допуском. Другие сегменты сети могут включать в себя веб-серверы, маршрутизаторы и коммутаторы, а третьи – рабочие станции сотрудников. Каждый сегмент дополнительно может защищаться на физическом уровне – в соответствующие помещения могут допускаться только уполномоченные лица.

Защита периметра

Порядок защиты периметра зависит от компании и ее требований к безопасности. В одной компании может быть организован пропускной режим, в то время, как другая компания может позволять любому зайти в ее помещения. Защита периметра может включать в себя систему видеонаблюдения, оборудованную детекторами движения, сенсорами, сигнализацией, контролирующую зоны парковки, места ожидания, ограждения вокруг здания, проходы во внутренней территории. Все это примеры механизмов защиты периметра, которые обеспечивают управление доступом на физическом уровне с целью защиты людей, зданий и того, что находится в этих зданиях.

Управление компьютерами

Каждый компьютер может иметь физические средства контроля, такие как, защита от вскрытия, снятие дисководов и CD-ROM'ов для предотвращения копирования конфиденциальной информации, установка устройств, снижающих побочные электромагнитные излучения и т.д.

Отделение рабочих областей

Доступ в определенные помещения может быть разрешен только уполномоченным сотрудникам. Например, в исследовательских компаниях ограничивают доступ в лаборатории, чтобы никто не нарушил ход экспериментов и не получил тестовые данные. Доступ к серверным комнатам и коммуникационным шкафам предоставляют только соответствующему ИТ-персоналу, чтобы избежать ошибок и попыток саботажа. В финансовых организациях ограничивают доступ в хранилища ценностей и кассовые помещения.

Резервное копирование данных

Резервное копирование данных – это физическая мера безопасности, обеспечивающая доступность данных после аварий в сети или системах. Администратор ведет банк резервных копий на внешних носителях и хранит его в огнестойком сейфе или перевозит в другое здание.

Прокладка кабелей

Для передачи информации по сети могут использоваться различные типы кабелей. Некоторые из них обладают специальной оболочкой, обеспечивающей защиту от перехвата электрических сигналов. В других, каждый провод покрывается защитным материалом («экраном»), обеспечивающим отсутствие наводок от других проводов. Все кабели следует прокладывать только внутри здания, в недоступных для доступа посторонних местах, чтобы исключить повреждение кабелей и перехват информации.

Контролируемые зоны

Здание компании следует разделить на зоны в зависимости от критичности деятельности, выполняющейся в каждой из них. Фойе может рассматриваться как публичное место, помещения, в которых выполняется разработка программного обеспечения, могут считаться совершенно секретными, а офисы руководства – секретными. Не так важна используемая классификация, важно понимать, что одни помещения могут быть более критичны, чем другие, и что помещения (зоны) требуют различных механизмов управления доступом, основанных на требованиях к уровню защиты. Отдельные помещения (зоны) могут потребовать использования специальных материалов и/или оборудования, препятствующих утечке информации по техническим каналам.

То же самое справедливо и для сети компании. Она должна быть сегментирована, и средства управления доступом в каждом сегменте должны выбираться на основе критичности устройств, подключенных к этому сегменту, а также на основе критичности данных, обрабатываемых в нем.

7.4. Технический уровень

Технические (логические) меры безопасности – это программные средства, используемые для ограничения доступа субъектов к объектам. Это могут быть компоненты операционных систем, отдельные пакеты безопасности, приложения, аппаратные сетевые устройства, протоколы, механизмы шифрования, матрицы контроля доступа. Эти меры безопасности работают на разных уровнях в сети или системах, но при этом должна быть обеспечена их совместная работа для защиты от несанкционированного доступа к ресурсам и гарантий доступности, целостности и конфиденциальности ресурсов. Технические меры защищают целостность и доступность ресурсов, ограничивая число субъектов, которые могут иметь к ним доступ, а также конфиденциальность ресурсов, предотвращая их раскрытие неавторизованным субъектам. Следующие разделы поясняют, как работают некоторые технические меры, и в какие части среды они внедряются.

Доступ к системам

Различные типы средств и механизмов безопасности управляют доступом к ресурсам. Если компания использует архитектуру MAC, определяется уровень допуска пользователя и сравнивается с уровнем классификации ресурса для проверки, может ли пользователь получить доступ к запрашиваемому объекту. Если компания использует архитектуру DAC, операционная система проверяет, имеет ли пользователь разрешение на доступ к запрошенному ресурсу. Критичность данных, уровни допуска пользователей, права и разрешения пользователей используются в качестве логических мер безопасности для управления доступом к ресурсам. В качестве технических мер могут использоваться комбинации имени пользователя и пароля, реализации системы Kerberos, биометрические системы, инфраструктуры открытых ключей (PKI), RADIUS, TACACS, системы аутентификации посредством смарт-карт. Эти технологии, используя различные механизмы аутентификации, проверяют, что пользователь является именно тем, за кого себя выдает. После успешной аутентификации пользователь может быть авторизован и ему может быть предоставлен доступ к ресурсам. Эти технологии будут рассмотрены в следующих разделах, но на данный момент нужно понимать, что доступ к системам – это разновидность технических мер безопасности, реализующих управление доступом.

Сетевая архитектура

Архитектура сети может быть построена и реализована посредством нескольких логических мер защиты, обеспечивающих изоляцию и защиту окружения. Сеть может быть изолирована физически (стенами, выделенными помещениями), либо логически (отдельными адресными пространствами, подсетями, сегментами, управляемыми коммуникационными потоками между сегментами). Часто очень важно контролировать взаимодействие различных сегментов между собой. На рисунке 2-16 показан пример того, как компания может сегментировать свои сети и организовать взаимодействие между сегментами. В приведенном примере компания не хочет, чтобы между внутренней сетью и демилитаризованной зоной (DMZ) были открытые и неограниченные коммуникационные маршруты. Обычно внутренним пользователям нет необходимости напрямую обращаться к системам в DMZ, а исключение таких маршрутов взаимодействия снижает возможности для внутренних атак на эти системы. Кроме того, если атака из Интернета успешно компрометирует систему в DMZ, атакующий не должен получить при этом возможность простого доступа во внутреннюю сеть, для чего должен применяться соответствующий тип логической изоляции.

На этом примере также показано, что управляющий сегмент может взаимодействовать со всеми другими сетевыми сегментами, но эти сетевые сегменты не могут взаимодействовать с управляющим сегментом, так как в нем находятся консоли, управляющие межсетевыми экранами и IDS, и нет причин для взаимодействия с ними пользователей и других администраторов.

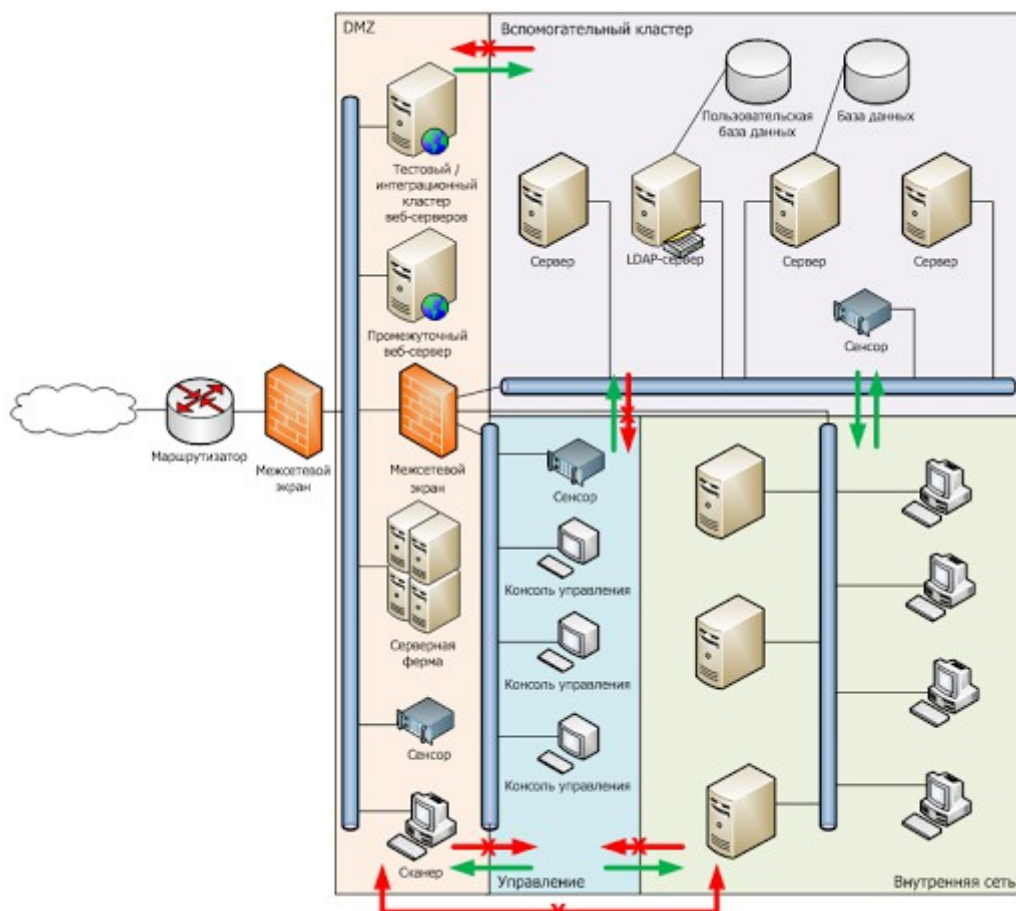


Рисунок 2-16. Сегментация сети на техническом уровне управляет порядком взаимодействия различных сегментов сети

Доступ к сети

Системы используют логические средства защиты, определяющие, кто может получить доступ к ним и что он может делать с ними после аутентификации. Это справедливо и для

сетей. Маршрутизаторы, коммутаторы, межсетевые экраны и мосты работают как технические средства защиты, обеспечивающие ограничения доступа в/из сети, а также между различными сегментами сети. Если атакующий из Интернета хочет получить доступ к определенному компьютеру, он должен взломать сначала межсетевой экран, маршрутизатор и коммутатор, и только потом он сможет приступить к взлому этого компьютера во внутренней сети. Каждое устройство имеет свои собственные логические средства, которые принимают решения, кто может получить к нему доступ и какие действия выполнять.

Доступ к различным сетевым сегментам должен быть определен достаточно детально. Маршрутизаторы и межсетевые экраны могут использоваться для обеспечения того, что только определенные типы сетевого трафика могут передаваться в каждый сегмент.

Шифрование и протоколы

Шифрование и протоколы работают как технические меры для защиты информации, проходящей через сети и находящейся на компьютерах. Это обеспечивает гарантии, что информация получена именно тем, кому была направлена, и что она не была изменена в процессе передачи. Эти защитные меры могут обеспечить конфиденциальность и целостность данных, а также ограничить возможные маршруты взаимодействия. (Криптография и механизмы шифрования детально рассматриваются в Домене 06).

Аудит

Средства аудита – это технические меры, которые отслеживают действия в рамках сети, на сетевых устройствах или на отдельных компьютерах. Аудит позволяет выявить не только действия, в выполнении которых было отказано пользователю, но и производить мониторинг, позволяющий определить вид использованного доступа к ресурсу, обнаружить недостатки в системе безопасности, выявить подозрительную деятельность. Эта информация может использоваться для обнаружения слабых мест в других технических средствах защиты и помочь администратору понять, где и что нужно изменить, чтобы достичь необходимого уровня безопасности окружения.

ПРИМЕЧАНИЕ. Многие из вопросов, которых мы коснулись в этих разделах, будут более подробно рассматриваться в следующих доменах. На данный момент важно понимать, что административный, технический и физический уровни должны работать совместно, чтобы обеспечить надлежащее управление доступом.

8. Типы управления доступом

Различные типы управления доступом (административный, физический и технический) работают на разных уровнях в рамках своих категорий. Например, охранники являются разновидностью защитных мер, которые предназначены для того, чтобы отпугнуть злоумышленника и предоставить доступ в здание только уполномоченному персоналу. Если злоумышленник сможет так или иначе преодолеть или обмануть охранников, его зафиксируют датчики движения, остановят замки на дверях, сработает сигнализация. Эти уровни изображены на рисунке 2-17.

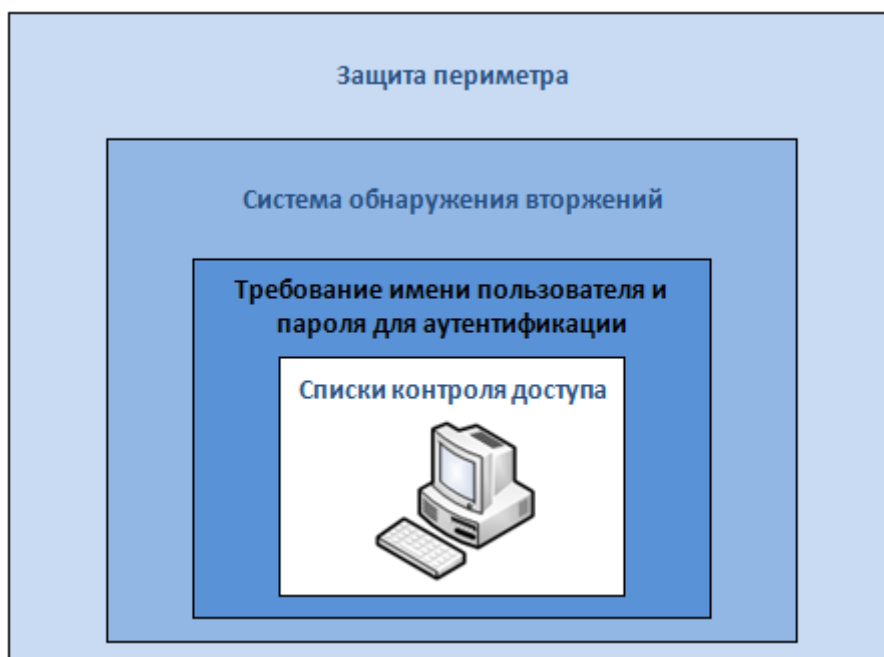


Рисунок 2-17. Безопасность должна быть реализована в виде отдельных уровней, которые ставят различные барьеры для злоумышленника

Различные защитные меры работают с разной степенью детализации и имеют разную функциональность. Существуют следующие виды мер: *превентивные, детективные, корректирующие, сдерживающие (устрашающие), восстанавливающие, компенсирующие и руководящие.*

Хорошо понимая функции различных защитных мер, вы сможете принимать более правильные решения о том, какие меры лучше всего использовать в той или иной ситуации. Существует семь разновидностей функциональности управления доступом:

- **Сдерживание** – предназначены для сдерживания (устрашения) потенциального злоумышленника
- **Превентивные меры** – предназначены для предотвращения инцидента
- **Корректировка** – предназначена исправление компонентов или систем после инцидента
- **Восстановление** – предназначено для восстановления функционирования защитных мер
- **Детективные меры** – помогают выявить связанную с инцидентом деятельность
- **Компенсирующие меры** – обеспечивают альтернативные варианты защиты
- **Руководящие меры** – обязательные защитные меры, которые должны быть реализованы в соответствии с действующими требованиями

Для обеспечения безопасности окружения наиболее эффективно применять превентивную модель, а затем для поддержки этой модели использовать детективные, восстанавливающие и корректирующие механизмы. Обычно мы хотим предотвратить проблемы и не допустить, чтобы они произошли, однако нужно быть готовыми адекватно реагировать на неприятности, если они все-таки случаться. Все защитные меры следует строить на основе концепции превентивной защиты. Однако невозможно предотвратить все, и мы должны иметь возможность быстро обнаруживать проблемы, которые мы не в состоянии предотвратить. Именно поэтому всегда следует внедрять одновременно и превентивные и детективные меры, которые будут дополнять друг друга. Если мы обнаруживаем проблему,

мы должны принять корректирующие меры для исправления ситуации, а также для того, чтобы предотвратить это в следующий раз. Таким образом, все три вида защитных мер работают вместе: превентивные, детективные и корректирующие.

Описанные далее типы защитных мер (административные, физические и технические) являются превентивными. Это важно понимать при разработке модели безопасного управления доступом, а также при сдаче экзамена CISSP.

8.1. Превентивные: Административные

Для обеспечения управления доступом и защиты компании в целом применяются следующие механизмы:

- Политики и процедуры
- Эффективная практика приема персонала на работу
- Проведение проверок кандидатов перед приемом на работу
- Управляемый процесс увольнения
- Классификация и маркирование данных
- Повышение осведомленности по вопросам безопасности

ПРИМЕЧАНИЕ. Очень хорошей практикой является требование от сотрудников подписание соглашения о неразглашении информации и порядке использования ресурсов, к которым им будет предоставлен доступ. Это соглашение может использоваться как некая гарантия неразглашения информации компании после увольнения сотрудника, либо как основание для привлечения сотрудника к ответственности за ненадлежащее использование ресурсов компании, либо разглашение информации. Именно неправильное управление доступом является причиной большинства случаев несанкционированного доступа.

8.2. Превентивные: Физические

Следующие механизмы могут физически ограничивать доступ в здание, отдельные помещения или к компьютерным системам.

- Бейджи, магнитные карты
- Охрана, собаки
- Ограждения, замки, шлюзы

8.3. Превентивные: Технические

Следующие механизмы являются логическими мерами и могут быть реализованы, как часть операционной системы, в виде приложений третьих фирм или в виде аппаратных устройств.

- Пароли, биометрия, смарт-карты
- Шифрование, протоколы, системы обратного вызова, представления баз данных, ограниченные пользовательские интерфейсы
- Антивирусное программное обеспечение, ACL, межсетевые экраны, маршрутизаторы, уровни отсечения (clipping level)

Таблица 2-3 показывает, как эти категории механизмов управления доступом выполняют различные функции безопасности. Однако следует отметить, что таблица 2-3 не покрывает все их возможности. Например, ограждения могут обеспечивать как превентивные, так и сдерживающие меры, усложняя злоумышленникам доступ в здание, но они также могут быть и компенсирующей мерой. Если компания не может себе позволить нанять охрану, она может установить ограждения в качестве компенсирующей физической меры. Каждая защитная мера может использоваться шире, чем это указано в таблице. Таблица 2-3 – это

только пример, показывающий взаимоотношения между различными защитными мерами и атрибутами безопасности, которые они предоставляют.

Тип защитных мер:																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																				
-------------------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Таблица 2-3. Функциональность защитных мер

ПРИМЕЧАНИЕ. Замки обычно считают задерживающими механизмами, поскольку они только задерживают злоумышленника на некоторое время. Их цель – задержать злоумышленника на время, достаточное для реакции на инцидент правоохранительным органам или охране.

Любые меры могут, в конечном счете, быть компенсирующими. Компания может выбрать компенсирующие меры, если другие меры слишком дороги, но защита все-таки необходима. Например, компания не может себе позволить нанять охрану, поэтому она устанавливает ограждения, которые будут являться компенсирующей мерой. Другой причиной использовать компенсирующие меры являются потребности бизнеса. Если специалисты по безопасности рекомендуют закрыть определенный порт на межсетевом экране, но бизнес требует доступности для внешних пользователей сервиса, который работает через этот порт, в этом случае могут быть внедрены компенсирующие меры в виде системы обнаружения вторжений (IDS), которая будет отслеживать весь трафик, поступающий на этот порт. Существует несколько видов механизмов безопасности, которые должны работать совместно. Однако сложность защитных мер и окружения, в котором они находятся, может

привести к тому, что некоторые меры будут противоречить друг другу, либо оставлять «дыры» в системе безопасности. Это, в свою очередь, может привести к появлению неожиданных недостатков в системе безопасности компании, которые не были выявлены или не были полностью поняты специалистами, внедрявшими защитные механизмы. Компания может иметь очень стойкие технические меры управления доступом, а также все необходимые административные меры, но если любой человек сможет получить физический доступ к любой системе в здании, возникнет реальная угроза безопасности всей среды. Все эти защитные меры должны работать в гармонии для обеспечения здорового, безопасного и продуктивного окружения.

9. Подотчетность

Возможности аудита обеспечивают ответственность пользователей за свои действия, проверку соблюдения политики безопасности, а также могут быть использованы при проведении расследований. Есть целый ряд причин, по которым сетевые администраторы и специалисты по безопасности хотят быть уверены, что механизмы журналирования событий включены и правильно настроены: для отслеживания неправильных действий пользователей, выявления вторжений, реконструкции событий и состояния системы, предоставления материалов по запросам правоохранительных органов, подготовки отчетов о проблемах. Журналы регистрации событий хранят горы информации – не так просто «расшифровать» ее и представить в наглядном и удобном для использования виде.

Подотчетность обеспечивается отслеживаем и записью системой действий пользователя, системы и приложения. Эта запись выполняется посредством функций и механизмов, встроенных в операционную систему или приложение. Журнал регистрации событий содержит информацию о действиях операционной системы, событиях приложений, а также действиях пользователей. Этот журнал может использоваться, в том числе, для проверки правильности работы системы, выявления ошибок и условий, при которых они возникают. После критического сбоя системы сетевой администратор просматривает журналы регистрации событий, пытаясь восстановить информацию о состоянии системы в момент аварии и понять какие события могли привести к ней.

Журналы регистрации событий могут также использоваться для выдачи предупреждений в случае выявления подозрительных действий, которые могут быть расследованы позднее. Дополнительно они могут быть очень ценны при определении, как далеко зашла атака, и какие повреждения она вызвала. Важно убедиться, что обеспечивается надлежащая защита журналов регистрации событий, гарантирующая, что любые собранные данные могут быть в случае необходимости предоставлены в неизменном виде и полном объеме для проведения расследований.

Целесообразно учитывать следующие моменты при организации системы аудита:

- Журналы регистрации событий должны храниться защищенным образом
- Следует использовать надежные инструменты работы с файлами журналов регистрации событий, которые сохраняют размер файлов журналов
- Журналы регистрации событий должны быть защищены от изменения неуполномоченными лицами
- Следует обучать соответствующий персонал правильным процедурам анализа данных в журналах регистрации событий
- Необходимо обеспечить гарантии того, что удаление журналов регистрации событий доступно только администраторам
- Журналы регистрации событий должны включать в себя действия всех высокопривилегированных учетных записей (root, administrator)

Администратор настраивает систему аудита, указывая, какие действия и события должны отслеживаться и журналироваться. В высокозащищенной среде администратор может настроить журналирование большего количества действий и установить порог критичности этих действий. События могут просматриваться для выявления недостатков в системе безопасности и нарушений политики безопасности. Если среда не требует такого уровня безопасности, анализируемых событий в ней может быть меньше, а пороги – выше.

Список событий и действий, которые подлежат журналированию, может стать бесконечным. Специалист по безопасности должен быть способен оценить среду и цели безопасности, знать, какие действия подлежат аудиту, понимать, что нужно делать с собранной информацией. При этом не должно расходоваться слишком много дискового пространства, процессорных ресурсов и рабочего времени персонала. Ниже приведен обзор групп событий и действий, которые могут быть выявлены и журналированы:

- **События системного уровня**

- Производительность системы
- Попытки регистрации пользователей (успешные и неудачные)
- Идентификатор регистрирующегося пользователя
- Дата и время каждой попытки регистрации
- Блокировка пользователей и терминалов
- Использование административных утилит
- Использование устройств
- Выполнение функций
- Запросы на изменение конфигурационных файлов

- **События прикладного уровня**

- Сообщения об ошибках
- Открытия и закрытия файлов
- Изменения файлов
- Нарушения безопасности в рамках приложения

- **События пользовательского уровня**

- Попытки идентификации и аутентификации
- Использование файлов, сервисов и ресурсов
- Выполнение команд
- Нарушения безопасности

Должен быть настроен порог (уровень отсечения) и параметры для каждого из этих событий и действий (например, администратор может настроить учет всех попыток регистрации, либо только неудачных). В качестве счетчиков производительности может использоваться, к примеру, объем дискового пространства, используемого за восьмичасовой период.

Система обнаружения вторжений (IDS) должна постоянно сканировать журнал регистрации событий на предмет подозрительных действий. При выявлении нежелательных событий или событий, свидетельствующих о попытке вторжения, журналы регистрации событий должны быть сохранены, чтобы использовать их позднее в качестве улик и доказательств. Если произошло серьезное событие безопасности, система IDS должна отправить

соответствующее уведомление администратору или другому персоналу, ответственному за реакцию на инциденты, для принятия оперативных мер и прекращения деструктивной деятельности. Например, если выявлена вирусная угроза, администратор может отключить почтовый сервер. Если злоумышленник получил доступ к конфиденциальной информации в базе данных, соответствующий сервер может быть временно отключен от сети или Интернета. Если атака продолжается, администратору может потребоваться проследить действия злоумышленника. IDS может показать эти действия в режиме реального времени и/или просканировать журналы регистрации событий и показать специфические последовательности (шаблоны) или поведение.

9.1. Анализ журналов регистрации событий

Журналы регистрации событий обязательно должны просматриваться и анализироваться. Это можно делать как вручную, так и с помощью автоматизированных средств. Если компания просматривает журналы вручную, необходимо систематизировать эту деятельность – что, как, когда и почему подлежит анализу. Обычно журналы регистрации событий становятся крайне востребованными после инцидента безопасности, непонятной работы системы или сбоя системы. Администратор или другой ответственный персонал пытается по-быстрому сопоставить различные действия, которые привели к этому инциденту. Такой тип анализа журналов регистрации событий называется ориентированным на события (event-oriented). Кроме того, журналы могут просматриваться на периодической основе, чтобы выявить необычное поведение пользователей и систем, а также убедиться в исправной работе системы. При этом перед администраторами должна быть поставлена соответствующая задача периодического просмотра журналов. Для контроля журналов в режиме реального времени (или близко к этому) существуют специальные автоматизированные средства. Данные журналов регистрации событий обычно должны анализироваться, а затем сохраняться в отдельное место для хранения в течение определенного периода времени. Это должно быть отражено в политике и процедурах безопасности компании.

Анализ журналов регистрации событий вручную крайне трудоемок. Следует использовать для этого специализированные приложения и инструменты анализа, которые сокращают объем журналов и повышают эффективность ручных процедур анализа. Основное время при анализе журналов регистрации событий тратится на несущественную информацию, а эти инструменты позволяют выбирать необходимую информацию по заданным критериям и представлять ее в более наглядной и удобной форме.

Существует три основных типа инструментов анализа журналов регистрации событий:

- **Инструменты для уменьшения объема журналов** (audit-reduction tool) отбрасывают обычные события работы программ и пользователей, записи счетчиков производительности системы, оставляя только те события, которые могут быть интересны специалистам по безопасности и администраторам
- **Инструменты для выявления нарушений** (variance-detection tool) отслеживают использование компьютера или ресурса, фиксируя стандартную картину, а затем выявляют отклонения от нее. Например, обычно использование ресурса происходит с 8:00 до 17:00 по рабочим дням. Факт использования этого ресурса ночью в выходной день является поводом отправки предупреждения администратору.
- **Инструменты выявления сигнатур атак** (attack signature-detection tool) имеют специализированную базу данных, содержащую информацию о событиях, которые могут указывать на определенные атаки (сигнатуры атак). В журналах отыскиваются последовательности событий, соответствующие сигнатурам атак.

9.2. Мониторинг нажатия клавиш

Мониторинг нажатия клавиш (keystroke monitoring) – это вид мониторинга, позволяющий проанализировать и записать последовательность нажатий клавиш пользователем в течение сеанса его работы. При этом все символы, набранные пользователем, записываются в специальный журнал, который может быть позднее проверен. Обычно такой тип аудита используется только для выполнения специальных задач и только на короткое время, поскольку объем собранной информации будет огромен, а ее значительная часть будет неважной. Если специалист по безопасности или администратор подозревают пользователя в проведении несанкционированных действий, они могут воспользоваться этим видом мониторинга. На некоторых санкционированных этапах проведения расследования между клавиатурой и компьютером может быть вставлено специальное устройство, перехватывающее все нажатия клавиш, включая набор пароля для загрузки системы (на уровне BIOS).

Хакеры также могут использовать такой вид мониторинга. Если атакующий сможет установить троянскую программу на компьютер, она сможет внедрить специальную утилиту перехвата данных, вводимых с клавиатуры. Обычно таким программам наиболее интересны учетные данные пользователя, и они могут уведомить атакующего, когда такие данные будут успешно перехвачены.

Нужно соблюдать осторожность при проведении такого мониторинга, т.к. это может нарушать законодательство. Следует заранее уведомить сотрудников о возможности такого мониторинга, а также получать разрешение руководства на его проведение. Если компания намерена использовать такой контроль, следует внести информацию об этом в политику безопасности, сообщать об этом на тренингах по вопросам безопасности, уведомить пользователей другими доступными способами о возможности такого мониторинга. Это позволит защитить компанию от претензий в нарушении неприкосновенности частной жизни, а также проинформирует пользователей об ограничениях использования рабочего компьютера в личных целях.

9.3. Защита данных аудита и журналов регистрации событий

Если воры ограбили квартиру, они стараются не оставлять следов (например, отпечатков пальцев), которые позволят связать произошедшее преступление с ними. То же самое происходит при компьютерном мошенничестве и нелегальной деятельности.

Злоумышленник старается скрыть следы своей деятельности. Часто атакующие удаляют журналы регистрации событий, которые содержат компрометирующую их информацию, либо удаляют из журналов регистрации событий только информацию о своем присутствии (scrubbing). Это может привести к тому, что администратор не узнает о произошедшем инциденте безопасности. Поэтому данные журналов регистрации событий должны быть защищены посредством строгого управления доступом к ним.

Только определенные лица (администраторы и персонал подразделения безопасности) должны иметь доступ на просмотр, изменение или удаление журналов регистрации событий. Никакие другие люди не должны иметь возможность просматривать данные журналы и, тем более, изменять или удалять их. Целостность этих данных может быть обеспечена средствами ЭЦП, хэш-функциями и строгим управлением доступом. Их конфиденциальность может быть защищена с помощью шифрования и управления доступом. Кроме того, целесообразно сохранять данные журналов регистрации событий на внешних носителях информации с однократной записью (например, компакт-дисках) для предотвращения потери или модификации информации. Все попытки несанкционированного доступа к журналам регистрации событий должны фиксироваться и своевременно анализироваться.

Журналы регистрации событий могут использоваться в качестве доказательств или улик, а

также для демонстрации процесса атаки или подтверждения факта инцидента. Целостность и конфиденциальность этих журналов должна постоянно контролироваться.

Ссылки по теме:

- **Authentication, Authorization, and Accounting (AAA) Charter**
- **Google authentication categories**
- **Security in *Open Systems*, NIST Special Publication 800-7 (July 1994)**

10. Практика управления доступом

Итак, мы рассмотрели, как идентифицировать пользователей, провести их аутентификацию, авторизацию и как контролировать их действия. Это необходимые части здоровой и безопасной сетевой среды. Но вы также хотите предпринять определенные шаги, чтобы убедиться в отсутствии ненужных открытых "дверей", а также в том, что среда остается на том же уровне безопасности, над достижением которого вы упорно трудились? Для этого необходимо внедрить хорошие практики управления доступом. Конечно, трудно устранять все проблемы в сети, бороться в политических баталиях, выполнять все запросы пользователей, и при этом своевременно выполнять все небольшие задачи по текущему обслуживанию. Тем не менее, невыполнение этих небольших задач вызывает больше всего проблем и уязвимостей.

Ниже представлен список задач, которые следует выполнять на регулярной основе, чтобы безопасность оставалась на достаточном уровне:

- Запретить доступ к системам пользователям, не прошедшим аутентификацию, и анонимным учетным записям.
- Ограничить и контролировать использование административных и иных привилегированных учетных записей.
- Блокировать учетную запись или вносить задержку после нескольких неудачных попыток регистрации.
- Удалять учетные записи уволенных сотрудников сразу же после их ухода из компании.
- Блокировать учетные записи, которые не использовались 30 – 60 дней.
- Внедрить строгие критерии доступа.
- Применять принцип «должен знать» и принцип минимальных привилегий.
- Отключить ненужные функции системы, службы и порты.
- Заменить пароли «по умолчанию» для встроенных учетных записей.
- Ограничить и контролировать правила глобального доступа.
- Убедиться, что названия учетных записей не раскрывают должностных обязанностей пользователей, которым они принадлежат.
- Удалить излишние правила использования ресурсов учетными записями и группами.
- Удалить из списков доступа к ресурсам излишние идентификаторы пользователей, учетные записи и роли.
- Организовать периодическую смену паролей.
- Установить требования к паролям (по длине, содержимому, сроку действия, распространению, хранению и передаче).

- Организовать журналирование системных событий и действий пользователей, а также периодический просмотр журналов.
- Обеспечить защиту журналов регистрации событий.

Но даже если все перечисленные выше контрмеры внедрены и надлежащим образом контролируются, несанкционированный доступ к данным по-прежнему возможен. В следующем разделе будет подробнее рассказано об этих проблемах и соответствующих контрмерах.

10.1. Несанкционированное разглашение информации

Некоторые вещи могут сделать информацию доступной неуполномоченным лицам, что может привести к неблагоприятным последствиям. Это может происходить как умышленно, так и случайно. Информация может быть разглашена непреднамеренно, когда человек становится жертвой специализированной атаки (например, социальной инженерии, использования скрытых каналов, вредоносного программного обеспечения, перехвата электромагнитных излучений). Также, информация может быть разглашена случайно посредством повторного использования объектов, о котором рассказано далее. (Социальная инженерия обсуждается в Домене 01, а скрытые каналы – в Домене 03).

Повторное использование объекта

Проблема **повторного использования объекта** связана с получением другим субъектом носителя информации, который ранее содержал один или несколько объектов. Это означает, что прежде чем кому-то будет передан для использования жесткий диск, USB-накопитель или лента, они должны быть очищены от любой остаточной информации, которая по-прежнему может находиться на них. Это также относится к повторному использованию объектов процессами компьютера, например, ячеек памяти, переменных и регистров. Любая критичная информация, которая может быть оставлена процессом, должна надежно удаляться, прежде чем другому процессу будет разрешен доступ к этому объекту. Это обеспечивает невозможность получения посредством повторного использования объектов доступа к информации неуполномоченных лиц и любых других субъектов. Часто в процессе работы происходит хаотичный обмен USB-накопителями. Что если руководитель передал свой USB-накопитель сотруднику, не стерев с него информацию, среди которой были конфиденциальные отчеты о работе сотрудников подразделения и прогнозы сокращению штата в следующем году? Это может иметь крайне негативные последствия, если информация распространится среди сотрудников компании. Форматирование диска или удаление файлов реально удаляет только ссылки на файлы, но не сами файлы. Эти файлы остаются на диске и доступны, пока операционной системе не потребуется это пространство, и она не перезапишет информацию поверх этих удаленных файлов. Таким образом, носители информации, содержащие критичную информацию, должны подвергаться более серьезным мерам, обеспечивающим реальное удаление информации, а не только ссылок на файлы.

Критичные данные должны быть классифицированы (секретно, совершенно секретно, конфиденциально, неклассифицированные и т.д.) владельцами данных. Должно контролироваться хранение критичных данных и процесс их использования. Но и это не все! Перед повторным использованием носителя информации, на котором ранее хранились критичные данные, вся информация с него должна быть надежно удалена специализированными средствами. (Ответственность за выполнение этой работы обычно лежит на подразделении эксплуатации ИТ.) Если носитель содержит критичную информацию и не может быть очищен, должны быть предприняты шаги, обеспечивающие надлежащее уничтожение самого носителя информации, чтобы никто другой не мог получить эту информацию.

ПРИМЕЧАНИЕ. Иногда хакеры настраивают сектор на жестком диске, помечая его как «плохой», чтобы его не могла использовать операционная система, однако при этом сам сектор

остается исправным и в нем могут храниться вредоносные данные. Операционная система не будет записывать информацию в этот сектор, поскольку она думает, что он поврежден. Это один из способов скрытия данных. Некоторые загрузочные вирусы (boot-sector virus) способны размещать часть своего кода в специальном секторе жесткого диска, перезаписывая любые данные, которые там хранились ранее, а затем помечая его как «плохой».

Защита от утечки информации по техническим каналам

Все электронные устройства излучают электрические сигналы, которые могут содержать важную информацию. При этом если атакующий купит специальное оборудование и встанет на нужное место, он сможет перехватить эту информацию из электромагнитных волн и получить передаваемые данные без физического подключения к сетевым проводам.

Произошло уже немало инцидентов, в которых злоумышленники с помощью недорогого оборудования перехватывали электромагнитные излучения компьютеров. Это оборудование может воспроизводить потоки данных и отображать их на мониторе компьютера злоумышленника, позволяя ему получить доступ к секретной информации. И это не просто бред из шпионских романов, это происходит в реальном мире. Поэтому были разработаны соответствующие контрмеры против таких атак.

Защита от электронной слежки (TEMPEST) началась с исследования Министерства обороны США, а затем превратилась в стандарт, который описывает разработку контрмер, предназначенных для контроля побочных электрических сигналов, излучаемых электрическим оборудованием. Для подавления этих сигналов до определенного приемлемого уровня используется специальное экранирование, которое предотвращает перехват злоумышленниками информации из прослушиваемых электромагнитных волн. Существуют специальные стандарты на такое экранирующее оборудование, которые ранжируют его по степени экранирования. Производители этого оборудования должны быть сертифицированы по этим стандартам.

Устройства (мониторы, компьютеры, принтеры и т.д.) снабжают внешним металлическим покрытием, называемым «*клеткой Фарадея*». Это покрытие имеет необходимую толщину, от которой зависит количество излучаемого во внешний мир сигнала. Кроме того, в устройствах, ранжированных по уровню защиты от электронной слежки, изменены и другие компоненты, в первую очередь блоки питания, имеющие пониженное энергопотребление.

При соблюдении допустимых предельных уровней излучения, безопасность считается достаточной. Одобренные продукты должны обеспечить соблюдение этих предельных уровней излучения. Такая защита обычно требуется только для военных учреждений, хотя и другие высокозащищенные среды используют такие защитные меры.

Многие военные организации беспокоятся о паразитных радиоволнах, излучаемых компьютерами и другим электронным оборудованием, так как злоумышленник может перехватить их, реконструировать данные и получить доступ к секретам, которые должны были оставаться тайной.

Технологии защиты от электронной слежки сложны, дороги и громоздки, поэтому они применяются только для защиты данных высшего уровня конфиденциальности.

Существуют две альтернативы средствам защиты от электронной слежки – белый шум или использование концепции контрольных зон.

ПРИМЕЧАНИЕ. TEMPEST (защита от электронной слежки) – это название программы (а сейчас – стандарта), которая была разработана в конце 1950-х годов американским и английским правительствами, чтобы решить проблему электрических и электромагнитных излучений от электрического оборудования – в первую очередь от компьютеров. Оборудование для электронной слежки обычно используется разведками, военными, правительствами и правоохранительными органами, его продажа находится под постоянным контролем.

Белый шум (white noise) – это контрмера для противодействия извлечению информации из электрического излучения, которая представляет собой однородный спектр случайных

электрических сигналов. Белый шум распространяется по всему спектру в рамках полосы пропускания и злоумышленник не может отделить реальную информацию от случайного шума или случайной информации.

Контрольная зона (control zone) – это другая альтернатива оборудованию защиты от электронной слежки. В стенах некоторых зданий используются специальные материалы, проводящие электрические сигналы. Это не позволяет злоумышленникам получить доступ к информации, излучаемой в виде электрических сигналов сетевыми устройствами. Контрольные зоны являются разновидностью периметра безопасности и создаются для защиты от доступа неуполномоченных лиц к данным и компрометации критичной информации.

11. Мониторинг управления доступом

Мониторинг управления доступом – это метод отслеживания тех, кто пытается получить доступ к определенным ресурсам компании. Это важный детективный механизм, для реализации которого существуют различные технологии, такие как IDS, IPS, сетевые снифферы, хосты-приманки (honeypot). Недостаточно просто вложить деньги в антивирус и межсетевой экран, компаниям необходим контроль своих собственных внутренних сетей.

11.1. Выявление вторжений

Системы выявления вторжений (IDS – intrusion detection system) отличаются от традиционных межсетевых экранов, т.к. они созданы для выявления недостатков в системе безопасности – несанкционированного использования или атак компьютеров, сетей или телекоммуникационной инфраструктуры. IDS позволяют снизить ущерб от хакерских атак, взлома критичных компьютеров и сетевых устройств. Основная задача средства IDS – отмечать подозрительные действия в сети и своевременно уведомлять о них администратора (посредством подачи звукового сигнала, передачи сообщения на консоль управления, отправки SMS-сообщения на мобильный телефон и т.п.), либо даже автоматически вносить изменения в настройки ACL межсетевого экрана. Средства IDS могут просматривать потоки данных, находя в них последовательности битов, которые могут свидетельствовать о сомнительных действиях или событиях, либо осуществлять мониторинг системных журналов и иных файлов журналирования деятельности. Следует выявлять любое ненормальное поведение, которое может свидетельствовать о вторжении (или попытке вторжения).

Хотя существуют различные разновидности IDS, все они имеют три общих компонента: сенсоры, анализаторы и административные интерфейсы. Сенсоры собирают трафик или данные о действиях пользователей и отправляют их анализаторам, которые ищут в них подозрительные действия. В случае выявления анализатором таких действий (на которые он запрограммирован), он отправляет соответствующее сообщение в административный интерфейс. Существует два основных типа IDS: **уровня сети** (network-based), которые отслеживают весь сетевой трафик, и **уровня хоста** (host-based), которые анализируют действия в рамках одной компьютерной системы.

IDS могут быть настроены для выявления атак, анализа журналов аудита, прерывания соединений, уведомления администратора о происходящих атаках, защиты системных файлов, указания на уязвимости, которые должны быть учтены, а также для помощи в отслеживании действий отдельных хакеров.

IDS уровня сети

IDS уровня сети (NIDS – network-based IDS) используют сенсоры, являющиеся отдельными компьютерами с установленным на них специальным программным обеспечением, либо специальными выделенными устройствами (appliance). Каждый сенсор имеет сетевую карту (NIC – network interface card), работающую в режиме прослушивания сети (promiscuous

mode). Обычные сетевые карты получают только сетевые пакеты, адресованные этой сетевой карте, широковещательные (broadcast) и многоадресные (multicast) пакеты. Драйвер сетевой карты копирует данные из передающей среды и отправляет и отправляет его стеку сетевых протоколов для обработки. Если сетевая карта находится в режиме прослушивания, драйвер сетевой карты захватывает весь трафик, делает копии всех пакетов, а затем передает одну копию в стек TCP, а вторую копию – анализатору, для поиска определенных шаблонов (сигнатур).

NIDS контролирует сетевой трафик и не может увидеть действия, происходящие внутри отдельного компьютера. Для отслеживания таких действий следует использовать IDS уровня хоста.

IDS уровня хоста

IDS уровня хоста (HIDS – host-based IDS) может быть установлена на отдельные рабочие станции и/или серверы для выявления нежелательных или аномальных действий. HIDS обычно используются для обеспечения уверенности, что пользователи не удаляют системные файлы, не изменяют важные настройки или подвергают систему риску иными способами. Так, если NIDS понимает и контролирует сетевой трафик, средства HIDS ограничиваются только самим компьютером. HIDS не понимает и не отслеживает сетевой трафик, а NIDS не контролирует действия внутри системы. Каждое из этих средств имеет свои задачи и выполняет их своими способами.

В большинстве сред системы HIDS устанавливаются только на критичные серверы, а не на каждый компьютер в сети, поскольку это могло бы вызвать существенное повышение нагрузки на компьютеры и на администраторов.

Чтобы сделать жизнь немного проще, HIDS и NIDS могут быть одного из следующих типов:

- Сигнатурные (signature based)
 - Отслеживающие шаблоны (pattern matching)
 - Отслеживающие состояние (stateful matching)
- Основанные на аномалиях (anomaly based)
 - Основанные на статистических аномалиях (statistical anomaly-based)
 - Основанные на аномалиях протоколов (protocol anomaly-based)
 - Основанные на аномалиях трафика (traffic anomaly-based)
- Основанные на правилах (rule-based) или эвристические (heuristic-based)

Выявление вторжений на основе знаний или сигнатур

Знания об отдельных атаках накапливаются производителями IDS и хранятся в виде моделей, отражающих процесс их выполнения. Эти модели называются **сигнатурами**. Как только выявляется новый вид атаки, производитель сигнатурного IDS создает соответствующую сигнатуру, которая в последующем используется при проверке сетевого трафика для обнаружения такой же атаки. Разрешаются любые действия, которые не были идентифицированы как атака.

ПРИМЕЧАНИЕ. Выявление атак на основе сигнатур также называют обнаружением по шаблонам (pattern matching).

Примером сигнатуры является сетевой пакет, в котором адрес отправителя и получателя совпадают – это, так называемая, Land-атака. В Land-атаке хакер изменяет заголовок пакета и когда система получателя отправляет отправителю ответ, она отправляет его на свой же адрес. Сейчас это выглядит довольно безобидно, но все еще есть уязвимые системы, не имеющие программного кода, позволяющего распознать такую ситуацию, которая приводит

к их «зависанию» или перезагрузке.

Сигнатурные IDS являются наиболее популярными IDS в наше время, но их эффективность напрямую зависит от регулярности обновления баз сигнатур, как в антивирусном программном обеспечении. Этот тип IDS практически не защищает от новых атак, так как он не может их распознать до появления их сигнатур в его базе данных. Атаки или вирусы, обнаруженные в реальных средах, называются «дикими» (in the wild). Атаки или вирусы, существующие, но не выпущенные в реальный мир, называются «в зоопарке» (in the zoo). Это не шутка.

IDS на основе состояния

Перед тем, как углубиться в изучение работы IDS на основе состояния, вам следует понять, что представляет собой состояние системы. Каждое изменение в работе системы (вход пользователя, запуск приложения, взаимодействие приложений, ввод данных и т.д.) приводит к изменению состояния. С технической точки зрения, все операционные системы и приложения представляют собой просто множество строк команд, написанных для выполнения определенных функций над данными. Команды работают с переменными, которые содержат данные. Когда вы, например, используете утилиту «Калькулятор» и вводите цифру «5», специальная пустая переменная сразу же получает это значение. Введя эту цифру, вы изменяете состояние приложения. Когда приложения взаимодействуют друг с другом, они заполняют пустые переменные специальными наборами команд. Таким образом, переход состояния – это когда меняется значение переменной, что происходит постоянно в любой системе.

При проведении атак, происходят соответствующие изменения состояний (действия). Например, если атакующий выполняет удаленное переполнение буфера, происходят следующие изменения состояний.

1. Удаленный пользователь подключается к системе.
2. Удаленный пользователь отправляет данные приложению (объем передаваемых данных превышает выделенный для них буфер в соответствующей пустой переменной).
3. Данные принимаются, перезаписывая при этом буфер и, возможно, другие сегменты памяти.
4. Выполняется вредоносный код.

Таким образом, *состояние* – это «снимок» (snapshot) значений операционной системы в оперативной, полупостоянной и постоянной областях памяти. В IDS на основе состояния первоначальным состоянием является состояние перед началом атаки, а скомпрометированным состоянием – состояние после успешного вторжения. IDS имеет правила, которые описывают последовательность переходов состояния, свидетельствующих о происходящей атаке. Действия, которые происходят между первоначальным и скомпрометированным состояниями – это именно то, что ищет данный тип IDS. Он отправляет сигнал опасности, если любая последовательность переходов состояния совпадает с предварительно настроенными правилами. Этот тип IDS осуществляет поиск сигнатур атак в контексте потока действий, а не просто смотрит отдельные пакеты. Он также может выявить только известные атаки и требует частого обновления своих сигнатур.

Выявление вторжений на основе статистических аномалий

IDS на основе статистических аномалий (statistical anomaly-based IDS) – это системы, основанные на поведении. Они не используют сигнатуры. Вместо этого они создают профиль «нормальной» деятельности, работая в режиме обучения. Этот профиль строится на основе постоянного анализа происходящей в среде деятельности. Точность профиля и качество защиты зависят от времени нахождения IDS в режиме обучения. После создания

профиля, весь последующий трафик и деятельность сравниваются с ним. Все что не похоже на профиль, считается атакой, о которой направляется соответствующее уведомление. Такие IDS используют сложные статистические алгоритмы, выискивая аномалии в сетевом трафике и действиях пользователей. Каждому пакету присваивается рейтинг его «аномальности», который указывает на степень его отклонения от нормального профиля. Если рейтинг выше определенного порога «нормального» поведения, выполняется заранее определенное действие.

Преимуществом IDS на основе статистических аномалий является их возможность реагировать на новые типы атак, в том числе атаки «нулевого дня», для которых еще нет сигнатур или патчей. Эти IDS также могут выявлять «низкие и медленные» атаки, при которых атакующий пытается остаться ниже порога срабатывания средств мониторинга, отправляя пакеты понемногу в течение длительного периода времени. IDS на основе статистических аномалий могут выявлять эти типы атак, поскольку они достаточно отличаются от профиля.

Но есть и плохие новости. Крайне сложно создать для сети качественный профиль «нормальной» деятельности, который не приводит к огромному количеству ложных срабатываний. Это связано с тем, что в сети происходят постоянные изменения. Часто это приводит к тому, что компании просто отключают свои IDS, из-за того, что они требуют крайне много времени для своей надлежащей поддержки. Чтобы сократить число ложных срабатываний, нужен очень высококвалифицированный ИТ-персонал. Также, важным моментом является правильное определение порога срабатывания IDS.

Если атакующий выявляет наличие IDS в сети, он попытается определить ее тип, чтобы попробовать обойти ее. Для поведенческой IDS атакующий может попытаться совместить свою деятельность с шаблоном поведения сетевого трафика. Если это удастся ему, его деятельность будет выглядеть нормальной для IDS и поэтому останется не выявленной. Кроме того, крайне важно гарантировать отсутствие атак во время работы IDS в режиме обучения, т.к. иначе IDS будет считать такие атаки «нормальной» деятельностью и не будет сообщать о них в будущем.

Если компания решает использовать IDS на основе статистических аномалий, она должна убедиться, что ее сотрудники, которые будут внедрять и поддерживать систему, разбираются в проведении анализа протоколов и пакетов. Это связано с тем, что такие IDS (в отличие от IDS других типов) отправляют малоинформативные уведомления и сетевой инженер должен в каждом случае разбираться, в чем на самом деле заключается проблема. Например, сигнатурная IDS сообщает тип выявленной атаки, IDS на основе правил сообщает какое правило было нарушено. IDS на основе статистических аномалий просто сообщают о том, что произошло что-то «ненормальное», не соответствующее профилю.

ПРИМЕЧАНИЕ. Поведенческие IDS также называют эвристическими IDS. Этот термин означает создание новой информации из различных источников данных. IDS собирает различные факты из сети или систем и рассчитывает вероятность, что происходит атака. Если рассчитанный рейтинг превышает порог, передается сигнал тревоги.

Техники атак. При подготовке к атаке, хакеры чаще всего первым делом определяют, установлен ли IDS в сети, которую они собираются атаковать. Если IDS установлен, они проводят DoS-атаку на него, чтобы нарушить его работу. Другой тактикой является отправка IDS некорректных данных, которые заставят IDS отправить уведомление о начавшейся атаке, хотя никакой атаки в действительности не будет. Это делается для того, чтобы добиться отключения IDS специалистами компании из-за ее «неправильной» работы, либо чтобы отвлечь внимание этих специалистов на анализ некорректных пакетов, пока будет происходить реальная атака.

Что в имени? Сигнатурные IDS также известны как системы выявления некорректного использования (misuse-detection system), а поведенческие IDS – как системы, основанные на профилях (profile-based system).

Определение правильных порогов статистически значимых отклонений – это на самом деле

ключ к успешному использованию поведенческой IDS. Если порог установлен на слишком низком уровне, обычная деятельность будет часто ошибочно расцениваться в качестве атак (false positive, «ложное срабатывание»). Если порог, наоборот, установлен слишком высоко, некоторые вредоносные действия не будут выявлены (false negative, «нераспознавание»).

Когда IDS обнаруживает атаку, она может производить различные действия, в зависимости от своих возможностей и настроенных на ней политик. IDS может отправить уведомление на консоль управления, чтобы сообщить об атаке соответствующим специалистам; отправить сообщение по электронной почте или на мобильный телефон специалисту, ответственному за реакцию на атаки; сбросить соединение, с которого зарегистрирована атака; или перенастроить маршрутизатор или межсетевой экран, чтобы попытаться остановить все последующие похожие атаки. Реакция может варьироваться от блокировки конкретного IP-адреса, до перенаправления или блокировки отдельных видов деятельности.

IDS на основе аномалий протоколов

IDS на основе статистических аномалий могут использовать фильтры, основанные на аномалиях протоколов. Такие IDS имеют знания о каждом протоколе, который они контролируют. Аномалии протоколов выявляются на основании формата и поведения протокола. IDS строит модель (профиль) «нормального» использования каждого протокола. Нужно иметь в виду, что теоретическое использование протоколов описано в соответствующих RFC, но их реальная работа почти всегда отличается от теоретической, поскольку производители программного обеспечения в большинстве случаев не строго следуют RFC. Таким образом, большинство профилей отдельных протоколов является смесью из официальных и реальных вариантов их использования. Когда IDS включается в работу, она ищет аномалии, которые не совпадают с профилями, построенными для конкретных протоколов. Хотя в самих операционных системах и приложениях достаточно эксплуатируемых уязвимостей, большинство успешных атак используют уязвимости самих протоколов. Например, на канальном уровне модели OSI протокол ARP (Address Resolution Protocol) не имеет защиты от атак, при которых в таблицу ARP вставляются поддельные данные. На сетевом уровне протокол ICMP (Internet Control Message Protocol) может использоваться для Loki-атаки для передачи данных из одного места в другое, хотя этот протокол создан только для передачи информации о состоянии, но не пользовательских данных. Заголовки IP могут быть легко модифицированы для проведения спуфинга. На транспортном уровне пакеты в соединение между двумя системами могут быть внедрены TCP-пакеты для захвата сеанса связи (session hijacking attack).

ПРИМЕЧАНИЕ. Если злоумышленник взламывает компьютер и устанавливает на него бэкдор (backdoor, «черный ход»), он может взаимодействовать с этим компьютером через бэкдор, оставаясь при этом незаметным для межсетевого экрана и IDS. Хакеры знают, что небольшой объем кода можно вставлять в пакеты ICMP, которые будут затем интерпретироваться программным обеспечением бэкдора, установленного на скомпрометированной системе. Устройства безопасности обычно не настраиваются на мониторинг такого трафика, поскольку ICMP является протоколом, который (теоретически) передает только информацию о состоянии, а не команды или данные.

Интеграция фильтров, основанных на аномалиях протоколов, в любую сетевую поведенческую IDS является неплохой идеей, потому что формирование и доставка каждого пакета происходят с использованием множества протоколов, а для этих протоколов существует множество различных вариантов атак.

IDS на основе аномалий трафика

Большинство поведенческих IDS имеют фильтры, основанные на аномалиях трафика, которые выявляют изменения в шаблонах трафика, например DoS-атаки или появление новых сервисов в сети. Созданный профиль является неким базисом обычного трафика среды, и весь последующий трафик сравнивается с этим профилем. Как и во всех фильтрах, здесь требуется настройка порога срабатывания для уменьшения числа ложных

срабатываний (как ложных разрешений, так и ложных запретов). Такой тип IDS также способен выявлять неизвестные атаки.

IDS на основе правил

IDS на основе правил (rule-based IDS) используют другой подход, отличающийся от подхода сигнатурных IDS или IDS на основе статистических аномалий. Например, если сигнатурная IDS выявляет пакет, в котором все флаги заголовка TCP установлены в «1», он знает, что это свидетельствует об xmas-атаке, и отправляет соответствующее оповещение. IDS на основе статистических аномалий также достаточно прямолинейны. Например, если Боб зарегистрировался на своем компьютере в 6 утра, а в соответствии с его профилем это ненормально, IDS отправляет уведомление об этом, поскольку это выглядит как действия, которые должны быть проанализированы. IDS на основе правил поступает хитрее в зависимости от сложности применяющихся правил.

IDS на основе правил похожи на экспертные системы. Экспертная система основана на базе знаний (knowledge base), механизме логических выводов (inference engine) и программировании на основе правил (rule-based programming). Знания представляют собой правила, а анализируемые данные – факты. Знания системы описываются с помощью программирования на основе правил (*ЕСЛИ ситуация ТОГДА действие*). Эти правила применяются к фактам, к данным, получаемым сенсором, или к контролируемым системам. Например, в сценарии 1 IDS получает данные из журнала аудита системы и временно сохраняет их базе данных фактов, как показано на рисунке 2-18. Затем к этим данным применяются предварительно настроенные правила, которые проверяют, есть ли что-нибудь подозрительное в произошедших событиях. В нашем сценарии правило указывает, что *«ЕСЛИ пользователь root создал File1 И создал File2 ПРИ ЭТОМ оба эти файла находятся в одной директории ЗАТЕМ запустил «Административную Утилиту1» ТОГДА отправить уведомление»*. Это правило определяет, что если пользователь root создал два файла в одной директории, а затем запустил определенную административную утилиту, должно быть отправлено уведомление.

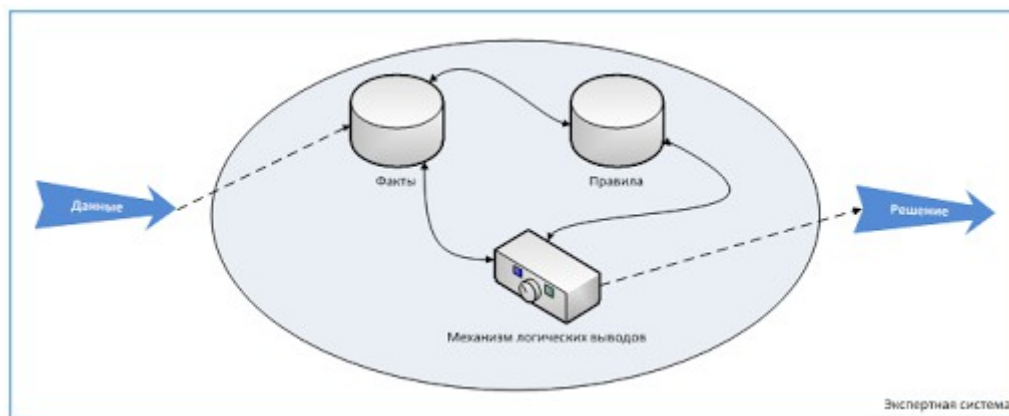


Рисунок 2-18. Компоненты IDS, основанной на правилах, и экспертной системы

Механизм логических выводов реализует некий искусственный интеллект в данном процессе. Он может делать выводы, создавая новую информацию из полученных данных, используя правила. Чтобы понять это, давайте взглянем на следующее.

Сократ – это человек.

Все люди смертны.

Поэтому мы можем сделать вывод, что Сократ – смертен.

Обычные языки программирования – это примерно как «черное и белое» в жизни. Они отвечают только «да» или «нет», но не «может быть да» или «может быть нет». Хотя компьютеры выполняют сложные вычисления гораздо быстрее, чем люди, им труднее делать догадки, выводы, давать ответы – поскольку они очень жестко структурированы. Языки

программирования пятого поколения (языки искусственного интеллекта) способны работать с «серым» цветом жизни и могут пытаться принимать правильные решения на основе полученных данных.

Таким образом, в IDS основанных на правилах, работающих аналогично экспертным системам, IDS собирает данные с сенсоров или из журналов аудита, а механизм логических выводов обрабатывает их, используя предварительно запрограммированные правила. Если характеристики удовлетворяют правилу, отправляется соответствующее уведомление или выполняется определенное действие для решения возникшей проблемы.

Типы IDS. Важно понимать характеристики, отличающие друг от друга различные типы технологий IDS. Ниже представлено краткое резюме:

- **Сигнатурные (signature-based)**
 - Отслеживающие шаблоны (pattern matching), работающие подобно антивирусному программному обеспечению
 - Сигнатуры должны постоянно обновляться
 - Не могут выявить новые атаки
 - Два типа:
 1. Отслеживающие шаблоны (pattern matching) – сравнивают пакеты с сигнатурами
 2. Отслеживающие состояние (stateful matching) – сравнивают действия с шаблонами
- **Основанные на аномалиях (anomaly-based)**
 - Основанные на поведении системы (behavioral-based), которые изучают «нормальную» деятельность в среде
 - Могут выявлять новые атаки
 - Также называются поведенческими или эвристическими
 - Три типа:
 1. Основанные на статистических аномалиях (statistical anomaly-based) – создают профиль «нормальной» деятельности и сравнивают реальную деятельность с этим профилем
 2. Основанные на аномалиях протоколов (protocol anomaly-based) – выявляют факты необычного использования протоколов
 3. Основанные на аномалиях трафика (traffic anomaly-based) – выявляют необычные действия в сетевом трафике
- **Основанные на правилах (rule-based)**
 - Используют ЕСЛИ/ТОГДА программирование, основанное на правилах, в рамках экспертных систем
 - Используют экспертную систему, имеющую характеристики искусственного интеллекта
 - Более сложные правила
 - Не может выявлять новые атаки

Ссылки по теме:

- “The Science of IDS Attack Identification” (Cisco Systems whitepaper)
- “Intrusion Detection Terminology (Part Two),” by Andy Cuff (SecurityFocus Infocus Archive, last updated Sept. 24, 2003)
- Honeypots.net IDS Software links page

- “State of the Practice of Intrusion Detection Technologies,” by Julia Allen, et al., Software Engineering Institute, Carnegie Mellon University (Jan. 1999)

Сенсоры IDS

IDS сетевого уровня используют сенсоры для осуществления мониторинга. Сенсор, который работает как аналитический движок (analysis engine), размещается в сетевом сегменте, который должна контролировать IDS. Сенсор получает «сырые» (raw) данные от генератора событий, как показано на рисунке 2-19, и сравнивает их с базой данных сигнатур, профилем или моделью – в зависимости от типа IDS. Если выявляется некое совпадение, которое свидетельствует о подозрительной активности, сенсор работает как модуль реагирования для определения вида действий, которые нужно предпринять (сообщение через систему мгновенных сообщений, мобильный телефон, электронную почту, перенастройка межсетевого экрана и т.д.). Сенсор предназначен для фильтрации поступающих на него данных, отбрасывая ненужную информацию и выявляя подозрительную деятельность.

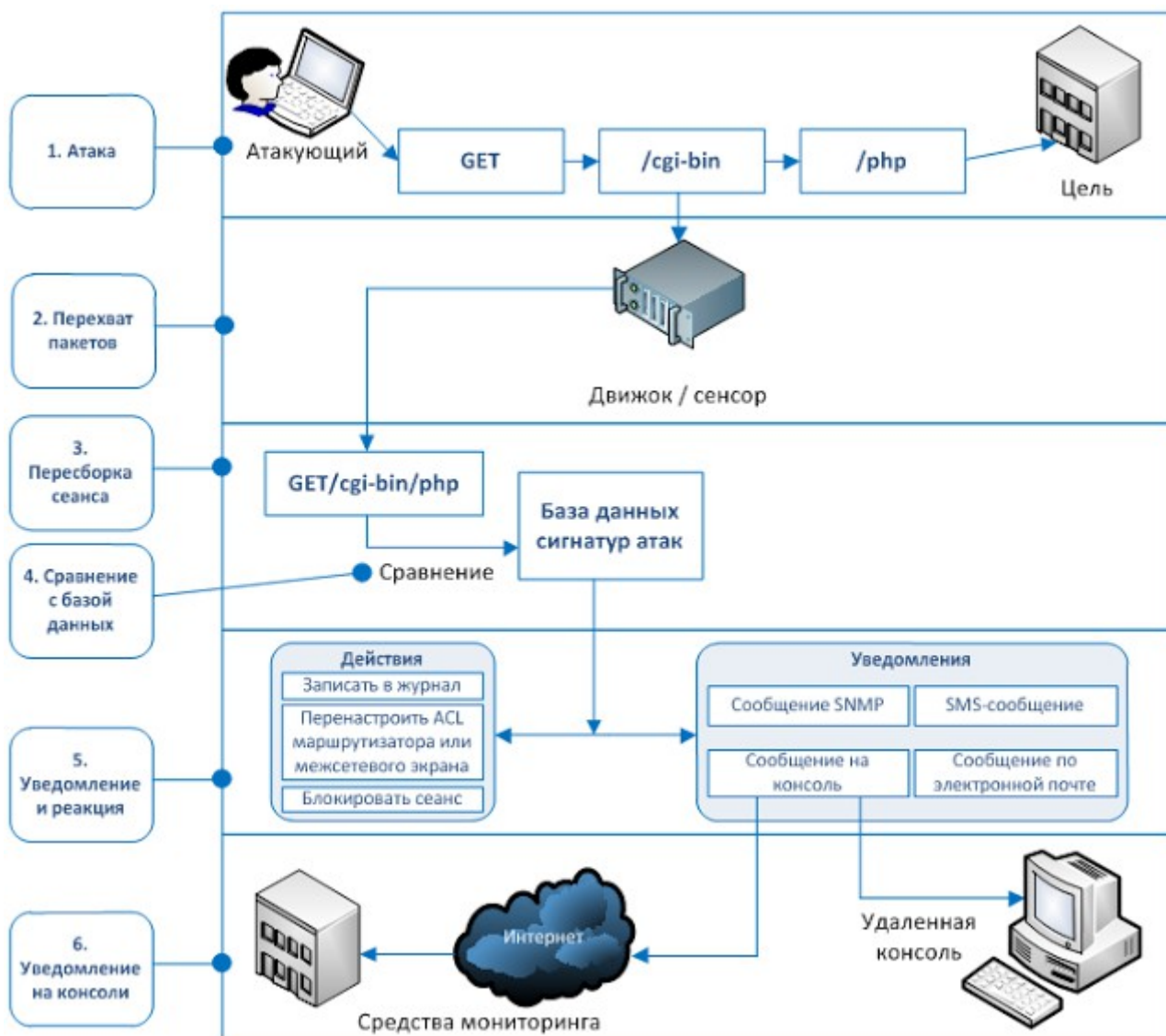


Рисунок 2-19. Основная архитектура NIDS

Коммутируемая среда. NIDS сложнее работать в коммутируемой среде по сравнению с традиционными, некоммутируемыми средами, поскольку данные в ней передаются через независимые виртуальные каналы, а не транслируются, как в некоммутируемых средах. Поэтому в коммутируемой среде сенсор должен быть подключен к специальному порту (spanning port) на коммуникационном устройстве, на который автоматически копируется весь трафик, проходящий через все виртуальные каналы. Это позволит сенсору иметь доступ ко всему трафику, проходящему через коммутируемую сеть.

Консоль мониторинга контролирует все сенсоры и предоставляет сетевому персоналу обзор работы всех сенсоров в рамках всей сети. Размещение сенсоров является одним из критичных этапов конфигурирования IDS. Компания может разместить один сенсор перед межсетевым экраном для выявления атак, а другой сенсор за межсетевым экраном (в сетевом периметре) для выявления реальных вторжений. Сенсоры следует также размещать в высококритичных областях, DMZ и в экстрасетях. Рисунок 2-20 показывает сенсоры, передающие данные на центральную консоль.

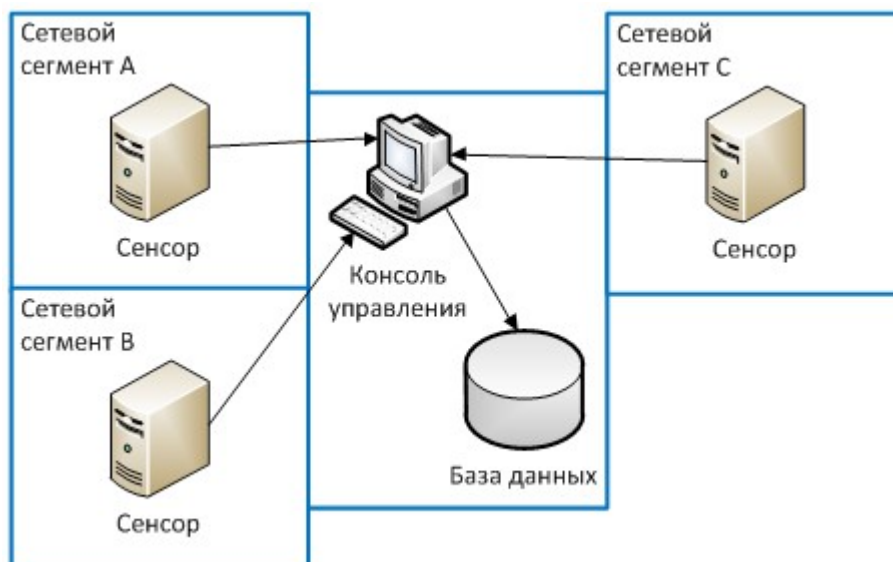


Рисунок 2-20. Сенсоры должны быть размещены в каждом сетевом сегменте, который подлежит контролю IDS

IDS может быть централизованной (например, встроенная в межсетевой экран), либо распределенной (с множеством сенсоров, распределенных по всей сети).

Сетевой трафик

Если объем сетевого трафика превышает порог IDS, отдельные атаки могут остаться незамеченными. Каждая IDS имеет собственный порог, о котором обязательно нужно знать до покупки и внедрения IDS.

В средах с очень большим объемом трафика следует размещать множество сенсоров, чтобы обеспечить уверенность, что все пакеты проанализированы. Если необходимо оптимизировать пропускную способность и скорость работы сети, можно настроить различные сенсоры на анализ каждого пакета на соответствие различным (отдельным) наборам сигнатур. Таким образом, нагрузка по анализу пакетов может быть разбита на несколько различных точек.

11.2. Системы предотвращения вторжений

Поскольку средства IDS не способны остановить доступ плохих парней к активам компании, а только выявляют такие факты и отправляют уведомление администратору, у компаний возникла потребность в новых продуктах и технологиях, позволяющих решить эту проблему. В результате появились **системы предотвращения вторжений** (IPS – intrusion prevention system), целью которых является не только выявление несанкционированной деятельности, но и предотвращение доступа злоумышленника к цели, как показано на рисунке 2-21. Таким образом, IPS является превентивной и проактивной технологией, сосредоточенной в первую очередь на предотвращении атак, в отличие от IDS, являющейся детективной технологией, применяемой к уже свершившимся фактам. Основной идеей является комбинирование средств IDS и IPS в одном продукте – межсетевом экране, который проводит глубокий

анализ сетевых пакетов, выявляет атаки и останавливает их. Как и в мире IDS, существуют средства IPS уровня хоста (HIPS) и уровня сети (NIPS). Большинство сетевых IPS являются линейными (inline) устройствами, включаемыми «в разрыв» сети и пропускающими через себя и контролирующими весь трафик. Однако это может привести к появлению «узкого места» (бутылочного горлышка), снизить производительность сети и стать единой точкой отказа. Производители упорно работают над технологиями, позволяющими решить все эти проблемы.



Рисунок 2-21. Архитектура IDS vs. IPS

Реакция на вторжения. Большинство IDS имеют несколько видов реакции на произошедшие события. IDS может отправить специальный сигнал, чтобы отбросить или уничтожить соединение как на стороне источника, так и на стороне получателя пакета. Это эффективно разрывает соединения и не позволяет осуществлять передачу. IDS может блокировать доступ пользователя к ресурсу на отдельной системе при достижении порога срабатывания IDS. IDS может отправить уведомление о событии на другой компьютер, на консоль управления или администратору. И, наконец, некоторые IDS могут перенастроиться для выполнения неких заранее определенных действий.

Большинство NIPS имеют два сетевых интерфейса – внешний и внутренний. Трафик приходит на внешний интерфейс, анализируются и, в случае признания пакетов безопасными, они направляются на внутренний интерфейс. Если пакеты признаются вредоносными, они отбрасываются.

Время покажет, смогут ли продукты IPS полностью заменить продукты IDS. Некоторым кажется, что это просто новый маркетинговый термин, тогда как другие считают IPS новым шагом в эволюции технологий информационной безопасности.

Ссылки по теме:

- “Intrusion Prevention Systems: The next step in the evolution of IDS,” by Neil Desai (SecurityFocus Infocus Archive, last updated Feb. 27, 2003)
- “Intrusion Prevention Systems (IPS), Part 1: Deciphering the inline Intrusion Prevention hype, and working toward a real-world, proactive security solution” (Secure Computing whitepaper, 2003)

Хосты-приманки

Хост-приманка (honeypot) – это специально настроенный компьютер-жертва, на котором открыто много портов, запущено много служб. Такой компьютер должен привлечь злоумышленника больше, чем реальные системы в сети. Хост-приманка не содержит никакой реальной информации компании, и нет никаких рисков в случае его успешного взлома.

Это позволяет администратору узнавать о происходящих атаках, своевременно укреплять сетевую среду, а также упрощает поиск злоумышленника. Чем больше времени хакер остается на этом компьютере, тем больше информации можно получить о его методах.

Очень важно провести четкую грань между **приманкой** (enticement) и **провокацией** (entrapment) при внедрении хостов-приманок. Следует учитывать при этом особенности действующего законодательства. Если система просто имеет открытые порты и службы, которыми злоумышленник может захотеть воспользоваться – это является примером приманки. Если же система имеет интернет-страницу, на которой указано, что пользователь может бесплатно скачать некие файлы, а администратор, как только пользователь сделает это, будет пытаться привлечь его к ответственности – это уже будет примером провокации. Провокация – это когда атакующего склоняют к совершению преступления. Провокация в большинстве случаев будет являться незаконной и не может быть использована для привлечения человека к ответственности за хакерство или несанкционированные действия.

Ссылки по теме:

- The Honeynet Project
- “What Is a Honeypot?” by Loras Even, SANS (July 12, 2000)

Сетевые sniffеры

Пакетный или сетевой **сниффер** – это программа или устройство, позволяющее анализировать трафик в сетевом сегменте. Трафик, который передается по сетевой среде, представляет собой электрические сигналы, закодированные с помощью двоичного представления. Сниффер должен иметь возможности для анализа протоколов, чтобы распознать различные протоколы и правильно интерпретировать их.

Сниффер должен иметь доступ к сетевому адаптеру, работающему в режиме прослушивания сети (promiscuous mode), и драйверу, который захватывает данные. Объем этих данных может быть огромен, поэтому они должны надлежащим образом фильтроваться.

Отфильтрованные данные сохраняются в буфере, показываются пользователю и/или сохраняются в специальном журнале. Некоторые утилиты имеют сниффер и средства модификации сетевых пакетов, что позволяет им проводить некоторые виды атак (например, спуфинг или атаки «человек-по-середине» (man-in-the-middle attack) и т.п.).

Сетевые снифферы обычно используются администраторами и специалистами по безопасности («белыми шляпами») для попытки выявления проблем в сети. Но эти же средства могут использовать злоумышленники и хакеры («черные шляпы») для анализа данных, проходящих в определенном сетевом сегменте, перехвата паролей и другой критичной информации, несанкционированной модификации данных, а также для проведения различных атак.

ПРИМЕЧАНИЕ. Снифферы очень опасны, т.к. крайне сложно выявить факт их работы в сети.

12. Несколько угроз управлению доступом

Большинство специалистов по безопасности знают, что более высокие риски и больший ущерб для компании несут атаки внутренних злоумышленников, чем внешних. Однако многие люди не знают этого, поскольку они слышали только истории о внешних хакерах, которые делали дефейс веб-сайтов или обходили межсетевые экраны, чтобы получить доступ к внутренней конфиденциальной информации.

Внешние атакующие могут войти в сеть компании через точки удаленного доступа, пройти через межсетевые экраны и веб-серверы, взломав их, или воспользоваться коммуникационными каналами партнеров (экстранет, подключения производителей и т.д.). А инсайдеры изначально имеют разрешенный доступ к системам и ресурсам, они могут использовать свои привилегии ненадлежащим образом или проводить реальные атаки.

Опасность инсайдеров состоит в первую очередь в том, что они уже имеют широкие права доступа, а внешнему хакеру нужно постараться, чтобы получить даже небольшую часть этих прав. Кроме того, инсайдеры гораздо лучше знают внутреннюю среду компании, и, как правило, им доверяют. Мы рассмотрели много различных механизмов управления доступом, которые нужны для того, чтобы оставить внешних лиц снаружи и ограничить до минимума возможности инсайдеров, а также вести аудит их действий. Теперь мы рассмотрим несколько самых популярных в настоящее время атак, проводимых внешними и внутренними злоумышленниками.

12.1. Атака по словарю

Некоторые программы позволяют атакующему (или проактивному администратору) подобрать пароли пользователей по словарю. Такие программы имеют список (словарь) наиболее часто используемых в качестве паролей слов или комбинаций символов, они последовательно хэшируют их и сравнивают с перехваченным хэшем пароля или системным файлом с паролями, который также хранит не сами пароли, а результаты односторонних хэш-функций над ними. Если хэш-значения совпадают, это означает, что пароль успешно подобран. Полученный таким способом пароль атакующий может использовать для аутентификации от имени уполномоченного пользователя. Однако, поскольку многие системы ограничивают количество неудачных попыток регистрации, такой подбор возможен только в случае перехвата передаваемого по сети хэша пароля или получения системного файла с паролями.

Файлы базовых словарей поставляются вместе с программами взлома паролей, а расширенные варианты словарей можно найти в сети интернет.

ПРИМЕЧАНИЕ. Пароли никогда не должны храниться или передаваться в виде открытого текста. Большинство операционных систем и приложений обрабатывает пароли с помощью алгоритмов хэширования, которые выдают в результате значения хэшей, называемые также значениями дайджестов сообщений.

Контрмеры

Для надлежащей защиты систем от взлома паролей по словарю (или другими методами подбора паролей), необходимо обеспечить следующее:

- Не допускайте отправки паролей открытым текстом
- Шифруйте или хэшируйте пароли
- Используйте токены для генерации одноразовых паролей
- Используйте сложно угадываемые пароли
- Чаще меняйте пароли
- Используйте IDS для выявления подозрительных действий
- Самостоятельно используйте средства взлома паролей для нахождения слабых паролей, выбранных пользователями
- Используйте специальные символы, цифры, заглавные и строчные буквы в пароле
- Защищайте файлы, в которых хранятся пароли

12.2. Атака полного перебора (брутфорс-атака)

Атака полного перебора (брутфорс-атака - brute force attack) – это последовательный перебор всех возможных комбинаций символов до нахождения комбинации, подходящей в качестве пароля.

Наиболее эффективны гибридные атаки, совмещающие в себе атаки по словарю с брутфорс-

атаками. Если атака по словарю (или сам атакующий) определяет, что пароль начинается со слова Dallas, брутфорс-атака пытается добавлять к нему различные символы, пока не будет подобран подходящий пароль.

Такие атаки применяются при проведении атак wardialing, при которых атакующий вставляет длинный список телефонных номеров (или целые диапазоны номеров) в специальную программу автоматического обзвона, в надежде найти номера модемов (по их отклику), которые можно использовать для несанкционированного доступа в сеть. При этом из таких списков обычно заранее исключаются заведомо известные голосовые номера (например, номера справочных служб), а также известные номера факс-машин. Получив номера, с которых ответили модемы, атакующий пытается использовать их для несанкционированного доступа к сети или системам.

Таким образом, брутфорс-атака представляет собой определенный вид деятельности с различными входными параметрами, выполняющейся пока не будет достигнута цель.

Контрмеры

Для противодействия телефонным брутфорс-атакам следует использовать журналирование и мониторинг таких действий для выявления шаблонов, которые могут свидетельствовать об атаке wardialing:

- Самостоятельно выполняйте брутфорс-атаки для поиска слабозащищенных и доступных модемов
- Убедитесь, что только необходимые телефонные номера сделаны общедоступными
- Используйте строгие механизмы управления доступом, что снизит вероятность успеха брутфорс-атак
- Ведите журналирование и анализируйте такие действия
- Используйте IDS для выявления подозрительной деятельности
- Установите порог блокировки

12.3. Подделка окна регистрации в системе

Злоумышленник может использовать программу, предоставляющую пользователю поддельный экран регистрации, в который обманутый пользователь вводит свои учетные данные, пытаясь зарегистрироваться в системе. У пользователя запрашивается имя и пароль, которые затем сохраняются для последующего использования злоумышленником. При этом пользователь думает, что это обычный экран регистрации, поскольку он выглядит точно также. После ввода пользователем своих учетных данных выводится поддельное сообщение об ошибке, говорящее ему о том, что он ошибся при вводе своих учетных данных. При этом поддельная программа закрывается, и управление передается операционной системе, которая выводит настоящее окно регистрации, запрашивая у пользователя имя и пароль. Пользователь думает, что сделал опечатку при вводе пароля и регистрируется повторно, но атакующий теперь знает его учетные данные.

Контрмеры

- В Windows-системах требовать нажатия пользователем CTRL+ALT+DEL перед вводом учетных данных
- Настроить операционную систему для отображения количества произошедших неудачных попыток регистрации

12.4. Фишинг

Фишинг – это разновидность социальной инженерии, которая направлена на получение

персональной информации, учетных данных, номеров кредитных карт или финансовых данных. Атакующий «выуживает» критичные данные различными способами.

Термин *фишинг* появился в 1996 году, когда хакеры начали воровать пароли доступа к провайдеру America Online (AOL). Хакеры представлялись сотрудниками AOL и отправляли сообщения жертвам с запросом их паролей «для проверки правильности информации биллинга» или «проверки правильности работы учетной записи в AOL». Когда пользователь отправлял свой пароль, злоумышленник аутентифицировался от его имени и использовал его электронную почту для криминальной деятельности, такой как рассылка спама, порнографии и т. п.

Хотя фишинг существовал еще в 1990-х, большинство людей не знали о нем до середины 2003 года, когда фишинговые атаки участились. Фишеры направляли жертвам по электронной почте убедительные сообщения, прося их перейти по ссылке, чтобы обновить информацию об их банковском счете. Жертвы переходили по этой ссылке, и перед ними появлялась форма, запрашивающая номера банковских счетов, учетные данные и другие виды данных, которые использовались затем для кражи личности. Количество фишинговых сообщений электронной почты очень быстро растет последние годы. Фишеры представляются крупными банками, платежными системами и другими хорошо известными компаниями.

Фишеры создают веб-сайты, которые выглядят очень похоже на настоящие сайты и заманивают на них жертв посредством других сайтов и рассылок по электронной почте с целью сбора их персональной информации. Такие сайты требуют от жертвы ввести свой номер социального страхования, дату рождения, девичью фамилию матери с целью «аутентификации для обновления информации о своих счетах».

Поддельные веб-сайты не только выглядят и работают как настоящие веб-сайты, злоумышленники часто используют еще и доменные имена, очень похожие на адреса настоящих сайтов. Например, www.amzaon.com вместо www.amazon.com. Или используют специальным образом размещенный символ @. например, адрес www.msn.com@notmsn.com в действительности приведет жертву на сайт notmsn.com и автоматически введет на нем имя пользователя www.msn.com. Такое имя пользователя на сайте notmsn.com будет отсутствовать, поэтому жертве будет просто показана стартовая страница notmsn.com, поддельного сайта, выглядящего и работающего подобно www.msn.com. Жертва чувствует себя в полной безопасности и вводит свои учетные данные на этом сайте.

Были разработаны даже некоторые функции JavaScript для того, чтобы показывать жертве некорректный веб-адрес в адресной строке браузера. С помощью них, например, можно заменить в адресной строке браузера адрес www.evilandwilltakeallyourmoney.com на www.citibank.com, чтобы жертва (даже та, которая проверила правильность адреса) чувствовала себя в полной безопасности.

ПРИМЕЧАНИЕ. Использувавшиеся ранее варианты атак, заменяющие содержимое адресной строки, уже были исправлены. Однако важно понимать, что злоумышленники постоянно находят новые способы проведения таких атак. Одно только знание о фишинге не означает, что вы сможете выявить или предотвратить его. Как специалист по безопасности, вы должны быть осведомлены о новых стратегиях и трюках, использующихся злоумышленниками.

Некоторые атаки используют «всплывающие» веб-формы, появляющиеся когда жертва находится на легитимном сайте. Например, когда вы находитесь на реальном веб-сайте банка, всплывает окно, запрашивающее у вас некоторую конфиденциальную информацию. Вероятно это не вызовет беспокойства, если вы уверены, что находитесь на реальном сайте банка. Вы поверите, что это окно относится к банковскому сайту, и заполните форму в соответствии с инструкциями. К сожалению, это всплывающее окно может иметь другой источник, и ваши данные могут попасть в руки злоумышленнику, а не вашему банку.

Получив эту персональную информацию, злоумышленник сможет открыть новый счет на имя жертвы, получить несанкционированный доступ к ее банковскому счету, сделать нелегальную покупку по кредитной карте или даже снять наличные. Согласно отчету Gartner в 2003 году убытки от фишинга составили 2,4 миллиарда долларов. По оценке Gartner 57 миллионов людей в США получали за 2003 год фишинговые сообщения, а 1,8 миллионов из них в ответ передавали свою персональную информацию.

Поскольку все больше людей узнает о таких видах атак, и они опасаются переходить по ссылкам, указанным в сообщениях электронной почты, фишеры изменили свои методы. Например, они стали направлять электронные письма, которые говорят пользователям, что они выиграли приз или что существует проблема с их банковским счетом. Сообщение электронной почты говорит пользователю, что ему нужно позвонить на некий телефонный номер, на котором автоматизированная голосовая система требует от человека ввести для аутентификации номер своей кредитной карты или номер социального страхования.

В 2006 году, было выявлено по крайней мере 35 фишинговых веб-сайтов, которые использовались для атак на многие банки, применяющие аутентификацию на основе токенов с одноразовыми паролями. Действующие в США правила требуют от финансовых компаний внедрять двухфакторную аутентификацию для онлайн-транзакций. Чтобы удовлетворить этому требованию, некоторые банки предоставили своим клиентам аппаратные токены, генерирующие одноразовые пароли. Для противодействия этому, фишеры создали поддельные веб-сайты, которые выглядели как сайт финансовой компании, и обманывали жертв, требуя их ввести свои одноразовые пароли. Затем эти веб-сайты отправляли эти учетные данные на реальный веб-сайт банка, аутентифицируясь от имени законных пользователей и получая доступ к их счетам.

Похожий тип атак называется **фармингом** (pharming), при котором жертва перенаправляется на поддельный сайт, выглядящий как легитимный. В этом типе атаки, злоумышленник использует отравление DNS (DNS poisoning), при котором сервер DNS неправильно разрешает имя узла в IP-адрес. Например, при вводе www.nicebank.com в адресную строку вашего браузера, компьютер в действительности не знает, что это такое. Браузер просматривает настройки TCP/IP, и находит в них IP-адрес DNS-сервера. Затем он посылает запрос на этот DNS-сервер, спрашивая "есть ли у тебя IP-адрес для www.nicebank.com?" DNS-сервер просматривает свои записи о ресурсах, и, найдя информацию об этом сайте, он посылает IP-адрес сервера, на котором размещена страница www.nicebank.com, обратно вашему компьютеру. Получив IP-адрес, браузер показывает главную страницу запрошенного вами сайта банка.

Теперь представьте, что будет, если злоумышленник изменит («отравит») кэш этого DNS-сервера таким образом, чтобы запись об этом ресурсе имела неправильную информацию? При вводе пользователем www.nicebank.com и отправке его системой запроса на DNS-сервер, он пришлет имеющуюся у него информацию об IP-адресе веб-сервера, не зная, что она неверна. И вместо www.nicebank.com, он отправит пользователя на www.thethief.com, который выглядит и работает точно также как и запрошенный веб-сайт. Пользователь, ничего не подозревая, вводит на нем свое имя и пароль, которые сохраняются для злоумышленника.

Преимуществом фарминг-атак для злоумышленника является то, что такие атаки могут оказать влияние на большое количество жертв, без необходимости отправки им электронной почты. Кроме того, таким образом проще обмануть жертв, поскольку они сами переходят на эти веб-сайты.

Контрмеры

Контрмеры против фишинговых атак включают следующие:

- Скептически относитесь к пришедшим по электронной почте требованиям ввести

свои учетные данные или финансовую информацию, предупреждениям о блокировке учетной записи, если не выполнить определенные действия на некоем сайте

- Звоните в реальную компанию для подтверждения полученного по электронной почте требования
- Проверяйте правильность ввода имени домена в адресной строке браузера
- Вводите учетные данные или финансовую информацию только при наличии SSL-соединения и изображения закрытого замка в соответствующем месте браузера
- Не переходите по ссылкам, полученным в HTML-сообщениях электронной почты – вручную копировать их в адресную строку браузера
- Не принимайте электронную почту в формате HTML

12.5. Кража личности

Кража личности (identity theft) относится к ситуации, когда кто-то получает ключевые элементы личной информации, например, номер водительских прав, номер банковского счета, учетные данные или номер социального страхования, а затем использует эту информацию для того, чтобы выдавать себя за другого. Как правило, украденные личные данные используются для получения кредитов, приобретения товаров или услуг, получения доступа к компьютерным системам от имени жертвы (настоящего владельца этих данных). Это может привести для жертвы к испорченной кредитной истории, появлению ложных записей о криминальной деятельности, ошибочному аресту невиновных лиц. Кража личности имеет две категории: кража честного имени или захват счета. Кража честного имени означает, что вор использует украденную личную информацию для открытия новых счетов (новый банковский счет, чековый счет, кредитная карта, регистрации сотового телефона и т.п.). При захвате счета, злоумышленник использует персональную информацию, чтобы получить доступ к существующим счетам жертвы. Например, он может изменить адрес электронной почты, связанный со счетом для отправки выписок, и по-быстрому сделать с этого счета крупную покупку, прежде чем человек, личность которого была украдена, узнает о проблеме. Интернет существенно упростил использование украденной личной информации, поскольку многие операции могут осуществляться без личного присутствия.

13. Резюме

Управление доступом – это функция безопасности, которая обычно считается первой линией обороны при защите активов. Управление доступом используется для того, чтобы указывать, каким образом объекты используют доступ к субъектам, основная цель управления доступом заключается в защите объектов от несанкционированного доступа. Эти защитные меры могут носить административный, физический или технический характер и обеспечивать превентивные, детективные, сдерживающие (или устрашающие), восстанавливающие, компенсирующие, а также корректирующие возможности.

Управление доступом определяет, каким образом пользователи должны быть идентифицированы, аутентифицированы и авторизованы. Эти задачи решаются по-разному в различных моделях и технологиях управления доступом, компании должны определить, какие из этих моделей и технологий лучше всего соответствуют их бизнесу и потребностям в обеспечении безопасности.

Тест

Вопросы экзамена CISSP являются концептуальными, поэтому они сформулированы соответствующим образом. Задачей кандидата является выбор наилучшего из всех представленных вариантов ответа. Среди вариантов ответа может не быть идеального ответа на поставленный вопрос - кандидат должен выбрать лучший ответ из имеющихся вариантов.

1. Какое из следующих утверждений правильно описывает биометрические технологии?

- ☐ A. Они наименее дорогие, но обеспечивают наилучшую защиту
- ☐ B. Они очень дорогие, но обеспечивают небольшой уровень защиты
- ☐ C. Они наименее дорогие и обеспечивают небольшой уровень защиты
- ☐ D. Они очень дорогие и обеспечивают наилучшую защиту
2. Что является производной из парольной фразы и фактически используется для аутентификации в системе?
- ☐ A. Персональный пароль
- ☐ B. Виртуальный пароль
- ☐ C. Идентификатор пользователя
- ☐ D. Правильный пароль
3. Какое из следующих утверждений правильно описывает пароли?
- ☐ A. Они наименее дорогие, но обеспечивают наилучшую безопасность
- ☐ B. Они очень дорогие, но обеспечивают небольшую безопасность
- ☐ C. Они наименее дорогие и обеспечивают небольшую безопасность
- ☐ D. Они очень дорогие и обеспечивают наилучшую безопасность
4. В чем заключается смысл внедрения принципа разделения обязанностей?
- ☐ A. Никто не может самостоятельно выполнить все шаги критичной задачи
- ☐ B. Он порождает атмосферу заговора
- ☐ C. Он увеличивает зависимость от конкретных людей
- ☐ D. Он упрощает выполнение критичных задач
5. Что из перечисленного ниже не является логическим управлением доступом?
- ☐ A. Шифрование
- ☐ B. Сетевая архитектура
- ☐ C. Идентификационные бейджи
- ☐ D. Матрица контроля доступа
6. Модель управления доступом должна отдавать предпочтение _____ способам
- ☐ A. Детективным
- ☐ B. Восстанавливающим
- ☐ C. Корректирующим
- ☐ D. Превентивным
7. Какая политика управления доступом применяется, когда в среде используется недискреционная модель?
- ☐ A. На основе правил
- ☐ B. Ролевая
- ☐ C. На основе идентификации
- ☐ D. Мандатная
8. Как используется протокол запрос/ответ в случае применения токенов?
- ☐ A. Этот протокол не используется, применяется криптография
- ☐ B. Служба аутентификации генерирует запрос, а «умный» токен генерирует ответ на основе запроса
- ☐ C. Токен запрашивает у пользователя имя и пароль
- ☐ D. Токен запрашивает пароль пользователя у базы данных, в которой хранятся учетные данные
9. Какая модель управления доступом является управляемой пользователем?
- ☐ A. Недискреционная
- ☐ B. Мандатная
- ☐ C. На основе идентификации
- ☐ D. Дискреционная
10. Что обеспечивает наилучшую аутентификацию?
- ☐ A. Что человек знает
- ☐ B. Кем человек является
- ☐ C. Что человек имеет
- ☐ D. Что человек имеет и знает
11. Что из перечисленного ниже не является частью реализации аутентификации посредством Kerberos?
- ☐ A. Код аутентичности сообщения
- ☐ B. Служба предоставления билетов
- ☐ C. Служба аутентификации
- ☐ D. Пользователи, программы и службы

12. Какая модель реализует матрицы контроля доступа для управления взаимодействием субъектов с объектами?

- ☐ А. Мандатная
- ☐ В. Централизованная
- ☐ С. Децентрализованная
- ☐ D. Дискреционная

13. Что означает аутентификация?

- ☐ А. Регистрация пользователя
- ☐ В. Идентификация пользователя
- ☐ С. Проверка пользователя
- ☐ D. Авторизация пользователя

14. Если компания имеет большую текучесть кадров, какая структура управления доступом подойдет ей лучше всего?

- ☐ А. Ролевая
- ☐ В. Децентрализованная
- ☐ С. На основе правил
- ☐ D. Дискреционная

15. Для чего обычно используется пароль?

- ☐ А. Идентификация
- ☐ В. Регистрация
- ☐ С. Аутентификация
- ☐ D. Авторизация

16. Процесс взаимной аутентификации подразумевает, что _____.

- ☐ А. Пользователь аутентифицируется системой, а система аутентифицируется пользователем
- ☐ В. Пользователь аутентифицируется на двух системах одновременно
- ☐ С. Пользователь аутентифицируется сервером, а затем – процессом
- ☐ D. Пользователь аутентифицируется, получает билет, а затем с помощью билета аутентифицируется службой

17. Примером какого типа функций безопасности является просмотр журналов регистрации событий?

- ☐ А. Превентивный
- ☐ В. Детективный
- ☐ С. Сдерживающий (устрашающий)
- ☐ D. Корректирующий

18. При использовании дискреционного управления доступом, кто имеет полномочия предоставления прав доступа к данным?

- ☐ А. Пользователь
- ☐ В. Офицер безопасности
- ☐ С. Политика безопасности
- ☐ D. Владелец

19. Что может быть единой точкой отказа при использовании функциональности единого входа?

- ☐ А. Сервер аутентификации
- ☐ В. Рабочая станция пользователя
- ☐ С. Учетные данные
- ☐ D. RADIUS

20. Какую роль играет биометрия в управлении доступом?

- ☐ А. Авторизация
- ☐ В. Аутентичность
- ☐ С. Аутентификация
- ☐ D. Подотчетность

21. Что (или кто) определяет, какую модель управления доступом следует использовать компании – дискреционную, мандатную или недискреционную?

- ☐ А. Администратор
- ☐ В. Политика безопасности
- ☐ С. Культура
- ☐ D. Уровень безопасности

22. Какой тип атаки пытается перебрать все возможные варианты?

- ☐ А. По словарю
- ☐ В. Брутфорс

- ☐ C. Человек-по-середине
- ☐ D. Спуфинг

23. Что из перечисленного ниже правильно описывает понятие спуфинга?

- ☐ A. Прослушивание коммуникационного канала
- ☐ B. Работа посредством списка слов
- ☐ C. Захват сессии
- ☐ D. Выдавание себя за кого-то или что-то другое

24. Что из перечисленного ниже не является преимуществом централизованного администрирования управления доступом?

- ☐ A. Гибкость
- ☐ B. Стандартизация
- ☐ C. Высокий уровень безопасности
- ☐ D. Отсутствуют различия в интерпретации необходимого уровня безопасности

25. Что из перечисленного ниже лучше всего описывает то, что дает компаниям ролевое управление доступом в части снижения административных расходов?

- ☐ A. Позволяет принимать решения о том, кто может и кто не может иметь доступ к ресурсам тем людям, которые лучше всего знают эти ресурсы
- ☐ B. Обеспечивает централизованный подход к управлению доступом, что освобождает от обязательств руководителей подразделений
- ☐ C. Членство пользователей в ролях может быть легко отменено, новые пользователи включаются в соответствующие роли при назначении на работу
- ☐ D. Обеспечивает реализацию политик безопасности, стандартов и руководств на уровне всей компании

26. Что из перечисленного ниже лучше всего описывает каталоги (directories) и то, как они относятся к управлению идентификацией?

- ☐ A. Большинство является иерархическими и следует стандарту X.500
- ☐ B. Большинство имеет плоскую архитектуру и следует стандарту X.400
- ☐ C. Большинство пришло из LDAP
- ☐ D. Многие используют LDA

27. Что из приведенного ниже не является частью инициализации (provisioning) пользователей?

- ☐ A. Создание и блокировка (деактивация) пользовательских учетных записей
- ☐ B. Внедрение бизнес-процесса
- ☐ C. Поддержка и деактивация пользовательских объектов и атрибутов
- ☐ D. Делегирование администрирования пользователей

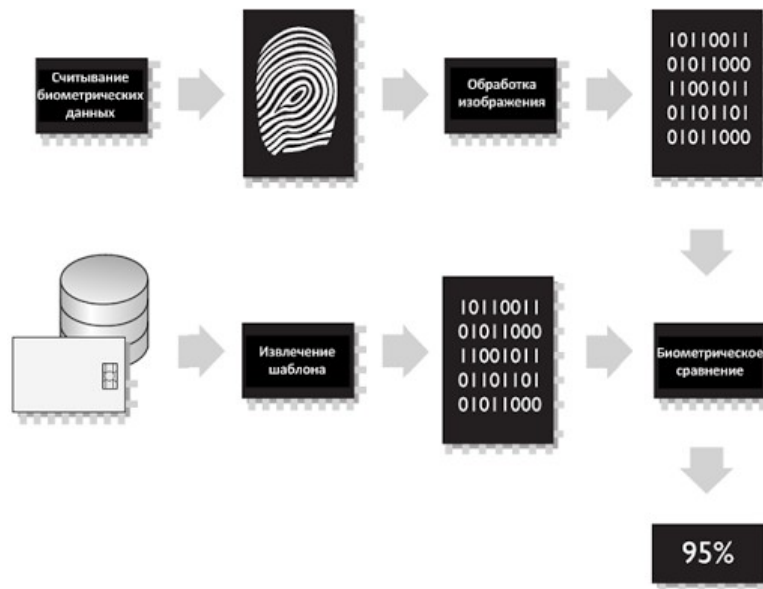
28. Что из перечисленного ниже является технологией, позволяющей пользователю помнить только один пароль?

- ☐ A. Генерация паролей
- ☐ B. Словари паролей
- ☐ C. Rainbow-таблица паролей
- ☐ D. Синхронизация паролей

29. Что из перечисленного ниже не является системой выявления вторжений (IDS) на основе аномалий?

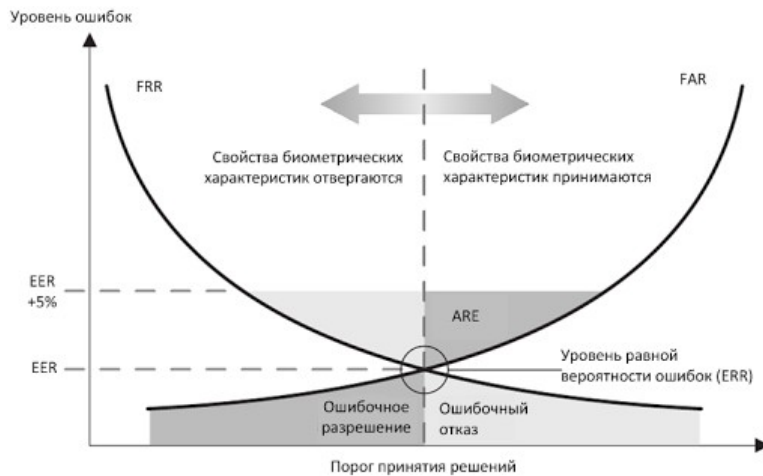
- ☐ A. На основе статистических аномалий
- ☐ B. На основе аномалий протоколов
- ☐ C. На основе аномалий по времени
- ☐ D. На основе аномалий трафика

30. Что из перечисленного ниже имеет отношение к этому рисунку:



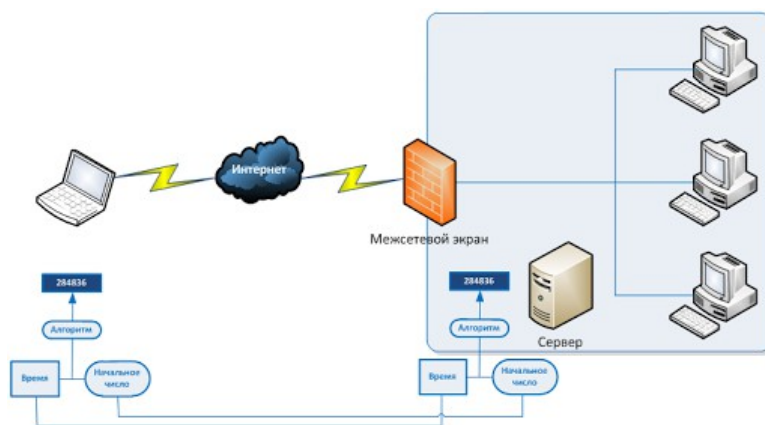
- ☐ А. Уровень пересечения вероятности ошибок (Crossover error rate)
☐ В. Проверка личности
☐ С. Уровень авторизации
☐ D. Уровень ошибок аутентификации

31. Объяснение какой из приведенных ниже концепций показано на рисунке:



- ☐ А. Уровень пересечения вероятности ошибок (Crossover error rate)
☐ В. Ошибки третьего рода
☐ С. FAR (уровень ошибочных разрешений) равен FRR (уровень ошибочных отказов) в системах, которые имеют высокий уровень пересечения вероятности ошибок (Crossover error rate)
☐ D. Биометрия является общепринятой технологией

32. Работу чего из перечисленного ниже иллюстрирует этот рисунок:



- ☐ A. Rainbow-таблицы
- ☐ B. Атака по словарю
- ☐ C. Одноразовые пароли
- ☐ D. Строгая аутентификация

Домен 03. Архитектура и модель безопасности.

Компьютерная и информационная безопасность охватывает многие области работы компании. Каждая область имеет свои уязвимости безопасности, и (надеюсь) соответствующие контрмеры, которые повышают уровень безопасности и обеспечивают наилучшую защиту. Непонимание различных областей и уровней безопасности сетевых устройств, операционных систем, аппаратных средств, протоколов и приложений может стать причиной уязвимостей, которые могут оказать влияние на всю среду компании в целом.

Двумя фундаментальными понятиями в компьютерной и информационной безопасности являются политика безопасности и модель безопасности. **Политика безопасности** – это заявление, в котором описывается, каким образом сущности взаимодействуют друг с другом, какие операции могут выполнять различные сущности, какой уровень защиты является необходимым для системы или программного продукта, а также какие действия следует предпринять, если эти требования будут нарушены. Политика безопасности описывает ожидания, которым должны удовлетворять аппаратные средства и программное обеспечение. **Модель безопасности** описывает требования, необходимые для реализации и надлежащей поддержки политики безопасности. Если политика безопасности требует, чтобы все пользователи были идентифицированы, аутентифицированы и авторизованы перед тем, как им будет предоставлен доступ к сетевым ресурсам, модель безопасности может содержать матрицу контроля доступа, построенную таким образом, чтобы удовлетворять требованиям политики безопасности. Если политика безопасности требует, чтобы никто на нижнем уровне безопасности не имел возможности просмотра или изменения информации на более высоком уровне безопасности, поддерживающая модель безопасности будет описывать необходимую логику и правила, которые должны быть предприняты для обеспечения того, чтобы ни при каких обстоятельствах не мог произойти несанкционированный доступ субъекта на нижнем уровне к объекту на более высоком уровне безопасности. Модель безопасности более глубоко объясняет требования к разработке операционной системы компьютера, реализация которых необходима для надлежащей поддержки конкретной политики безопасности.

ПРИМЕЧАНИЕ. Отдельные системы и устройства могут иметь собственные политики безопасности. Они не являются организационной политикой безопасности, содержащей директивы руководства. Политики безопасности систем и модели, которые они используют, должны обеспечивать внедрение организационной политики безопасности более высокого уровня. Системная политика указывает уровень безопасности, который должны обеспечивать отдельные устройства или операционная система.

Термин *компьютерная безопасность* может быть достаточно «скользким», поскольку часто он означает разные вещи для разных людей. Можно защищать многие системные аспекты и реализовывать безопасность на различных уровнях и в различной степени, но как уже говорилось в предыдущих доменах, информационная безопасность состоит из следующих основных атрибутов:

- **Доступность.** Предотвращение потери данных и ресурсов, либо доступа к данным и ресурсам.
- **Целостность.** Предотвращение несанкционированного изменения данных и ресурсов.
- **Конфиденциальность.** Предотвращение несанкционированного разглашения данных и ресурсов.

Эти основные атрибуты безопасности делятся на более мелкие, такие как аутентичность, подотчетность, неотказуемость и надежность. Как компания может узнать, какие из этих атрибутов ей нужны? В какой степени? Обеспечивают ли в действительности используемые компанией операционные системы и приложения необходимую защиту? Эти вопросы являются крайне сложными, если рассматривать их достаточно глубоко. Если компания

беспокоится о безопасности своих данных, ей недостаточно просто шифровать свою электронную почту, которая передается по открытым Интернет-каналам. Ей также нужно озаботиться защитой данных в своих базах данных; безопасностью веб-ферм, напрямую подключенных к Интернету; целостностью вводимых в бизнес-приложения данных; совместным использованием внутренними пользователями материалов, составляющих коммерческую тайну; потенциальной возможностью выведения из строя серверов внешними злоумышленниками; потенциальным воздействием компьютерных вирусов на производительность работы; внутренней целостностью и согласованностью данных в хранилище данных; а также многим другим.

Эти вопросы не только влияют на производительность и рентабельность компании, но могут также создать правовые проблемы и проблемы с неисполнением обязательств в отношении защиты данных. Компании, а также их руководители, могут быть привлечены к ответственности, если произойдут какие-либо проблемы в рамках перечисленных выше вопросов. Поэтому эти (как минимум) вопросы должны быть очень важны для компаний, руководители компаний должны быть уверены, что обеспечен необходимый им уровень безопасности, что защита реально обеспечивается приобретаемыми ими продуктами.

Многие из таких вопросов безопасности должны быть продуманы «до» и «во время» этапов проектирования и создания архитектуры продукта. Безопасность обеспечивается значительно лучше в продуктах, в которых она была спроектирована и встроена в самую основу приложения или операционной системы, а не добавлена в конце проекта, как запоздалая идея. После интеграции безопасности в проект, она должна быть тщательно спланирована, реализована, протестирована, проаудирована, оценена, сертифицирована и аккредитована. Безопасности, которую обеспечивает продукт, должен быть присвоен рейтинг в части доступности, целостности и конфиденциальности. Клиенты затем используют эти рейтинги, чтобы понять, обеспечивают ли конкретные продукты необходимый им уровень безопасности. Это большая работа со множеством участников, имеющих различные обязанности.

В этом домене мы пройдем от самых первых шагов, которые необходимо предпринять еще перед разработкой операционной системы, до заключительных этапов, связанных с оценкой и присвоением рейтингов. Однако прежде чем мы углубимся в эти концепции, важно понять, как работают основные компоненты компьютерной системы, составляющие архитектуру любого компьютера.

1. Архитектура компьютера

Архитектура компьютера включает в себя все элементы компьютерной системы, необходимые для ее функционирования, включая операционную систему, память, логические схемы, устройства хранения информации, устройства ввода/вывода, шины, компоненты безопасности, сетевые компоненты. Взаимосвязи и внутренняя работа всех этих частей могут быть весьма сложными, но еще сложнее заставить все это работать вместе безопасным образом. К счастью умные люди уже сделали это, и теперь нам предстоит узнать, как они это сделали и почему.

Чем лучше вы поймете, как эти различные компоненты работают и обрабатывают данные, тем лучше вы поймете, как в действительности появляется уязвимость, и как работают контрмеры, препятствующие появлению уязвимостей, их выявлению и использованию.

1.1. Центральный процессор

Центральный процессор (CPU – central processing unit) – это «мозг» компьютера. Он извлекает из памяти команды и выполняет их. Хотя процессор является частью аппаратного обеспечения, он имеет свой собственный набор команд (поддерживаемый операционной системой), необходимый ему для выполнения своих задач. Каждый процессор имеет определенную архитектуру и набор команд. Операционная система должна быть

спроектирована под конкретную архитектуру процессора. Именно поэтому операционная система, разработанная для процессора Pentium, не будет работать на процессоре SPARC.

ПРИМЕЧАНИЕ. SPARC (Scalable Processor Architecture) – это разновидность RISC-микросхем (Reduced Instruction Set Computing), разработанных Sun Microsystems. SunOS, Solaris и некоторые другие Unix-подобные операционные системы разработаны именно для такого вида процессоров.

Микросхемы в процессоре занимают всего несколько квадратных дюймов, но содержат десятки и сотни миллионов транзисторов. Все операции в процессоре выполняются с помощью электрических сигналов с различными напряжениями в различных комбинациях, каждый транзистор сохраняет определенное напряжение, переводимое компьютером в нули и единицы. Процессор имеет **регистры**, являющиеся сверхбыстрой памятью внутри процессора, предназначенной для временного хранения информации (например, указателя на адрес в памяти со следующей командой для выполнения процессором, указателя на текущую позицию в стеке, информации о состоянии обрабатываемых данных и т.п.). Доступ к памяти для получения очередной команды или данных – это значительно более медленный процесс, чем к обращению к регистру. Поэтому, когда процессор завершает одну задачу, он берет информацию о следующей задаче именно из регистров.

В действительности часть команд выполняется **арифметико-логическим устройством** (ALU – arithmetic logic unit). ALU выполняет математические функции и логические операции над данными. ALU является «мозгом» процессора, а процессор – «мозгом» компьютера.

Программное обеспечение хранит свои команды и данные в памяти. Когда необходимо произвести определенное действие над данными, адреса команд и данных поступают в соответствующие регистры процессора, как показано на рисунке 3-1. Когда устройство управления говорит процессору, что он может приступить к работе, адреса команд и данных поступают в процессор для реальной обработки, математических расчетов и управления данными. Результаты отправляются обратно в память запрашивающего процесса.

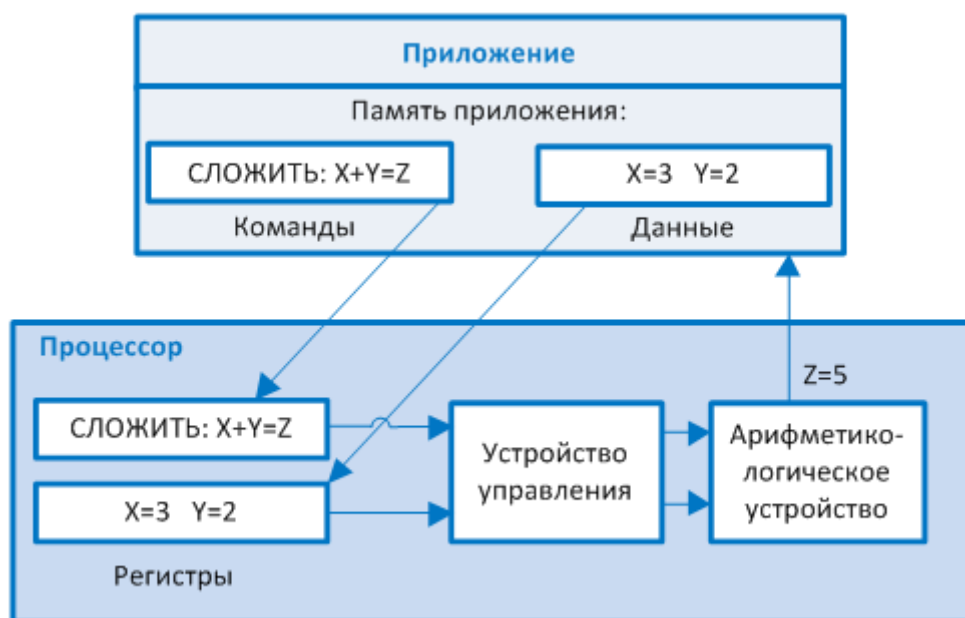


Рисунок 3-1. Адреса команд и данных передаются процессору для обработки

Операционная система или приложение являются просто множеством строк с командами. Эти команды содержат пустые переменные, которые заполняются реальными данными во время выполнения программы. Предположим, что вы запускаете программу «Калькулятор». В действительности, эта программа является просто набором команд, которые позволяют выполнять сложение, вычитание, деление и другие математические функции над введенными данными. Например, вы вводите в 3+5. При этом 3 и 5 являются значениями

данных. После нажатия кнопки =, «Калькулятор» говорит процессору, что нужно применить команду сложения к двум значениям данных – 3 и 5. ALU выполняет эту команду и возвращает результат 8 запрашивающей программе, после чего вы увидите значение 8 в соответствующем поле «Калькулятора». Пользователю кажется, что программа «Калькулятор» делает все это сама, но она не в состоянии сделать ничего без помощи процессора и других компонентов системы.

Устройство управления (control unit) управляет и синхронизирует систему в процессе выполнения кода операционной системы и различных приложений. Устройство управления загружает код, интерпретирует его и следит за выполнением различных наборов команд. Оно определяет, какие команды приложения передаются на обработку, с каким приоритетом, и на какое время. Управление процессом выполнения команд позволяет приложениям обрабатывать данные. При этом само устройство управления не обрабатывает данные. Оно работает подобно инспектору ГИБДД, приказывая потоку транспорта остановиться или продолжить движение. Процессорное время делится на отдельные кванты и предоставляется процессам. Именно это деление процессорного времени заставляет нас думать, что процессор выполняет различные команды одновременно, хотя на самом деле он выполняет их последовательно (по одной).

Процессор имеет несколько различных типов регистров, содержащих информацию о командах и данных, которые должны быть обработаны. **Регистры общего назначения** используются для хранения переменных и временных результатов. **Специальные регистры** (выделенные регистры) содержат такую информацию, как счетчик команд (program counter), указатель стека (stack pointer) и слово состояния программы (PSW – program status word). Регистр счетчика команд содержит адрес следующей команды для загрузки. После запуска команды на выполнение счетчик команд обновляется, в него записывается адрес памяти, содержащий следующую команду для выполнения. Это похоже на взаимоотношения на начальника и секретаря. Секретарь ведет календарь своего начальника и указывает ему на задачи, которые ему нужно выполнять. Это позволяет начальнику просто сосредоточиться на работе, и не отвлекаться на различные «фоновые» процессы.

Прежде чем мы перейдем к указателю стека, мы должны сначала понять, что такое стек. Каждый процесс имеет свой собственный **стек** (stack), являющийся структурой данных в памяти, которую процесс может читать и в которую он может записывать данные способом LIFO (последним пришел - первым ушел). Предположим, нам с вами нужно общаться через стек (представим его в виде стопки бумаги). Я пишу на отдельных листах бумаги все, что я должен сказать вам, и складываю все листы в одну стопку. На первом листе я пишу, как вы можете ответить мне, когда закончите работу (это называется указатель возврата (return pointer)). На следующем листе я пишу некоторые инструкции для вас, которые вы должны выполнить, и кладу этот лист поверх предыдущего. На следующем листе я пишу данные, которые вы должны использовать при выполнении этих инструкций, и также кладу его сверху. Когда вы приступаете к работе, вы берете первый лист из стопки и выполняете соответствующее задание. Затем вы берете следующий лист и так далее, пока не дойдете до последнего листа в этой стопке, содержащего мой указатель возврата, указывающий на место в моем адресном пространстве, куда вам нужно отправить результаты всех выполненных по моим заданиям работ. Таким же образом взаимодействуют процессы между собой и с центральным процессором. Процессору нужно отслеживать текущее местоположение в стеке, для чего и служит **указатель стека**. После получения первого элемента стека, указатель стека перемещается вниз, чтобы сообщить процессору, где в стеке находится следующий фрагмент данных.

Слово состояния программы хранит различные биты состояния. Один из битов указывает, работает ли процессор в **реальном режиме** (user mode), или в **защищенном режиме** (privileged mode) (также называемом **режимом ядра** (kernel или supervisor mode)). Суть настоящего домена заключается в том, чтобы объяснить вам, как операционная система

обеспечивает собственную защиту. Чтобы обеспечить стабильную и безопасную среду, операционная система должна защищать себя от приложений, утилит и действий пользователей. Один из таких механизмов защиты реализуется посредством использования различных режимов работы. Когда приложение передает процессору для выполнения свои команды, процессор выполняет их в реальном режиме. Этот режим имеет низкий уровень привилегий, делающий недоступными для запрашивающего приложения многие из команд и функций процессора. Причиной для такой осторожности является то, что разработчики операционной системы не знают, кто разработал приложение, и как оно будет работать, поэтому процессор работает в непривилегированном режиме, когда выполняет такие команды. По аналогии, если вы ждете гостей, которые придут со своим двухлетним ребенком, вы уберете все хрупкие предметы, до которых ребенок в таком возрасте сможет дотянуться. Вы не знаете, что этот малыш будет делать, но предполагаете, что он может что-нибудь сломать. Точно также, операционная система и процессор не знают, что будет пытаться сделать приложение, поэтому его код выполняется с низким уровнем привилегий.

Если соответствующий бит в PSW указывает, что команды должны выполняться в защищенном режиме, это означает, что запрос сделал доверенный процесс (процесс самой операционной системы), который имеет доступ к функциональности, недоступной в реальном режиме. Примером может служить взаимодействие операционной системы с периферийным устройством. Это привилегированные действия, которые обычное приложение не может выполнить самостоятельно.

Адреса памяти, указывающие на местонахождение команд и данных для обработки, хранятся в регистрах, пока они не понадобятся процессору. Процессор подключен к *адресной шине*, которая физически подключена к микросхемам оперативной памяти и отдельным устройствам ввода/вывода. Память разделена на сегменты, которые имеют собственные адреса. Устройствам ввода/вывода (CD-ROM, USB-устройства, жесткий диск, дисковод и т.д.) также выделены отдельные уникальные адреса. Если процессору нужен доступ к определенным данным из памяти или из устройства ввода/вывода, он отправляет по адресной шине адрес, где находятся необходимые данные. Логическая схема, связанная с памятью или устройством ввода/вывода, распознает полученный от процессора адрес, и дает команду памяти или устройству ввода/вывода прочитать запрошенные данные, а затем направляет их процессору по *шине данных*. Таким образом, адресная шина используется процессором для указания местоположения команд или данных, подлежащих обработке, а память или устройство ввода/вывода отвечает, посылая запрошенные данные через шину данных. Этот процесс показан на рисунке 3-2.

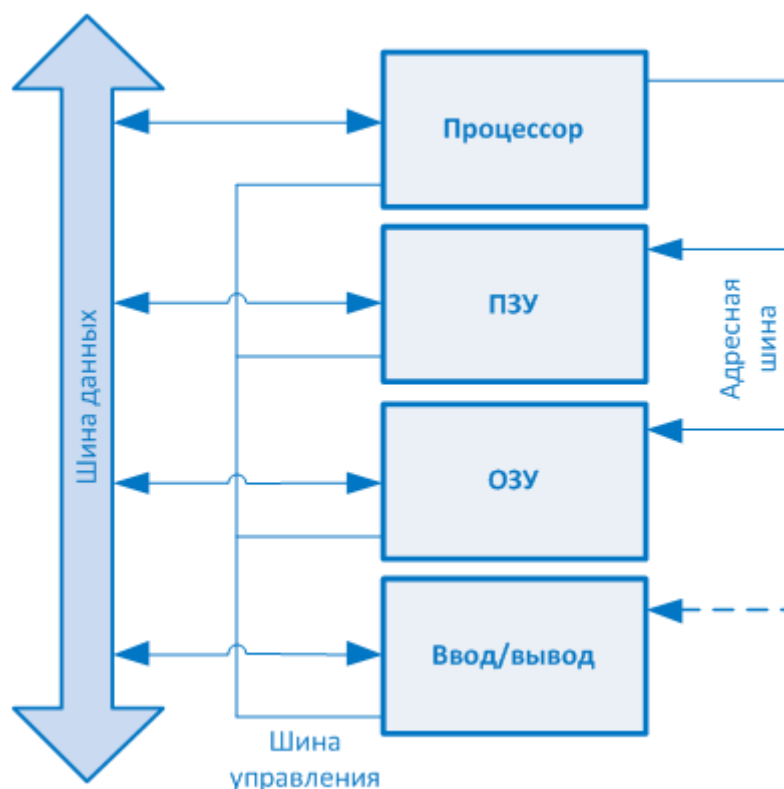


Рисунок 3-2. Адресная шина и шина данных разделены и выполняют различные функции

Когда процессор заканчивает вычисления, он должен вернуть результат в память запрашивающей программы. Для этого процессор посылает соответствующий адрес по адресной шине, а результаты – по шине данных вместе с командой записи. Таким образом, эти новые данные записываются в память запрашивающей программы.

Адресная шина и шина данных могут иметь ширину 8, 16, 32 или 64 бит. Большинство систем сегодня используют 32-битную адресную шину, означающую, что система может иметь достаточно большое адресное пространство (232). Системы могут также иметь 32-битную шину данных, означающую, что система может перемещать данные параллельно туда и обратно между памятью, устройствами ввода/вывода и процессором, порциями по 32 бита.

1.2. Многопроцессорная обработка

В некоторых компьютерах для повышения производительности установлено более одного процессора. Для эффективной работы операционной системы с несколькими процессорами, она должна быть специально разработана для таких систем. При этом компьютер может быть настроен на работу в симметричном или ассиметричном режиме. В **симметричном режиме** процессоры берут новые задания по мере необходимости, как показано на рисунке 3-3 с процессорами 1 и 2. Это похоже на среду с балансировкой нагрузки. Когда процессу нужно выполнить некоторые свои команды, планировщик определяет, какой процессор готов для дальнейшей работы и отправляет эти команды ему. Если процессор будет выделен специально для конкретной задачи или приложения, все другие программы будут выполняться на других процессорах. Например, на рисунке 3-3 процессор 4 выделен для одного приложения и его потоков, в то время как процессор 3 используется операционной системой. Когда процессор выделен, как в этом примере, система работает в **ассиметричном режиме**. Обычно это означает, что на компьютере работает несколько чувствительных ко времени выполнения приложений, которые нуждаются в собственном персональном процессоре. Таким образом, системный планировщик отправляет команды от такого приложения процессору 4, а все остальные команды (от операционной системы и других приложений) процессору 3. Эти различия показаны на Рисунке 3-3.

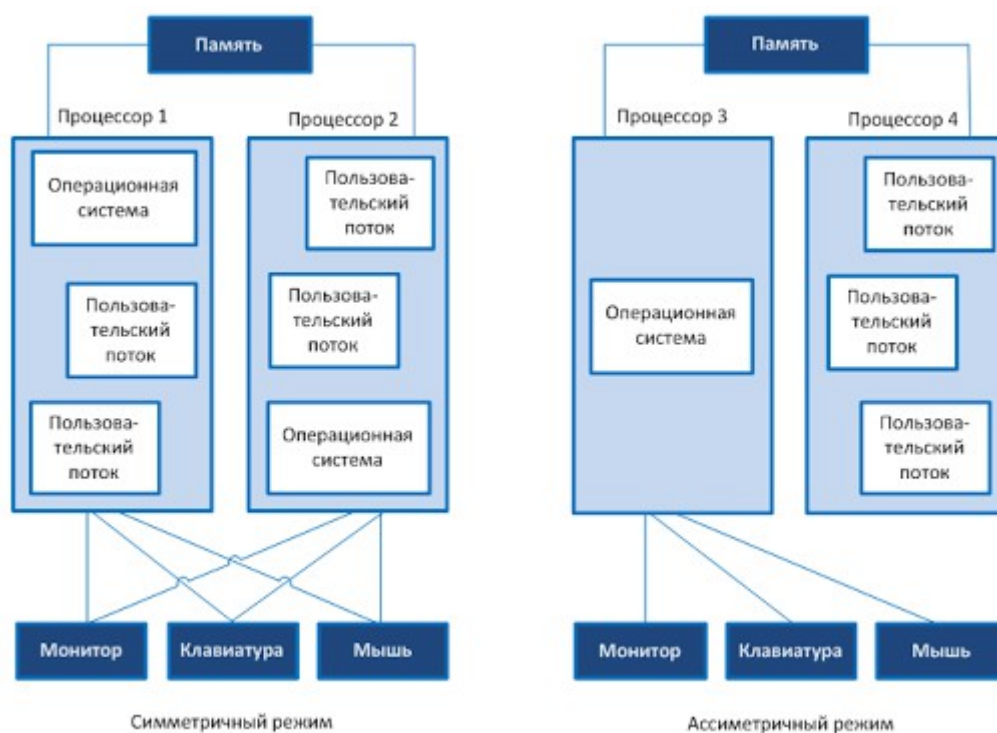


Рисунок 3-3. Симметричный и ассиметричный режимы

Эволюция процессоров. В следующей таблице приводятся некоторые характеристики процессоров, использовавшихся на протяжении последних лет.

Название	Год	Транзисторов	Микрон	Частота	Разрядность	MIPS
8080	1974	6 000	6	2 МГц	8 бит	0,64
80286	1982	134 000	1,5	6 МГц	16 бит	1
Pentium	1993	3 100 000	0,8	60 МГц	32 бита, Шина – 64 бита	100
Pentium 4	2000	42 000 000	0,18	1,5 ГГц	32 бита, Шина – 64 бита	1 700

Следующий список поясняет единицы измерения, используемые в приведенной выше таблице:

- **Микрон** – Наименьшая толщина проводника в микросхеме процессора в микронах (для сравнения, толщина человеческого волоса составляет 100 микрон).
- **Частота** – Тактовая частота, скорость, с которой процессор может выполнять команды. Для управления скоростью выполнения операций применяется внутренний таймер, который делит время на такты. Например, если система работает на частоте 100 МГц, это означает, что происходит 100 миллионов тактов в секунду.
- **Разрядность** – Указывает объем данных, который ALU может принять и обработать; пометка «шина – 64 бита» указывает на размер шины данных. Таким образом, современные системы передают по 64 бита данных, но ALU работает только 32 битами данных.
- **MIPS** – Миллионы операций в секунду, что является одним из основных показателей скорости работы процессора (однако на это оказывают влияние и другие факторы, такие как тактовая частота).

1.3. Архитектура операционной системы

Операционная система предоставляет среду для работы приложений и пользователей. Операционная система состоит из различных слоев и модулей, которые отвечают за управление оборудованием, памятью, операциями ввода/вывода, процессами, файловой системой, а также предоставляют системные сервисы. Далее мы вкратце рассмотрим каждую из этих функций, относящихся к любой операционной системе.

Управление процессами

Операционные системы, утилиты и приложения в действительности являются просто множеством строк команд. Эти статические строки кода оживают после инициализации программы и размещения в памяти. Приложения работают как отдельные модули, называемые процессами, операционная система также имеет множество различных процессов, выполняющих различные функции. **Процесс** – это набор команд, запущенный на выполнение. Программа не является процессом, пока она не загружена в память и не активирована операционной системой. При создании процесса операционная система выделяет ему ресурсы, такие как сегменты памяти, кванты процессорного времени, доступ к интерфейсам API (application programming interface), файлам и т. п.

Операционная система имеет целый ряд процессов, которые обеспечивают и поддерживают среду для работы пользователей и приложений. Например, некоторые процессы обеспечивают такие функции, как отображение данных на экране, буферизацию заданий для печати, сохранение данных во временных файлах и т.д. Современные операционные системы являются многопрограммными, т.е. они способны одновременно исполнять более одной программы (или процесса). Именно это позволяет одновременно запускать антивирусное программное обеспечение, текстовый редактор, персональный межсетевой экран и клиент электронной почты. Каждое из этих приложений выполняется в виде одного или более процессов.

ПРИМЕЧАНИЕ. Многие современные операционные системы обеспечивают многопрограммную работу и многозадачность. Многопрограммная работа означает просто, что более чем одно приложение может быть одновременно загружено в память. В действительности многопрограммность была заменена многозадачностью, означающей, что не только несколько приложений может быть одновременно загружено в память, но при этом операционная система может одновременно обрабатывать запросы от различных приложений.

Ранние операционные системы впустую расходовали самый драгоценный ресурс компьютера – процессорное время. Например, когда текстовый редактор запрашивал данные из файла на диске, процессор отправлял запрос дисководу, а затем ждал, пока дисковод инициализируется, найдет нужный трек и сектор, и, наконец, передаст процессору запрошенные данные через шину данных для обработки. Чтобы избежать таких расходов процессорного времени, была разработана многозадачность, позволившая нескольким программам работать одновременно. Вместо того чтобы простаивать, ожидая результат работы одного процесса, процессор может выполнять команды других процессов, в целом повышая скорость работы процессов.

Аналогично, если вы (процессор) положили хлеб в тостер и просто стоите и ждете, когда тостер закончит свою работу (процесс), вы теряете время. Однако если вы, пока готовится хлеб, покормили кошку, сделали кофе и придумали, как достичь мира во всем мире, то время было потрачено гораздо более продуктивно.

В ранних версиях операционных систем (Windows 3.1, Macintosh) была реализована **кооперативная многозадачность** (cooperative multitasking), когда процессы сами освобождали ресурсы компьютера, по своему усмотрению. В более современных версиях (Windows 2000, XP, Unix-системах и т.д.) реализована **вытесняющая многозадачность** (preemptive multitasking), при использовании которой операционная система сама управляет использованием ресурсов процессами. Кооперативная многозадачность не может обеспечить стабильную среду, т.к. если программист не написал (или неправильно написал) код, надлежащим образом освобождающий ресурс после выполнения своего приложения, этот ресурс может оказаться заблокированным на неопределенное время и недоступным для других процессов. При использовании вытесняющей многозадачности операционная система управляет тем, сколько времени процесс может использовать ресурс. Система может приостановить процесс, использующий процессор и позволить другим процессам получить доступ к нему с помощью *разделения времени* (time sharing). В операционных системах с

кооперативной многозадачностью, процессы имели слишком большой контроль над ресурсами, и если приложение зависало, это, как правило, затрагивало все остальные приложения, а иногда и саму операционную систему. В операционных системах с вытесняющей многозадачностью одно приложение не может также легко оказать негативное влияние на другое приложение.

Различные операционные системы используют различные модели процессов. Например, системы Unix и Linux позволяют своим процессам создавать новые дочерние процессы, что называется *ветвлением* (forking). Допустим, вы работаете в оболочке системы Linux. Эта оболочка является командным интерпретатором, а также интерфейсом, который позволяет пользователю взаимодействовать с операционной системой. Оболочка выполняется как процесс. Если вы введете в этой оболочке команду `cat file1 file2 | grep stuff`, вы скажете операционной системе, что нужно объединить (cat) два файла, а затем найти (grep) строки, которые имеют значение stuff. Когда вы нажмете клавишу ENTER, оболочка породит два дочерних процесса – один для команды cat и один для команды grep. Каждый из этих дочерних процессов получит характеристики родительского процесса, но будет иметь свое собственное пространство памяти, стек и значение счетчика команд.

Процесс может находиться в **состоянии выполнения** (running state – процессор выполняет команды процесса), в **состоянии готовности** (ready state – ожидание передачи команд процессору) или в **заблокированном состоянии** (blocked state – ожидание входящих данных). Эти различные состояния показаны на рисунке 3-4. Когда процесс заблокирован, он ждет, когда ему отправят некоторые данные. В предыдущем примере, после ввода команды `cat file1 file2 | grep stuff`, процесс grep не может выполнять свои функции поиска, пока процесс cat не выполнит объединение этих двух файлов. Процесс grep переведет себя в **спящий режим** и будет находиться в заблокированном состоянии, пока не выполнится процесс cat и не пошлет процессу grep входные данные, которые он должен будет обработать.



Рисунок 3-4. Процесс проходит через различные состояния

ПРИМЕЧАНИЕ. Не все операционные системы работают с процессами также, как системы Unix и Linux. Например, системы Windows не порождают новых дочерних процессов, вместо этого они создают новые потоки, которые работают в том же контексте, что и родительский процесс.

Операционная система отвечает за создание новых процессов, выделение каждому из них ресурсов, синхронизацию их взаимодействия и контроль, что не происходит ничего опасного. Операционная система хранит специальную **таблицу процессов** (process table), в которой содержится информация о каждом процессе. В этой таблице хранятся связанные с процессами параметры, например, состояние процесса, указатель стека, программный счетчик, распределение памяти, состояние открытых файлов и т.д. Вся эта информация нужна операционной системе для того, чтобы процессор загружал ее в свои регистры, когда ему необходимо взаимодействовать, например, с процессом 1. Когда закончится квант процессорного времени, выделенный процессу 1, вся текущая информация о состоянии процесса 1 сохранится в таблице процессов, чтобы, когда этот процесс снова получит квант времени, все эти сведения можно было загрузить обратно в регистры процессора. При этом

из таблицы процессов в регистры процессора будет загружена информация процесса 2, а когда закончится и его квант процессорного времени, информация из регистров снова будет записана в таблицу процессов. Эти шаги показаны на Рисунке 3-5.



Рисунок 3-5. Таблица процессов содержит данные о состоянии процессов, которые необходимы процессору

Но как процесс узнает, когда он может взаимодействовать с процессором? Для этого используются **прерывания** (interrupt). Операционная система обманывает нас и приложения, заставляя думать, что процессор одновременно выполняет все задачи (операционную систему, приложения, операции с памятью, ввод/вывод и действия пользователей). Но в действительности это невозможно. Большинство процессоров могут выполнять только одну операцию за раз, поэтому система имеет аппаратные и программные прерывания. Когда устройству необходимо взаимодействовать с процессором, оно должно дождаться своего прерывания, которое его вызовет. То же самое происходит и в программном обеспечении. Каждый процесс имеет прерывание, связанное с ним. Это как присваивание номеров в отделе обслуживания клиентов в магазине – вы не можете пойти к прилавку, пока ваш номер не был назван.

Когда процесс взаимодействует с процессором и происходит прерывание (еще один процесс запрашивает доступ к процессору), информация о текущем процессе сохраняется в таблице процессов, и следующий процесс получает свое время для взаимодействия с процессором.

ПРИМЕЧАНИЕ. Некоторые критические процессы не могут допустить прерывание своего выполнения другим процессом. Операционная система обеспечивает установку приоритетов для различных процессов. Когда одному процессу нужно прервать другой процесс, операционная система сравнивает уровни приоритета обоих процессов, чтобы определить, следует ли разрешать такое прерывание.

Есть две категории прерываний: маскируемые (maskable) и немаскируемые (non-maskable). **Маскируемые** прерывания связаны с событиями, которые не могут быть очень важны, и программист может указать, что даже если происходит маскируемое прерывание, программа не прерывает своей работы. Таким образом, маскируемые прерывания могут игнорироваться. **Немаскируемые** прерывания не могут быть проигнорированы приложением ни при каких обстоятельствах, поскольку события, которые вызывают немаскируемые прерывания, имеют

критическое значение. Например, кнопке RESET может быть назначено немаскируемое прерывание, чтобы при нажатии этой кнопки процессор сразу выполнял соответствующие команды.

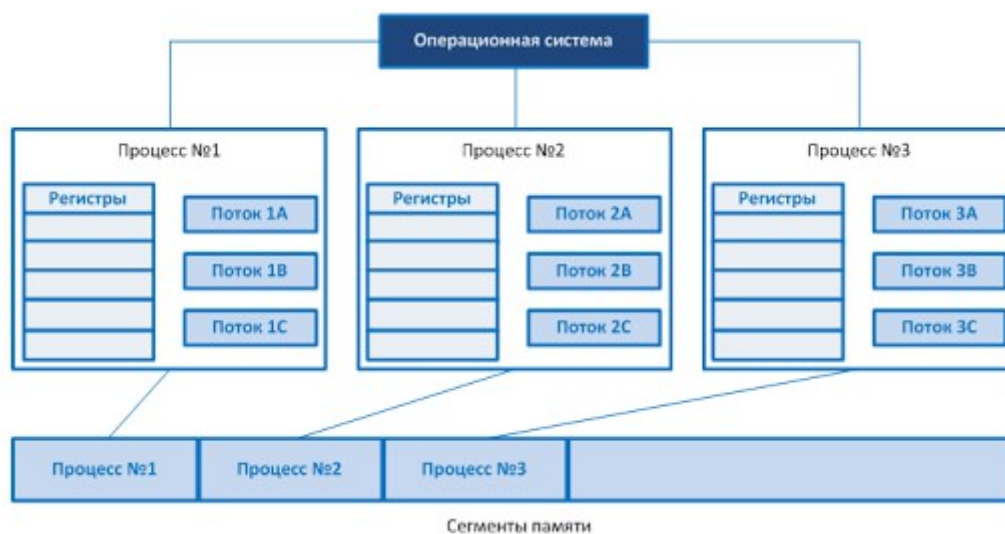
По аналогии, начальник может сказать своему секретарю, что он не будет отвечать ни на какие звонки, разве что будет звонить сам Президент. Это означает, что звонки всех остальных людей будут игнорироваться (маскируемые прерывания), но звонок Президента не будет проигнорирован (немаскируемое прерывание), поскольку этот начальник обязан ответить на звонок Президента.

Сторожевой таймер (watchdog timer) является примером критического процесса, который всегда должен выполнять свою работу. Этот процесс выполнит "горячую" перезагрузку системы, если она зависнет и не сможет восстановиться. Например, если возникают проблемы с управлением памятью и операционная система зависает, сторожевой таймер перезагрузит систему. Этот механизм обеспечивает более стабильную среду.

Управление потоками

Как уже говорилось ранее, процесс представляет собой программу в памяти. Точнее процесс является командами программы и всеми ресурсами, выделенными ему операционной системой. Значительно проще сгруппировать все эти команды и ресурсы вместе и контролировать их как единое целое (процесс). Когда процессу нужно отправить команды на выполнение процессору, он создает поток. **Поток** (thread) состоит из собственного набора команд и данных, которые должны быть обработаны процессором.

Большинство приложений имеют несколько различных функций. Текстовые редакторы могут открывать файлы, сохранять файлы, запускать другие программы (например, клиент электронной почты), а также печатать документы. Каждая из этих функций требует потока (набора команд), которые будут создаваться динамически. Так, например, если Том хочет распечатать свой документ, процесс текстового редактора создает поток, который содержит команды, непосредственно обеспечивающие печать документа (устанавливающие, в частности, шрифт, цвет, и т.д.). Если он решает отправить свой документ по электронной почте, создается другой поток, который дает команду клиенту электронной почты на открытие и отправку файла. Потоки создаются и уничтожаются динамически по мере необходимости. Когда Том напечатал свой документ, поток, который был создан для выполнения этой функции, уничтожается.



Используя потоки, программа может выполнять несколько задач одновременно (например, выводить изображение на экран, взаимодействовать с другими программами, отправлять документ на печать). Приложение, использующее эту возможность, называется **многопоточным** (multithreaded) приложением.

ПРИМЕЧАНИЕ. Каждый поток имеет общие ресурсы с процессом, который его создал. Все потоки процесса работают в том же адресном пространстве, что и сам процесс, имеют доступ к тем же файлам и системным ресурсам.

Определения. Концепции работы компьютерных операционных систем могут быть очень сложными. Убедитесь, что вы понимаете следующие основные определения:

- **Многопрограммность** – операционная система может одновременно загрузить в память более одной программы.
- **Многозадачность** – операционная система может одновременно обрабатывать запросы от нескольких различных процессов, загруженных в память.
- **Многопоточность** – приложение может запускать несколько потоков одновременно.
- **Многопроцессорность** – компьютер имеет более одного процессора.

Диспетчеризация процессов

Диспетчеризация (scheduling) и синхронизация (synchronizing) различных процессов и их действий – это часть управления процессами, выполняемого операционной системой. При разработке операционной системы, необходимо определить порядок (политику) диспетчеризации процессов. Политика диспетчеризации предназначена для управления взаимодействием процессов друг с другом. Различные операционные системы могут использовать различные варианты диспетчеризации (по сути, алгоритмы управления разделением процессорного времени). Как было сказано ранее, различные процессы имеют различные уровни приоритета (прерывания), которые учитываются при распределении процессорного времени. Операционная система создает и удаляет процессы по мере необходимости, меняет их состояние (готов, заблокирован, выполняется). Она также контролирует взаимные блокировки (deadlock) процессов, пытающихся использовать одни и те же ресурсы.

Когда один процесс делает запрос на доступ к ресурсу (памяти, принтеру, дисковому пространству и т.д.), операционная система создает определенные структуры данных и необходимые для выполнения этих действий процессы. Как только действия выполнены (напечатан документ, данные сохранены в файл или, наоборот, считаны с диска), процесс должен уничтожить эти структуры и освободить ресурсы, чтобы они стали доступны для других процессов. Если это не произойдет должным образом, может возникнуть **взаимная блокировка** процессов, или компьютеру может не хватить ресурсов, чтобы обрабатывать другие запросы (в результате возникнет отказ в обслуживании). Взаимная блокировка может возникнуть, когда один процесс ждет определенное событие, которое может быть вызвано только другим процессом, который, в свою очередь, ожидает результаты от первого процесса. При этом возникнет ситуация, когда оба процесса будут просто стоять и ждать друг друга.

Одним из примеров взаимной блокировки может быть ситуация, когда процесс А использует ресурс 1 и нуждается в ресурсе 2 для выполнения своих задач, но процесс В использует ресурс 2 и нуждается в ресурсе 1, чтобы закончить свою работу. Таким образом, оба процесса находятся в состоянии взаимной блокировки, поскольку у них нет ресурсов, которые им необходимы, чтобы завершить уже начатую работу. Такие ситуации происходят все реже по мере совершенствования технологий программирования. Кроме того, операционные системы теперь имеют специальные функции, обнаруживающие такие ситуации и освобождающие используемый ресурс, либо контролирующие надлежащее распределение ресурсов между процессами.

Операционные системы используют различные методы для работы с запросами ресурсов и их освобождением, чтобы решать возможные ситуации с взаимными блокировками. В некоторых системах, если запрашиваемый ресурс недоступен в течение определенного периода времени, операционная система уничтожает процесс, который «держит» этот ресурс. При этом ресурс освобождается и становится доступен для использования другими

приложениями, а уничтоженный процесс перезапускается. Другие операционные системы могут требовать от запускающейся программы заранее запрашивать все ресурсы, которые ей понадобятся для работы, прежде чем она фактически начнет выполняться, либо требовать, чтобы программа освобождала все используемые ресурсы перед тем, как запрашивать другие ресурсы.

1.4. Работа процессов

Компьютеры могут запускать различные приложения и процессы одновременно. Процессы совместно используют ресурсы и взаимодействуют друг с другом. Некоторые области памяти, файлы и переменные разделяются между различными процессами. Учитывая все это, крайне важно позаботиться об обеспечении стабильной, безопасной вычислительной среды и поддержании ее целостности. Необходимо гарантировать, что несколько процессов не могут одновременно осуществлять операции чтения и записи одного и того же ресурса. Именно операционная система является основной программой, которая обеспечивает соответствующие защитные механизмы и не позволяет программам повредить область памяти друг у друга. Операционная система работает с процессором, чтобы обеспечить деление времени на кванты с помощью прерываний, для предоставления процессам адекватного доступа к центральному процессору. Это также гарантирует, что вредоносные приложения не окажут негативного влияния на важные функции системы.

Для защиты процессов друг от друга операционная система может использовать **изоляция процессов** (process isolation). Изоляция процессов необходима, чтобы процессы «не наступали друг другу на ноги», взаимодействуя небезопасным образом, и не оказывали негативного влияния на производительность друг друга. Более старые операционные системы не обеспечивали достаточной изоляции процессов. Если в такой операционной системе «зависала» одна программа, «зависали» и все другие программы, а иногда и сама операционная система. В операционной системе с надлежащей изоляцией процессов «зависание» одного процесса не влияет на другие запущенные программы. (Изоляция процессов требует использования вытесняющей многозадачности). Для реализации изоляции процессов могут использоваться различные методы:

- Инкапсуляция объектов
- Временное мультиплексирование общих ресурсов
- Разделение имен
- Виртуальное отображение

Когда процесс **инкапсулирован**, никакие другие процессы не могут взаимодействовать с его внутренним кодом. Если процессу А нужно взаимодействовать с процессом В, процессу А достаточно знать как взаимодействовать с интерфейсом процесса В. Интерфейс определяет порядок взаимодействия между двумя процессами. Программные компоненты должны знать, как правильно взаимодействовать с интерфейсами друг друга. Интерфейсы диктуют типы запросов (которые будут принимать процессы), и типы результатов (которые они будут предоставлять). Таким образом, два процесса могут общаться друг с другом, даже если они написаны на разных языках программирования, поскольку они знают, как общаться с интерфейсом друг с друга. Инкапсуляция обеспечивает **скрытие данных**, т.е. за пределами программных компонентов не будет известно, как работает процесс и не будет возможности манипулировать внутренним кодом процесса. Этот механизм обеспечивает целостность и модульность программного кода.

Временное мультиплексирование уже обсуждалось ранее, но этот термин до сих пор не использовался. **Временное мультиплексирование** – это технология, которая позволяет процессам использовать одни и те же ресурсы. Как было сказано ранее, процессор должен совместно использоваться множеством процессов. Хотя создается впечатление, что все

приложения работают (исполняют свои команды) одновременно, операционная система распределяет время между процессами. Мультиплексирование означает, что есть несколько источников данных, а также отдельные части данных, по конвейеру поступающие в один коммуникационный канал. В этом случае операционная система координирует различные запросы от различных процессов и проводит их конвейерную обработку посредством одного общего процессора. Операционная система должна обеспечить надлежащее мультиплексирование времени (совместное использование ресурсов) для обеспечения стабильной работы среды для программного обеспечения и пользователей.

Разделение имен просто означает, что все процессы имеют собственное уникальное имя или идентификатор. Обычно процессам назначаются идентификаторы процесса (PID), которые операционная система и другие процессы используют для обращения к ним. Если каждый процесс изолирован, это означает, что каждый процесс имеет свое собственное уникальное значение PID.

Отображение виртуального адресного пространства отличается от физического отображения памяти. Для приложений создается иллюзия того, что каждое из них является единственным запущенным приложением в операционной системе. Когда приложению требуется память для работы, оно говорит менеджеру памяти операционной системы, сколько памяти ему нужно. Операционная система выделяет необходимый объем памяти и связывает его с запрашивающим приложением. Приложение использует свою собственную схему адресации, которая обычно начинается с 0, но в действительности приложение не работает с *физическим* адресным пространством (хотя ему кажется, что работает). Вместо этого, оно работает в адресном пространстве, которое ему предоставил менеджер памяти. Физическая память – это микросхемы памяти в системе. Операционная система берет часть этой памяти и связывает ее с запрашивающим процессом. Как только процессу предоставлено собственное пространство памяти, он может адресовать эту часть как он хочет, это называется виртуальным отображением адресов. Виртуальное отображение адресов позволяет различным процессам иметь собственное пространство памяти; менеджер памяти гарантирует невозможность ненадлежащего взаимодействия одного процесса с памятью другого процесса. Это обеспечивает целостность и конфиденциальность.

1.5. Управление памятью

Чтобы обеспечить безопасную и стабильную среду, операционная система должна осуществлять надлежащее управление памятью – это одна из наиболее важных ее задач. В конце концов, все происходит в памяти. Это подобно тому, как наше существование зависит от кислорода и гравитации.

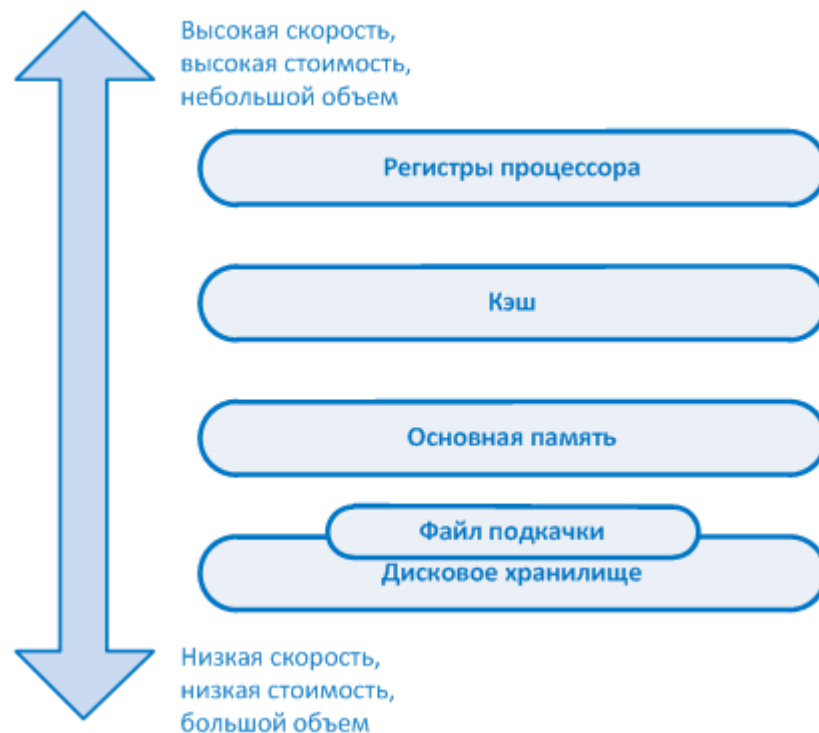
Целями управления памятью являются:

- Предоставление уровня абстракции программистам
- Максимизация производительности при ограниченном объеме доступной памяти
- Защита операционной системы и приложений, загруженных в память

Абстракция означает, что скрываются некие детали. Разработчики приложений не знают объем или тип памяти, который будет использоваться в каждой системе, на которую будет установлено их программное обеспечение. Если разработчик ориентировался на такие детали, его приложение будет иметь возможность работать только с очень ограниченным кругом систем, полностью соответствующих всем спецификациям. Для обеспечения переносимости приложений, менеджер памяти скрывает все вопросы в отношении памяти и просто предоставляет приложению сегмент памяти.

Каждый компьютер имеет иерархию памяти: небольшой объем скоростной и дорогой памяти (кэш и регистры) и большой объем более медленной и более дешевой (оперативная память, жесткие диски). Часть операционной системы, которая отслеживает использование

различных типов памяти, называется менеджером памяти. Он выделяет и освобождает различные сегменты памяти, реализует управление доступом, гарантируя, что процессы взаимодействуют только с их собственными сегментами памяти, переносит участки памяти из оперативной памяти на жесткий диск (файл подкачки).



Менеджер памяти отвечает за следующие пять основных функций:

Перемещение

- Переносит участки памяти из оперативной памяти на жесткий диск по мере необходимости (более подробно об этом рассказывается в разделе «Виртуальная память» далее в этом Домене).
- Предоставляет приложениям указатели, если их команды и сегменты памяти были перемещены в другое место в основной памяти.

Защита

- Ограничивает взаимодействие процессов только теми сегментами памяти, которые связаны с ними.
- Обеспечивает управление доступом к сегментам памяти

Совместное использование

- Использует комплексные защитные меры для обеспечения целостности и конфиденциальности при использовании процессами одних и тех же общих сегментов памяти.
- Позволяет множеству пользователей с различными уровнями доступа взаимодействовать с одними и теми же приложениями, запущенными в одном сегменте памяти.

Логическая организация

- Позволяет совместно использовать специальные программные модули, такие как динамически подключаемые библиотеки (DLL – dynamic link library).

Физическая организация

- Сегментирует пространство физической памяти для приложений и процессов операционной системы.

ПРИМЕЧАНИЕ. Динамически подключаемые библиотеки (DLL) представляют собой набор функций, которые приложения могут использовать при необходимости. Например, операционная система имеет библиотеку Crypt32.dll, используемую операционной системой и приложениями для выполнения криптографических функций. Windows имеет целый набор библиотек DLL, которые могут быть использованы приложениями.

Каким образом операционная система обеспечивает, что процесс взаимодействует только со своим сегментом памяти? Когда процесс создает поток для выполнения своих команд и обработки данных, процессор использует два регистра. **Базовый регистр** (base register) содержит начальный адрес, который был предоставлен этому процессу, а **регистр границы области памяти** (limit register) содержит конечный адрес, как показано на рисунке 3-6. Поток содержит адрес, где находятся команды и данные для обработки. Процессор сравнивает этот адрес с базовым регистром и регистром границы области памяти, чтобы убедиться, что поток не пытается получить доступ к сегменту памяти за пределами своих границ.



Рисунок 3-6. Базовый регистр и регистр границы области памяти используются для контроля выхода процесса за пределы своего сегмента памяти

Память также защищается с помощью пользовательского и привилегированного режимов выполнения, как уже упоминалось ранее. Более подробно это описано ниже в разделе "Режимы процессора и Кольца защиты" этого Домена.

1.6. Типы памяти

В следующих разделах рассматриваются различные типы памяти, используемые в компьютерных системах.

Память с произвольным доступом

Память с произвольным доступом (RAM – random access memory) – это разновидность временного хранилища данных, в котором данные и команды программного обеспечения могут храниться и изменяться. RAM используется операционной системой и приложениями

для выполнения операций чтения/записи. Эта память является временной, т.к. при отключении электропитания вся информация в ней теряется.

Микросхема RAM состоит из миллионов транзисторов и конденсаторов. В конденсаторах хранятся электрические заряды, которые преобразуются системой в 1 или 0. Транзистор работает как затвор (gate) или переключатель (switch). Для хранения двоичного значения 1, конденсатор держит внутри себя несколько электронов, которые имеют отрицательный заряд, а если конденсатор пуст, это соответствует значению 0. Когда операционная система записывает значение 0 поверх 1, фактически это приводит просто к освобождению конденсатора от электронов.

Одной из проблем является то, что конденсаторы не могут долго держать заряд. Поэтому контроллер памяти должен «перезаряжать» конденсаторы, постоянно считывая и записывая в них соответствующие значения – это называется регенерацией. Если контроллер памяти не обновит значение 1, конденсатор начнет терять электроны и значение в нем превратится в 0 или будет повреждено. Так организована работа *динамической RAM* (DRAM – dynamic RAM). Данные поступают на хранение в ячейки памяти, а затем постоянно динамически регенерируются, чтобы биты в них не исчезали. Операции регенерации осуществляется постоянно, что занимает определенное время, поэтому DRAM работает медленнее, чем статическая память.

ПРИМЕЧАНИЕ. Когда мы имеем дело с работой памяти, мы используем время, измеряемое в наносекундах (нс), что является одной миллиардной долей секунды. Если вы посмотрите на микросхему RAM и увидите маркировку 70 нс, это означает, что чтение и регенерация каждой ячейки памяти занимает 70 наносекунд.

Статическая RAM (SRAM – static RAM) не требует постоянной регенерации, т.к. она использует другие технологии, с помощью которых хранит биты в ячейках памяти без использования конденсаторов, но SRAM требует большего количества транзисторов, чем DRAM. SRAM не нуждается в постоянной регенерации, поэтому она быстрее, чем DRAM, но поскольку SRAM требует больше транзисторов, она занимает больше места в микросхеме. Производители не могут разместить на микросхеме также много ячеек памяти SRAM, как DRAM, поэтому SRAM стоит дороже. Таким образом, DRAM дешевле и медленнее, а SRAM дороже и быстрее. SRAM обычно используется как кэш-память, а DRAM – как основная оперативная память.

Аппаратная сегментация. Системам высшего уровня доверия может потребоваться реализовать аппаратную сегментацию памяти, используемой различными процессами. Это означает, что память разделяется физически, а не просто логически. Это добавляет еще один уровень защиты для обеспечения того, чтобы более низкопривилегированный процесс не имел доступа к памяти и не изменял пространство памяти процесса более высокого уровня.

Кроме того, существует множество других типов оперативной памяти. Основной причиной постоянной эволюции видов памяти является ее непосредственное влияние на скорость работы компьютера в целом. Многие люди ошибочно думают, что компьютер будет работать быстро, только если установить быстрый процессор. Однако, такие параметры, как тип и размер памяти, разрядность шины также имеют критическое значение. Представьте себе память компьютера в виде листов бумаги, используемых компьютером для хранения команд. Если компьютер использует листы небольшого размера (небольшой объем памяти), он будет тратить много времени на поиск нужных листов и их правильное расположение. Когда компьютер тратит много времени, перемещая данные из одной небольшой части памяти в другую, вместо реальной обработки данных, это приводит к существенному падению скорости системы.

Разрядность шины данных также вносит свой вклад в скорость системы. Вы можете представить себе шину данных, как шоссе, соединяющее различные части компьютера. Если необходимо переправить тонну данных от памяти к процессору, и есть только четырехполосное шоссе, то времени на перемещение данных будет затрачено больше, чем в

случае 64-полосного шоссе. Таким образом, процессор, тип памяти и ее размер, а также разрядность шины являются критическими составляющими производительности системы.

Вы должны быть знакомы со следующими дополнительными типами оперативной памяти:

- **SDRAM** (Synchronous DRAM) синхронизируется с процессором, возвращая ответ на поступивший управляющий сигнал не сразу, а после получения очередного тактового импульса. Это координирует действия с таймером процессора, синхронизируя таким образом частоты процессора и памяти. Это увеличивает скорость передачи данных.
- **EDO DRAM** (Extended Data Out DRAM) загружает следующий блок данных, пока предыдущий блок отправляется процессору для обработки. Это способ «предвидения», который повышает скорость доступа к памяти.
- **BEDO DRAM** (Burst EDO DRAM) работает аналогично EDO DRAM (и построена на ее базе), она также может одновременно передавать данные процессору и выполнять функцию чтения, однако при этом она может отправлять больше данных за раз (burst). Она может считывать и отправлять информацию из четырех адресов памяти в течение небольшого числа тактов.
- **DDR SDRAM** (Double Data Rate SDRAM) выполняет операции чтения на восходящем и нисходящем цикле тактового импульса. Таким образом, она выполняет две операции за такт вместо одной, что обычно удваивает скорость работы такой памяти по сравнению с обычной SDRAM при меньшем числе тактов.

ПРИМЕЧАНИЕ. Эти различные типы RAM требуют использования различных контроллеров для взаимодействия с ними, поэтому материнские платы всегда специфичны и предназначены для использования конкретных типов памяти.

Ну, хватит пока о RAM. Давайте взглянем на другие типы памяти, которые используются почти в каждом компьютере в мире.

Память только для чтения

Память только для чтения (ROM – read only memory) – это тип энергонезависимой памяти, в которой данные остаются в микросхемах памяти даже после выключения электропитания. Записанные в микросхемы ROM данные уже не могут быть изменены. Некоторые микросхемы ROM производятся с уже записанным программным обеспечением или подпрограммами. Программное обеспечение, которое хранится в ROM называется прошивкой.

Программируемая ROM (PROM – Programmable ROM) – это разновидность ROM, информация в которой может изменяться после ее изготовления, но только один раз, поскольку напряжение, которое используется для записи битов в ячейки памяти, фактически выжигает предохранители, которые соединяют отдельные ячейки памяти. Для прошивки PROM используются специальные программаторы.

Стираемая и программируемая ROM (EPROM – Erasable and programmable ROM) может быть очищена, изменена и обновлена. Данные в EPROM могут быть стерты с помощью ультрафиолетового света и перезаписаны электрическим способом. Для стирания микросхемы EPROM, ее нужно извлечь из компьютера и направить в имеющееся на ней специальное окно из кварцевого стекла ультрафиолетовый луч определенной мощности, который сотрет абсолютно все содержимое микросхемы. Чтобы не усложнять себе жизнь необходимостью использования ультрафиолета, изобрели другой вид ROM, данные на которой стираются электрическим способом без применения ультрафиолетовых лучей. Это **электрически стираемая и программируемая ROM** (EEPROM – Electrically erasable programmable ROM).

EEPROM похожа на EPROM, но хранящиеся на ней данные могут быть стерты и изменены электрическим способом непосредственно на плате с помощью специальной схемы

программирования и соответствующих электрических сигналов. Такая функция стирает только один байт за раз, что достаточно медленно. Поэтому была разработана еще одна технология, которая очень похожа на EEPROM, но работает быстрее.

Флеш-память (flash memory) – это специальный тип памяти, который в наше время используется в цифровых камерах, микросхемах BIOS, накопителях для ноутбуков, игровых консолях и т.д. Флеш-память изготавливается по твердотельной (solid-state) технологии, т.е. она не имеет движущихся частей, и используется больше как разновидность жестких дисков, чем как память.

Флэш-память работает на основе различных уровней напряжения, указывающих какое значение (1 или 0) хранится по определенному адресу. Ее работа больше похожа на работу ROM, чем RAM, она энергонезависима, также как ROM. Когда флэш-память должна быть стерта и возвращена в исходное состояние, программа активирует ее внутренние механизмы, которые выполняют стирание информации с помощью электрического поля. Удаление информации производится в рамках отдельных блоков или в рамках всей микросхемы, а не по одному байту за раз.

Флэш-память чаще всего используется в качестве небольшого дискового накопителя. Ее преимущества по сравнению с обычным жестким диском заключаются в том, что она меньше, быстрее и легче. Возможно когда-нибудь флеш-память полностью заменит жесткие диски, но сегодня она относительно дорога по сравнению с обычными жесткими дисками.

Ссылки по теме:

- [Unix/Linux Internals Course and Links](#)
- [Linux Knowledge Base and Tutorial](#)

Кэш-память

Кэш-память (cache memory) – это тип памяти, который используется для высокоскоростных операций чтения-записи. Когда система (ее программная логика) предполагает, что ей часто будет требоваться доступ к определенной информации, она организует хранение этой информации в кэш-памяти, чтобы она была быстро и легко доступна. Данные из кэша можно получить гораздо быстрее, чем данные, хранящиеся в оперативной памяти. Поэтому любая информация, необходимая процессору очень быстро и очень часто, обычно хранится в кэш-памяти, тем самым повышая общую скорость работы компьютерной системы.

Наш мозг хранит часто используемую информацию аналогичным образом. Если одна из основных должностных обязанностей Ирины состоит в том, чтобы она сообщала клиентам адрес компании, ее мозг сохраняет эту информацию в той части (разновидности кэша), из которой она может легко и быстро извлечена. Но если вдруг Ирину попросят вспомнить имя ее учителя в третьем классе, эта информация вряд ли хранится в ее «кэш-памяти», скорее она будет искать ее в более долгосрочном хранилище. Долгосрочное хранилище мозга можно сравнить с жестким диском компьютерной системы. Поиск и извлечение информации с жесткого диска занимает значительно больше времени по сравнению со специализированной кэш-памятью.

ПРИМЕЧАНИЕ. Различные материнские платы имеют различные типы кэш-памяти. Кэш-память уровня 1 (L1) быстрее, чем кэш-память уровня 2 (L2), а L2 быстрее, чем L3. Некоторые процессоры и контроллеры устройств имеют встроенную кэш-память. В процессоры и контроллеры встраивается обычно кэш-память уровней L1 и L2.

Отображение памяти

Поскольку различные типы памяти хранят различные типы данных, компьютерная система не хочет, чтобы любой пользователь, процесс или приложение имел доступ к любым типам

памяти в любое время, когда он захочет. Доступ к памяти должен контролироваться, чтобы данные не были повреждены, а критичная информация не была доступна неуполномоченным на доступ к ней процессам. Это обеспечивается с помощью отображения памяти и адресации.

Процессор является одним из самых доверенных компонентов системы, поэтому он может обращаться к памяти напрямую. Он использует физическую адресацию вместо указателей (логических адресов), поскольку физически (проводами) соединен с микросхемами памяти компьютера. Физическая адресация представляет собой точки пересечения проводников и рядов транзисторов в микросхеме памяти. Программное обеспечение использует не физическую, а логическую адресацию памяти. Такой не прямой доступ к памяти обеспечивает еще один уровень управления доступом программного обеспечения к памяти, что помогает обеспечить защиту и повысить эффективность системы. На рисунке 3-7 показана работа с памятью процессора (напрямую, с помощью физического адреса) и программного обеспечения (ненапрямую, посредством отображения памяти).

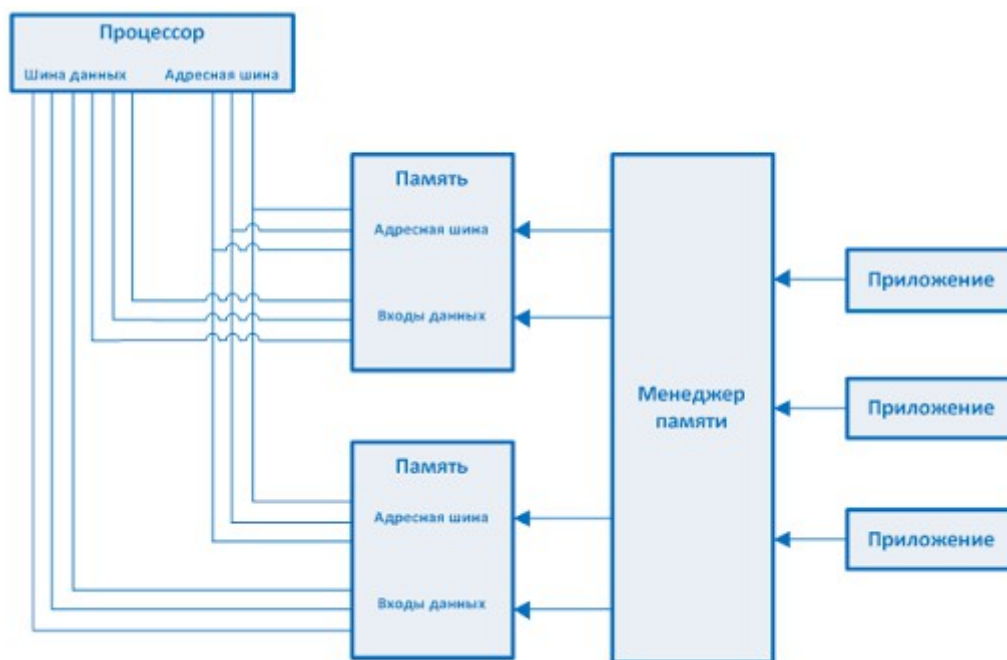


Рисунок 3-7. Доступ к памяти процессора и приложений организован по-разному

Давайте рассмотрим аналогию. Предположим, вы хотите купить земельный участок и вас заинтересовало объявление некоего г-на Маршалла, который продает свой участок. Вы хотели бы поговорить с г-ном Маршаллом, но вы не знаете его лично и не хотите давать ему свой домашний (физический) адрес, чтобы он пришел к вам. Вместо этого, вы предпочитаете более абстрактный и управляемый способ взаимодействия – вы звоните г-ну Маршаллу, чтобы поговорить с ним о земле и решить, хотите ли вы встретиться с ним лично. То же самое происходит и в компьютерах. Когда компьютер работает под управлением программного обеспечения, он не хочет без крайней необходимости позволять оказывать влияние на себя со стороны программ, написанных хорошими и плохими программистами. Компьютер разрешает программному обеспечению получать доступ к памяти с помощью индексных таблиц и указателей, вместо того, чтобы предоставить ему право на прямой доступ к памяти. Это один из способов самозащиты компьютерной системы.

Перед тем, как программе будет предоставлен доступ к памяти, проверяются ее права доступа. Команды программы выполняются способом, обеспечивающим невозможность влияния ее плохо написанного кода на другие программы и саму систему. Приложения и их процессы могут получить доступ только к той памяти, которая выделена для них (см. Рисунок 3-8). Такая архитектура обеспечивает защиту системы.

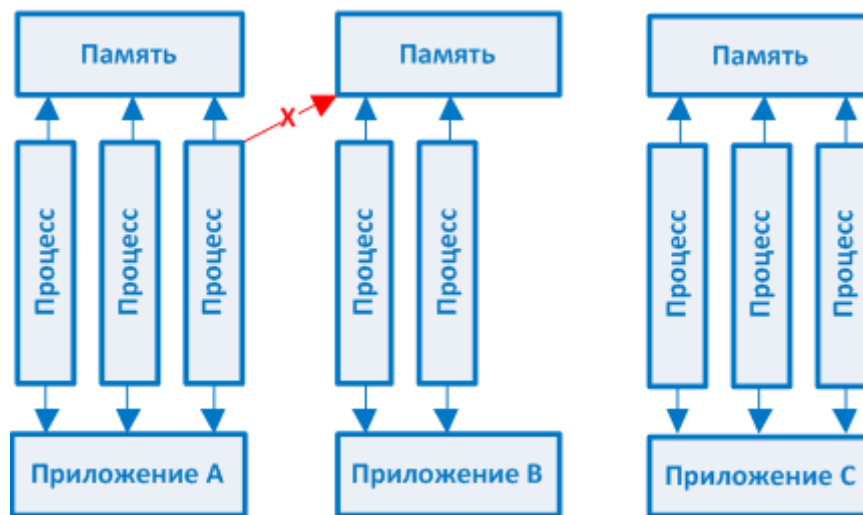


Рисунок 3-8. Приложения и процессы могут использовать только свои собственные сегменты памяти

Физические адреса памяти, которые использует процессор, называются **абсолютными адресами** (absolute address). Индексные адреса памяти, которые использует программное обеспечение, называются **логическими адресами** (logical address). Также используются **относительные адреса** (relative address), основанные на известном адресе и величине смещения относительно него. Как говорилось ранее, приложение «не знает», что память используется совместно различными приложениями. Когда программе для работы нужен сегмент памяти, она просто говорит менеджеру памяти, сколько памяти ей нужно. Менеджер памяти выделяет ей необходимый объем физической памяти, который может иметь физическую адресацию, например, от 34000 до 39000. Однако приложение не понимает такую адресацию. Оно разработано для работы с логическими адресами, начинающимися с 0 и заканчивающимися, скажем, 5000. Таким образом, менеджер памяти позволяет приложению использовать свою логическую схему адресации. Когда приложение обращается к одному из этих «фантомных» логических адресов, менеджер должен преобразовать его в реальный физический адрес.

Процесс отображения (преобразования) адресов показан на Рисунке 3-9. Если приложению нужно передать процессору команды и данные, физические адреса загружаются в базовый регистр и регистр границы области памяти. Когда поток передает команду для обработки, он использует логические адреса. Менеджер памяти преобразует логический адрес в физический адрес, чтобы процессор знал, где находится команда. Фактически поток использует относительный адрес, поскольку приложение работает в логическом адресном пространстве от 0 до 5000. Когда поток говорит, что ему нужно выполнить команду, которая находится в памяти по адресу 3400, менеджер памяти преобразует логический адрес 0 в фактический физический адрес, а затем находит физический адрес для логического адреса 3400. Т.е. логический адрес 3400 берется относительно начального адреса 0.

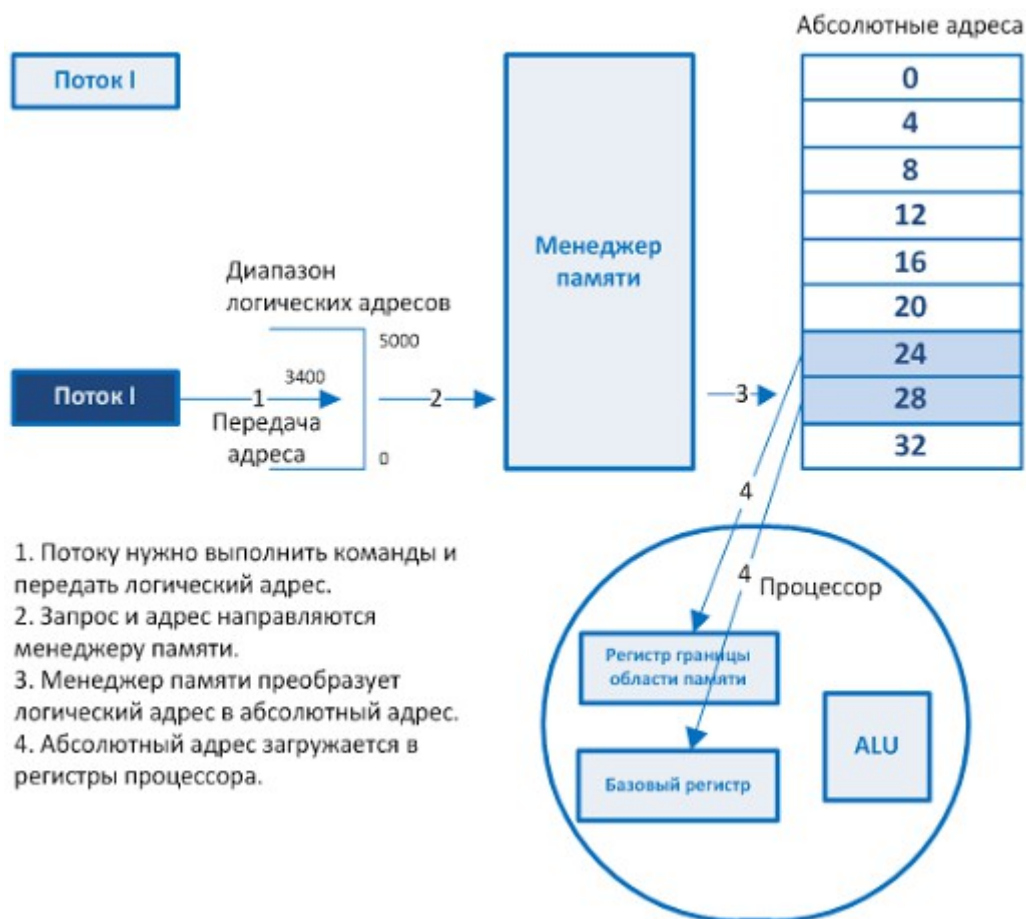


Рисунок 3-9. Процессор использует абсолютные адреса, а приложения – логические

Таким образом, приложения, работают в «своем мире» и используют «свои адреса», а менеджер памяти преобразует эти значения в реальные, абсолютные значения адресов.

Утечки памяти

Когда приложение запрашивает сегмент памяти, операционная система выделяет ему соответствующий объем памяти. Когда приложение заканчивает работу с этим сегментом памяти, оно говорит операционной системе, что нужно освободить эту память, чтобы она стала доступна другим приложениям. Это правильно. Однако некоторые плохо написанные приложения не сообщают системе, что ранее выделенная для них память им больше не требуется. Когда это происходит большое число раз, у операционной системы начинает заканчиваться свободная память, что крайне негативно влияет на производительность системы.

Когда утечку памяти обнаруживают хакеры, это позволяет им провести DoS-атаку. Например, когда было обнаружено, что Unix-реализация отдельных версий протокола Telnet, содержит утечку памяти, хакеры воспользовались этой проблемой, усилив ее воздействие. Они постоянно посылали запросы системам с этой уязвимостью. Системы выделяли ресурсы для каждого из этих запросов и не освобождали ее. Это вело к все большему и большему расходу памяти. В конечном итоге системе не хватало памяти и она зависала.

ПРИМЕЧАНИЕ. Утечки памяти могут происходить в операционных системах, приложениях и драйверах.

Существует две контрмеры против утечек памяти: лучше разрабатывать код программ, чтобы он должным образом освобождал память, либо использовать «сборщики мусора» (garbage collector), которые находят неиспользуемую память и сообщают системе, что ее можно считать свободной. Различные виды сборщиков мусора работают с различными операционными системами, языками программирования и алгоритмами.

1.7. Виртуальная память

Вторичным хранилищем (secondary storage) являются энергонезависимые носители информации, такие как жесткие диски, дискеты, компакт-диски и т.п. Когда RAM и вторичное хранилище используются совместно, вместе они образуют **виртуальную память**. Система использует пространство на жестком диске, чтобы увеличить объем своей оперативной памяти. *Пространство файла подкачки* – это зарезервированное пространство на жестком диске, которое используется для расширения оперативной памяти. Например, Windows-системы используют файл pagefile.sys для резервирования места на жестком диске. Когда система заполняет все пространство своей энергозависимой оперативной памяти, она записывает часть данных из памяти на жесткий диск. Когда программа запрашивает доступ к этим данным, они переносятся с жесткого диска обратно в память частями, называемыми **«страницами»** (pages). Этот процесс называется **«подкачкой»** (paging) виртуальной памяти. Доступ к данным, находящимся в файле подкачки на жестком диске, занимает больше времени, чем использование данных, находящихся в RAM, т.к. необходимо производить операции чтения/записи с жесткого диска. Внутреннее управление блоками, поддерживаемое операционной системой, отслеживает, какие страницы находятся в памяти, а какие доступны «офлайн» и готовы быть при необходимости перенесены в RAM для выполнения или обработки. Выигрыш заключается в создании впечатления, что система может хранить в памяти невероятное количество информации и команд программного обеспечения, как показано на Рисунке 3-10.



Рисунок 3-10. RAM и вторичное хранилище используются совместно, образуя виртуальную память

Следует иметь в виду, что данные, выгруженные из памяти в файл подкачки на жестком

диске, после выключения компьютера или завершения работы процессов, использовавших пространство файла подкачки, физически с жесткого диска не удаляются, просто указатели на соответствующие страницы памяти помечаются как свободные. Эти данные могут быть перехвачены и скомпрометированы. В очень безопасных операционных системах есть специальные процедуры для безопасной очистки пространства файла подкачки после завершения процесса и перед повторным использованием этого пространства. С помощью такой процедуры следует очищать файл подкачки перед завершением работы операционной системы, поскольку начиная с этого момента операционная система не сможет более обеспечивать контроль за доступом к содержимому жесткого диска.

ПРИМЕЧАНИЕ. Если программа шифрует данные при сохранении на жестком диске, то они расшифровываются при их обработке программой. Такие данные могут временно храниться в оперативной памяти в расшифрованном виде, а система может записать их в таком виде в файл подкачки на жестком диске. Потенциально, злоумышленники могут получить несанкционированный доступ к этому файлу.

Ссылки по теме:

- “Introduction to Virtual Memory,” by Tuncay Basar, Kyung Kim, and Bill Lemley
- “Virtual Memory,” by Prof. Bruce Jacob, University of Maryland at College Park (2001)
- LabMice.net links to articles on memory leaks

1.8. Режимы процессора и кольца защиты

Для обеспечения стабильности операционная система должна быть способна защитить себя от пользователей и их приложений. Для этого операционная система должна иметь возможность отличать свои операции от операций, выполняемых пользователями или приложениями, она должна отслеживать все действия приложений и убеждаться, что ни одно из них не нарушает политику безопасности системы. Это может быть очень сложной задачей, поскольку работа пользовательских приложений часто очень похожа на работу программного обеспечения операционной системы.

Операционная система имеет несколько защитных механизмов для исключения негативного влияния процессов друг на друга и на критичные компоненты самой операционной системы. Одним из таких механизмов является защита памяти, описанная ранее. Другим механизмом являются *кольца защиты* (protection rings). Эти кольца защиты предоставляют жесткие рамки, определяющие, что процессы могут делать и к чему иметь доступ в рамках каждого кольца. Процессы, которые работают в рамках внутреннего кольца, имеют больше привилегий, чем процессы, работающие на внешних кольцах. С внутренним кольцом разрешается взаимодействовать только наиболее надежным компонентам и процессам. Хотя в различных операционных системах число используемых ими колец защиты может различаться, основной принцип остается неизменным – процессы, работающие на внутреннем кольце, работают в защищенном режиме процессора, а процессы на внешних кольцах – в реальном режиме.

ПРИМЕЧАНИЕ. Фактическая архитектура колец, используемая системой, диктуется процессором и операционной системой. Аппаратная микросхема (процессор) создается с учетом предоставления определенного количества колец, операционная система должна быть разработана для работы в той же структуре колец. Это одна из причин, почему операционная система, разработанная для процессоров Intel, не может работать с процессорами Alpha, например. Они имеют различные архитектуры и способы интерпретации наборов команд.

Компоненты операционной системы работают на внутреннем кольце, что дает им полный доступ к ячейкам памяти, периферийным устройствам, системным драйверам, критичным параметрам конфигурации. Поскольку это кольцо предоставляет наиболее опасный доступ к важнейшим ресурсам, оно является самым защищенным. Приложения обычно работают на кольце 3, на котором ограничен доступ к памяти, периферийным устройствам и драйверам, на этом кольце осуществляется управление доступом посредством системных вызовов и с

помощью служб операционной системы. Различные кольца защиты показаны на рисунке 3-11. Типы и наборы команд, которые могут отправить процессору приложения, работающие на внешних кольцах, более ограничены по сравнению с внутренним кольцом. Если приложение пытается отправить процессору команды, которые выходят за рамки его разрешений, процессор генерирует исключение и пытается завершить работу этого приложения.

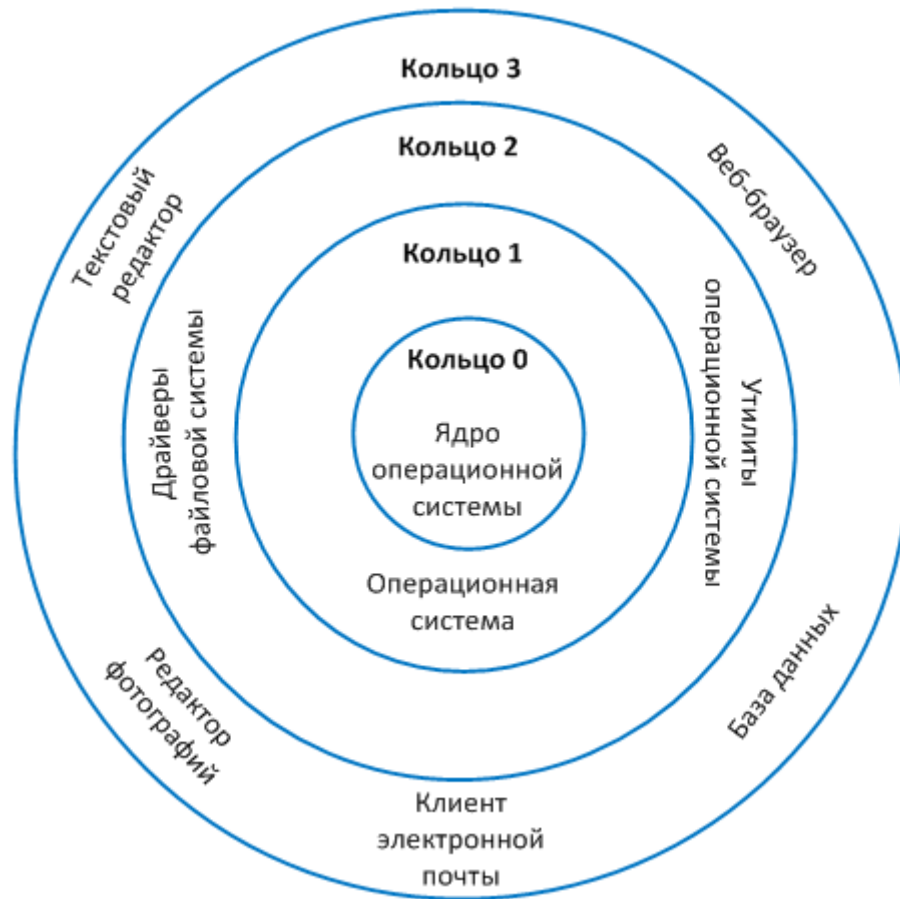


Рисунок 3-11. Более доверенные процессы работают на внутренних кольцах

Кольца защиты обеспечивают доступность, целостность и конфиденциальность необходимые многозадачной операционной системе. Наиболее часто используется архитектура с 4-мя кольцами защиты:

- **Кольцо 0:** Ядро операционной системы
- **Кольцо 1:** Остальные части операционной системы
- **Кольцо 2:** Драйверы и утилиты ввода/вывода
- **Кольцо 3:** Приложения и действия пользователей

Эти кольца защиты реализуют промежуточный слой между субъектами и объектами и используются для управления доступом при попытках субъектов получить доступ к объектам. Кольцо определяет уровень доступа к критичным ресурсам системы. Чем меньше номер кольца, тем больше привилегий у процесса, который работает в рамках этого кольца. Каждому субъекту и объекту логически присвоено число (от 0 до 3), соответствующее уровню доверия операционной системы к нему. Субъекты могут получить доступ только к объектам в рамках того же кольца, на котором находятся они сами, либо в рамках внешнего по отношению к ним кольца, но они не могут напрямую взаимодействовать с объектами на внутренних по отношению к ним кольцах. Так, например, субъекты на кольце 3 могут получить прямой доступ только к объектам на том же кольце 3, а субъекты на кольце 1 могут

получить прямой доступ к объектам на кольцах 1, 2 и 3. Если приложению требуется доступ к компонентам на других кольцах, к которым ему не разрешен прямой доступ, оно отправляет соответствующий запрос операционной системе для выполнения необходимых задач. Для этого используются системные вызовы, позволяющие приложению выполнить команды, недоступные в реальном режиме. Запрос передается системным службам операционной системы, которые работают на более привилегированном уровне и могут выполнять более критичные функции.

Когда операционная система выполняет команды процессов, находящихся на кольцах 0 и 1, она работает в защищенном режиме. Когда операционная система выполняет команды приложений и процессов на кольце 3, она работает в реальном режиме. Реальный режим предоставляет гораздо более ограниченную среду для работы приложения, что, в свою очередь, защищает систему от неправильного поведения программ.

ПРИМЕЧАНИЕ. Многие современные операционные системы не используют второе кольцо защиты.

Резюмируем вкратце информацию о кольцах защиты. Операционная система должна работать в рамках структуры и границ, предоставленных процессором. Процессор предоставляет операционной системе кольца защиты, пронумерованные от 0 до 3. Операционная система логически распределяет процессы по различным кольцам, основываясь на уровне своего доверия к этим процессам. Т.к. ядро операционной системы является самым надежным и доверенным компонентом, оно и его процессы размещаются на кольце 0. Остальные процессы операционной системе размещаются на кольце 1, а все пользовательские приложения размещаются на кольце 3.

Когда процессу, работающему на кольце 0, нужно передать процессору команды для выполнения, процессор проверяет номер кольца и убеждается, что этому процессу можно полностью доверять. После этого процессор позволяет этому процессу работать с любыми своими функциями, предоставляемыми процессам. Некоторыми из наиболее критичных действий, для выполнения которых необходимы максимальные привилегии (нахождение на кольце 0), являются операции ввода/вывода и доступ к памяти.

Когда другому процессу, находящемуся на кольце 3, нужно передать процессору команды для выполнения, процессор снова проверяет номер кольца процесса. Поскольку это процесс с кольца 3, процессор знает, что операционная система доверяет этому процессу в наименьшей степени, поэтому он ограничивает объем своих функциональных возможностей, доступных этому процессу.

1.9. Архитектура операционной системы

Операционная система может быть разработана с использованием различных типов архитектуры. Архитектура – это платформа, которая определяет размещение и взаимодействие служб и функций операционной системы. В этом разделе рассматриваются монолитная, многоуровневая и клиент/серверная архитектуры.

Монолитную архитектуру операционной системы (monolithic operating system architecture) обычно называют «большим беспорядком», ей явно не хватает структуры. Операционная система состоит в основном из различных процедур, бессистемно вызывающих друг друга. Системы этого типа имеют только один уровень безопасности, модули кода в них могут вызывать друг друга по мере необходимости. Взаимодействие между модулями не структурировано и не управляется как в многоуровневой архитектуре, скрытие данных не предусмотрено. Примером монолитной операционной системы является MS-DOS.

Многоуровневая архитектура операционной системы (layered operating system architecture) разделяет функциональность системы на иерархические уровни. Первой операционной системой, которая использовала многоуровневую архитектуру, была система THE

(Technische Hogeschool Eindhoven). TNE имела 5 уровней функциональности. Уровень 0 управлял доступом к процессору и обеспечивал многопрограммность; уровень 1 выполнял управление памятью; уровень 2 обеспечивал межпроцессное взаимодействие; уровень 3 был связан с устройствами ввода/вывода; а на уровне 4 работали приложения. Уровень 5 был пользовательским уровнем и не реализовывался напрямую TNE. Процессы на различных уровнях имеют интерфейсы для использования процессами, находящимися на уровень выше или ниже них.

Это отличается от монолитной архитектуры, в которой каждый модуль может взаимодействовать с любым другим модулем. Многоуровневые операционные системы обеспечивают *скрытие данных*, не позволяющее командам и данным (упакованным в виде процедур) на одном уровне получить прямой доступ к командам и данным на любых других уровнях. Каждая процедура на каждом уровне имеет доступ только к своим данным и набору функций, которые ей необходимы для выполнения своих задач. Если процедура может получить доступ к большему числу процедур, чем ей реально необходимо, это может стать причиной компрометации системы. Например, если атакующий сможет скомпрометировать и получить контроль над одной процедурой, а эта процедура имеет прямой доступ ко всем другим процедурам, атакующий сможет скомпрометировать и более привилегированные процедуры, получив таким образом возможность выполнить более разрушительные действия.

Монолитная операционная система имеет только один уровень безопасности. В многоуровневой системе каждый уровень должен обеспечивать свою собственную безопасность и управление доступом. Если один уровень управляет безопасностью всех других уровней, то этот уровень имеет слишком много информации (и доступа) обо всех объектах на всех уровнях, что является прямым нарушением концепции скрытия данных. Разделение программного обеспечения и его кода на модули повышает уровень предоставляемых системой гарантий, т.к. компрометация одного модуля не означает появления уязвимости для всех остальных модулей. Примерами многоуровневых систем являются TNE, VAX/VMS, Multics, Unix (хотя TNE и Multics больше не используются).

Другим подходом к проектированию системы является *клиент/серверная архитектура* (client/server architecture), в которой те части программного обеспечения и функции, которые ранее размещались в монолитном ядре, теперь находятся на более высоких уровнях операционной системы. Функции операционной системы делятся на несколько различных процессов, которые выполняются в реальном, а не в защищенном режиме.

Цель клиент/серверной архитектуры заключается в том, чтобы как можно больше кода вынести из ядра, работающего в защищенном режиме, оставив только компактное ядро, называемое микроядром. В этой модели процессы, осуществляющие запросы, называются клиентами, а процессы, выполняющие эти запросы, серверами (например, сервер файловой системы, сервер памяти, сервер ввода/вывода, сервер процессов). Эти серверы обычно называют подсистемами. Клиентом может быть пользовательский процесс или другая операционная система.

ПРИМЕЧАНИЕ. Не путайте клиент/серверную архитектуру операционной системы с клиент/серверной сетевой архитектурой, которая традиционно ассоциируется с понятием "клиент/сервер". В сети, приложение работает в клиент/серверной модели для использования распределенных вычислительных возможностей. При этом клиентская часть программного обеспечения устанавливается на рабочих станциях, а серверная часть, как правило, на сервер базы данных и/или сервер приложения.

1.10. Домены

Домен (domain) – это набор объектов, к которым могут обращаться субъекты. Доменом могут быть все ресурсы, к которым могут обращаться пользователи; все файлы, доступные программам; сегменты памяти, доступные процессам; службы и процессы, доступные

приложению. Субъекту необходим доступ к объектам (ресурсов) для выполнения своих задач, и именно домен определяет, какие объекты доступны этому субъекту, а к каким доступ не предоставляется.

ПРИМЕЧАНИЕ. Помните, что поток – это часть процесса. Когда генерируется поток, он использует тот же домен (ресурсы), что и породивший его процесс.

Эти домены должны быть идентифицированы, разделены и надлежащим образом реализованы. Операционная система и процессор могут работать в защищенном, либо реальном режиме. Даже это разделение режимов, связанных с кольцами защиты, обусловлено необходимостью определения различных доменов. Когда команды процесса выполняются в защищенном режиме, этот процесс работает в очень широком домене (имеет доступ к большому числу ресурсов) и поэтому может выполнять больше различных действий. Работая в защищенном режиме, процесс операционной системы может получить прямой доступ к сегментам памяти, переносить данные из незащищенного домена в защищенный домен, напрямую взаимодействовать с аппаратными устройствами. Приложение, работающее в реальном режиме, не может напрямую обращаться к памяти и аппаратным устройствам, оно имеет существенно более ограниченный объем доступных ресурсов и возможных действий. Приложению предоставляется доступ только к выделенным для него сегментам памяти, причем этот доступ предоставляется не напрямую и строго контролируется.

Процессу, находящемуся в привилегированном домене, при выполнении своих команд и обработке своих данных нужна уверенность в том, что на его среду не могут оказать негативного воздействия программы из других доменов. Это называется **доменом выполнения** (execution domain). Поскольку процесс в привилегированном домене имеет доступ к критичным ресурсам системы, его среда должна быть защищена от вредоносного программного кода и неожиданных действий, исходящих от программ в других доменах. Одни системы могут разделять только пользовательские и привилегированные области, тогда как другие имеют более сложную архитектуру, содержащую до десяти доменов выполнения.

Домены выполнения напрямую связаны с кольцами защиты, на которых находятся субъекты и объекты. Чем ниже номер кольца защиты, тем выше его привилегии и тем шире его домен. Эта концепция проиллюстрирована на Рисунке 3-12.

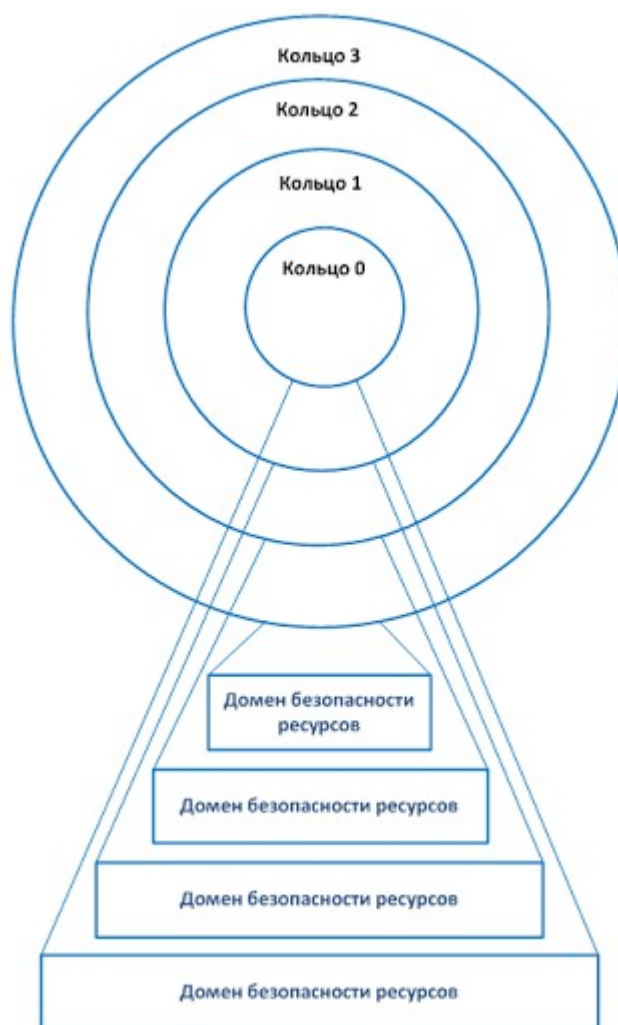


Рисунок 3-12. Чем выше уровень доверия, тем большее количество ресурсов доступно (т.е. более широкий домен)

1.11. Разделение на уровни и скрытие данных

Хотя теоретически существует три основных типа архитектуры операционных систем, термины **разделение на уровни** (layering) и **скрытие данных** (data hiding) часто используют, когда речь идет о механизмах защиты операционных систем, в т.ч. в отношении операционных систем с клиент/серверной архитектурой, поскольку они также используют технологии разделения на уровни и скрытия данных для собственной защиты.

Многоуровневая архитектура операционной системы *в основном* учитывает распределение функциональности и ее доступность пользователям и программам. Это обеспечивает иерархию функциональности, тогда как архитектура клиент/сервер предоставляет функциональность в более линейном виде. В клиент/серверной архитектуре запросу не нужно проходить через различные уровни, он просто направляется необходимой подсистеме. Но с точки зрения безопасности, обе архитектуры используют уровни и скрытие данных для защиты критичных процессов операционной системы от приложений, в сами приложения от других приложений.

Хотя используется так много различных терминов – домены выполнения, кольца защиты, разделение на уровни, скрытие данных, домены защиты, режимы процессора и т.п., в действительности все они являются различными способами описать одни и те же вещи, происходящие внутри любой современной операционной системы. Все эти концепции должны работать вместе очень хорошо организованным образом для надлежащей работы всей операционной системы и обеспечения необходимого уровня безопасности.

Как было сказано ранее, операционная система и процессор работают в рамках той же архитектуры, которую предоставляют кольца защиты. Домен защиты процесса (домен выполнения) определяется кольцом защиты, на котором этот процесс находится. Когда процесс передает команды процессору для выполнения, тот исполняет их в соответствующем режиме (реальном или защищенном), зависящем от кольца защиты, на котором находится процесс. Разделение на уровни и скрытие данных обеспечивается путем размещения процессов на различных кольцах защиты, а также управления взаимодействием менее доверенных процессов с более доверенными, работающими на более низких кольцах защиты.

Таким образом, разделение на уровни является способом обеспечения буферов между более доверенными и менее доверенными процессами. Менее доверенные процессы не могут напрямую взаимодействовать с более доверенными процессами, они должны направлять свои запросы службам операционной системы. Эти службы работают в качестве посредника или вышибалы, гарантируя невозможность несанкционированного доступа к более доверенным процессам. Такая архитектура защищает операционную систему в целом, включая все приложения и деятельность пользователей, происходящие внутри нее.

1.12. Эволюция терминологии

Хотя *теоретически* монолитная, многоуровневая и клиент/серверная архитектуры описывают то, как построена *операционная система*, эти термины накладываются друг на друга при описании построения ядра. В реальной жизни и на экзамене CISSP, когда вы видите термин «монолитная система», в действительности это означает, что весь код ядра работает в защищенном режиме. Сбивает с толку то, что платформу такой операционной системы называют монолитной (monolithic framework), и есть отдельный термин, относящийся только к ядру – монолитное ядро (monolithic kernel), однако в настоящее время эти термины объединены. Таким образом, если сегодня используется термин «монолитная система», он относится к тому, как построено ядро.

ПРИМЕЧАНИЕ. Помните, что режим ядра (kernel mode), защищенный режим (protected mode), привилегированный режим (privileged mode), режим супервизора (supervisory mode) – все это означает одно и то же.

Использование монолитного ядра означает, что вся работа ядра выполняется в защищенном режиме на кольце 0, как показано на Рисунке 3-13. Windows NT, 2000 и Vista являются всем известными монолитными системами, поскольку все их службы, реализующие функциональность операционной системы, выполняются в режиме ядра. Это создает угрозу безопасности, потому что если происходит проблема с одним из процессов внутри ядра, это может повлиять на все ядро. Это также означает, что большой объем кода работает в защищенном режиме, соответственно больше кода может быть использовано злоумышленниками для получения высокого уровня контроля над системой. Это означает, что создать безопасную монолитную систему очень сложно, и тем более сложно гарантировать ее безопасность.

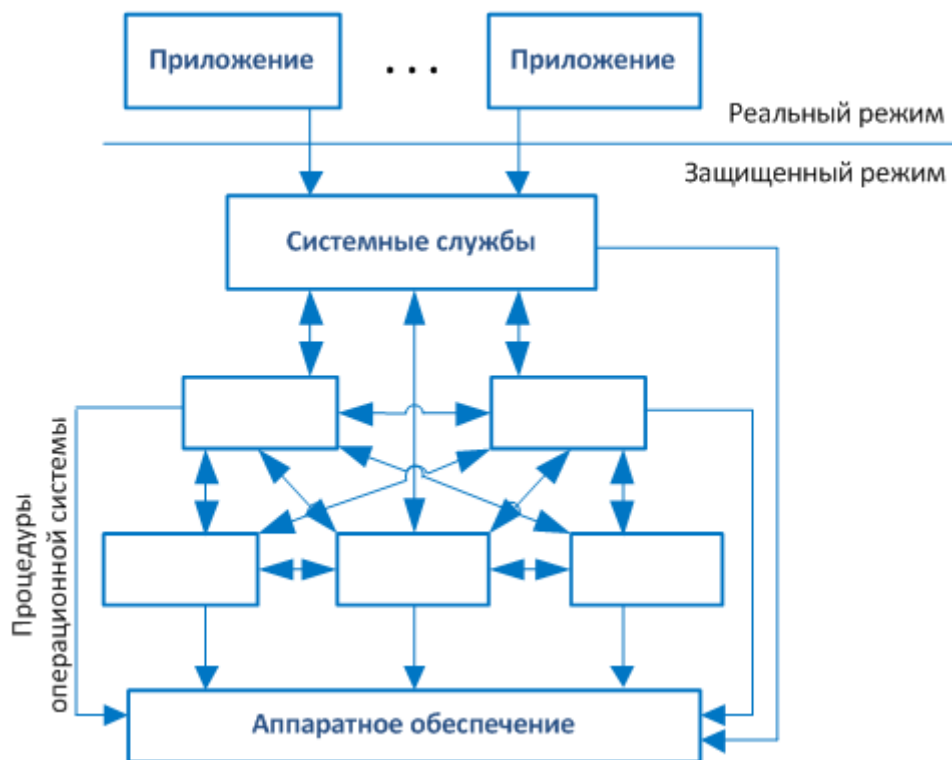


Рисунок 3-13. Подсистемы выполняют запросы клиентских процессов

Причиной, из-за которой операционная система Windows (а также Unix и Linux) была разработана для использования монолитного ядра, является производительность. Ведь когда одни компоненты ядра работают в реальном режиме, а другие в защищенном, выполнение команд занимает у процессора гораздо больше времени, поскольку ему нужно постоянно переключаться между защищенным и реальным режимами.

Таким образом, большинство современных наиболее распространенных операционных систем в основном используют кольца 0 и 3 архитектуры колец защиты.

Все ядро и драйверы устройств выполняются на кольце 0, а все пользовательские приложения – на кольце 3. Поскольку драйверы работают в защищенном режиме, крайне важно, чтобы они были правильно написаны и не могли быть использованы злоумышленниками каким-либо образом. Поскольку большинство драйверов устройств предоставляются третьими сторонами (собственно производителями этих устройств), нет гарантий, что они действительно были разработаны надлежащим образом и безопасны. Именно поэтому Microsoft установила очень жесткие требования к драйверам в операционной системе Windows Vista. Сторонние поставщики должны теперь при разработке драйверов учитывать гораздо более строгие критерии, чтобы их драйверы могли быть загружены в операционную систему.

1.13. Виртуальные машины

Если вы работали с компьютерами 10-15 лет назад, вы помните компьютерные игры тех лет, имеющие простейшую компьютерную графику, не похожую на графику современных игр. В те времена игры были 16-битными и написаны они были для работы в 16-битной среде MS-DOS. Когда операционные системы Windows перешли с 16 на 32 бита, в 32-битных операционных системах была обеспечена поддержка обратной совместимости, чтобы можно было продолжать работать с 16-битными приложениями. Это реализовывалось путем создания **виртуальных машин** (virtual machine), на которых запускались 16-битные приложения (в т.ч. Игры).

Виртуальная машина является искусственной средой. Когда 16-битному приложению,

выполняющемуся в рамках виртуальной машины, необходимо взаимодействовать с операционной системой, виртуальная машина сама делает системные вызовы и взаимодействует с памятью компьютера, переводя запросы приложения в форму, необходимую для работы в 32-битной среде. Таким образом, виртуальная машина эмулирует 16-битную операционную систему и когда приложение делает запрос, виртуальная машина преобразует его из 16-битного в 32-битный формат (что называется *переключением* (thunking)), чтобы операционная система могла обработать его должным образом. Когда система отправляет ответ на этот запрос, он изменяется с 32-битного формата на 16-битный, который понимает приложение.

В наше время виртуальные машины стали гораздо более «продвинутыми». Обычная виртуализация позволяет запускать на одном компьютере несколько операционных систем, существенно увеличивая коэффициент полезного использования аппаратных ресурсов компьютера и давая другие преимущества. Виртуализация – это создание виртуальных экземпляров операционных систем, приложений и устройств хранения.

В современном жаргоне экземпляр виртуальной операционной системы называется виртуальной машиной. Виртуальной машиной обычно называют гостевую операционную систему, запущенную в среде основной операционной системы компьютера. Виртуализация позволяет одновременно запускать на одном физическом компьютере несколько гостевых операционных систем, динамически распределяя ресурсы физической системы. Ресурсы компьютера, такие как оперативная память, процессорное время, хранилище информации, эмулируются основной средой для виртуальных сред. Виртуальные машины не обращаются напрямую к этим ресурсам, они взаимодействуют только со средой основной системы, которая обеспечивает управление системными ресурсами.

Таким образом вы можете на одном компьютере запустить несколько различных операционных систем одновременно (например, Windows 2000, Linux, Unix и Windows 2008). Представьте себе дом с несколькими комнатами. Каждая операционная система получает свою отдельную комнату, но все они совместно используют одни и те же ресурсы, которые обеспечивает дом – фундамент, электричество, вода, крыша и т.д. Операционной системе, которая «живет» в отдельной комнате, не нужно знать о других операционных системах и взаимодействовать с ними, чтобы использовать ресурсы дома. То же самое относится к компьютеру: каждая операционная система совместно с другими использует ресурсы, предоставляемые физической системой (такие как память, процессор, шины и т.п.). Они «живут» и работают в своей отдельной «комнате», являющейся гостевой виртуальной машиной. Физический компьютер является основной (host) системой.

Зачем это нужно? Основная причина заключается в том, что это значительно дешевле, чем иметь полную физическую систему для каждой операционной системы. Если все они могут «жить» на одной системе и совместно использовать ресурсы, их стоимость значительно снижается. По той же самой причине люди совместно арендуют квартиры, разделяя между собой арендную плату и совместно используя ресурсы квартиры.

Виртуализация также применяется и в качестве механизма безопасности. В качестве примера можно привести Java Virtual Machine (JVM), которая используется любым современным веб-браузером. JVM создает виртуальные машины (называемые песочницами (sandbox)), выполняющие java-апплеты. Это является защитным механизмом, поскольку песочница держит апплет внутри себя и не позволяет ему взаимодействовать напрямую с операционной системой и файлами. Действия, которые апплет пытается выполнить, проверяются JVM на предмет их безопасности. Если JVM решает, что действия безопасны, она выполняет запрос самостоятельно от имени апплета.

ПРИМЕЧАНИЕ. На сегодняшний день уже были написаны вредоносные программы, которые могут выходить за рамки песочницы, чтобы выполнять свои действия без контроля со стороны JVM. Эти способы компрометации, а также Java и JVM будут рассмотрены более подробно в

Приведенный ниже список преимуществ использования виртуализации взят со страницы www.kernelthread.com/publications/virtualization. Он написан достаточно давно, но он актуален и по сей день, и его достаточно для сдачи экзамена CISSP.

- Виртуальные машины могут использоваться для объединения нагрузки от нескольких серверов на меньшем количестве физических машин, может быть даже на одной машине (консолидация серверов). Выгодами при этом являются: снижение затрат на физическое оборудование, среду, управление и администрирование серверной инфраструктуры.
- Необходимость запуска унаследованных приложений также может быть реализована с использованием виртуальных машин. Унаследованное приложение может просто не работать на новом оборудовании и/или в новой операционной системе. Но даже если оно заработает, то в случае, если оно является серверным, для него возможно потребуется выделить отдельный сервер. Было бы выгоднее объединить все такие приложения на одном сервере. Это может быть очень трудно реализовать без использования виртуализации, поскольку такие приложения обычно не поддерживают сосуществование с другими приложениями в одной среде выполнения.
- Виртуальные машины могут использоваться для обеспечения безопасных, изолированных «песочниц» (sandboxes) для запуска недоверенных приложений. Можно даже создавать такие среды выполнения динамически, «на лету», например, по мере того, как вы скачиваете что-то из Интернета и запускаете это. Виртуализация – важная концепция создания безопасных вычислительных платформ.
- Виртуальные машины могут использоваться для создания операционных систем или сред выполнения с ограниченными ресурсами, для проверки правильной работы планировщика и менеджеров ресурсов. Это особенно важно при создании операционных систем, поддерживающих функциональность QoS (качество сервиса).
- Виртуальные машины могут предоставить иллюзию оборудования или конфигурации оборудования, которого вы реально не имеете (такого как устройства SCSI или несколько процессоров). Виртуализация также может использоваться для эмуляции сети независимых компьютеров.
- Виртуальные машины могут использоваться для одновременного запуска нескольких операционных систем, в частности, более старых версий или полностью других систем, из них можно формировать системы для «горячего резерва». Некоторые унаследованные системы могут быть сложно или вообще невозможно запустить на современном физическом оборудовании.
- Виртуальные машины могут использоваться в качестве мощных средств отладки и мониторинга производительности. При этом вы можете отлаживать операционную систему без потери производительности или реализуя более сложные сценарии отладки.
- Виртуальные машины могут изолировать то, что на них запущено, и не позволять сбоям и ошибкам работающего на них программного обеспечения влиять на работу основной системы. Вы можете умышленно вызвать ошибки в таком программном обеспечении, чтобы проанализировать его последующее поведение.
- Виртуальные машины – прекрасный инструмент для исследований и академических экспериментов. Поскольку они обеспечивают изоляцию, они более безопасны для такой работы. Они полностью инкапсулируют состояние запущенной системы, причем вы можете сохранить состояние системы, исследовать его, изменить, загрузить снова и т.д.

- Виртуализация может упростить и повысить управляемость таких задач, как миграция систем, резервирование и восстановление.
- Виртуализация аппаратных средств популярна для совместного хостинга. Многие из приведенных выше преимуществ делают такой хостинг безопасным, эффективным по стоимости и привлекательным в целом.

Ссылки по теме:

- *The Design of PARAS Microkernel*, Chapter 2, “Operating System Models,” by Rajkumar Buyya (1998)
- Chapter 12, “Windows NT/2000,” by M.I. Vuskovic
- Answers.com definitions of virtual machine

1.14. Дополнительные устройства хранения

Помимо памяти, рассмотренной ранее, следует рассмотреть отдельные виды физических устройств хранения информации, наряду со связанными с ними последствиями нарушения безопасности. Многие (если не все) из различных устройств хранения информации, используемых в настоящее время, позволяют украсть или скомпрометировать данные компании. По мере сокращения их размеров, их возможности возросли. Дискеты даже при относительно малой емкости (около 1.44MB), уже давно стали распространенным источником вирусов и кражи данных. Вор, имеющий физический доступ к компьютеру с небезопасной операционной системой, может использовать дискету для загрузки системы.

Многие рабочие станции имеют BIOS, позволяющий загружать компьютер с внешних носителей (дискета, компакт-диск, флеш-накопитель). Это можно усложнить, установив пароль на BIOS, запретив использование неразрешенных носителей информации, а также обеспечив контроль физического доступа к компьютеру.

Следует также учитывать, что внешние носители информации могут быть украдены или потеряны, что может привести к серьезным последствиям.

Любые портативные носители информации могут стать причиной нарушения безопасности. Особенно много проблем вызывают различные флеш-накопители, телефоны и плееры, которые могут хранить десятки гигабайт информации и при этом их крайне просто подключить практически к любому компьютеру через USB-порт. Первым шагом в предотвращении этой угрозы является обновление существующей политики безопасности (или внедрение новых политик), чтобы учесть новые технологии. Даже сотовые телефоны могут быть подключены к компьютеру для передачи данных, звука, изображений и видео, что может выходить за границы старой политики безопасности. Должны учитываться проблемы безопасности и уязвимости таких технологий, как Bluetooth, FireWire, Blackberry и т.д.

1.15. Управление устройствами ввода/вывода

Мы уже рассмотрели многие из обязанностей операционной системы, но мы еще не остановились. Операционная система также должна управлять всеми устройствами ввода/вывода. Она посылает им команды, разрешает прерывания от них, когда им нужно взаимодействовать с процессором, а также обеспечивает интерфейс между устройствами и приложениями.

Устройства ввода/вывода обычно делят на блочные и символьные. Блочные устройства работают с блоками данных фиксированного размера. Каждый блок имеет собственный уникальный адрес (пример – любой дисковый накопитель). Символьные устройства (например, принтер, сетевая карта или мышь) работают с потоками символов неопределенного размера. Эти данные не адресуются.

Когда пользователь дает команду распечатать документ, открыть сохраненный файл в текстовом редакторе или сохранить файлы на переносном диске, такая команда идет от приложения (с которым работает пользователь) к запрошенному устройству через операционную систему. Операционная система использует драйвер соответствующего устройства для взаимодействия с его контроллером, который может представлять собой плату расширения, установленную в специальный слот на материнской плате. Контроллер имеет собственное программное обеспечение, позволяющее обмениваться данными между устройством и операционной системой. Операционная система отправляет команды в соответствующие регистры контроллера, а контроллер выполняет необходимые действия по чтению и записи данных. Если команда требует извлечь данные с жесткого диска, контроллер считывает необходимые биты данных с жесткого диска, помещает их в блок необходимого размера, выполняет контрольные функции для проверки целостности считанных данных. Если проверка целостности не выявила нарушений, данные помещаются в память для передачи процессору.

Операционная система нуждается в том, чтобы использование и освобождение устройств и ресурсов компьютера происходило правильным образом. Различные операционные системы работают с устройствами и ресурсами по-разному. Например, считается, что Windows NT является более стабильной и безопасной средой обработки данных, чем Windows 9x, поскольку приложения в Windows NT не могут выполнять прямые запросы к аппаратным устройствам. Windows NT и Windows 2000 используют гораздо более управляемые методы доступа к устройствам, чем Windows 9x. Эти методы помогают защитить систему от плохо написанного кода, неправильно использующего и/или неправильно освобождающего ресурсы. Такой уровень защиты позволяет обеспечить целостность и доступность ресурсов.

Прерывания

Когда устройство ввода/вывода завершает задачу, оно должно проинформировать процессор, что необходимые данные уже в памяти и доступны для обработки. Для этого контроллер устройства отправляет по шине специальный сигнал (прерывание), который отслеживается контроллером прерываний. Если процессор занят, а прерывание устройства не имеет более высокого приоритета, чем уже выполняющаяся работа, то устройство ожидает, когда процессор освободится. Контроллер прерываний отправляет сообщение процессору, говоря ему, что устройство требует внимания. Операционная система имеет таблицу (называемую вектором прерывания) всех подключенных устройств ввода/вывода. Процессор сравнивает полученный номер со значениями в векторе прерывания и, таким образом, он узнает, какое именно устройство требует его внимания. Таблица содержит адреса памяти различных устройств ввода/вывода. Когда процессор выясняет, что внимания требует, к примеру, жесткий диск, он ищет в этой таблице соответствующий адрес памяти. Это новое значение счетчика команд, которое является начальным адресом, откуда процессор должен начинать чтение.

Одной из основных задач программного обеспечения операционной системы, управляющего работой устройств ввода/вывода, является обеспечение независимости его работы от конкретных устройств. Это означает, что написанное разработчиком приложение сможет читать (открывать файл) или записывать (сохранять файл) информацию практически на любое устройство (дискета, флеш-накопитель, жесткий диск, компакт-диск и т.д.). Этот уровень абстракции освобождает разработчиков приложений от необходимости писать различные процедуры для взаимодействия с различными устройствами ввода/вывода. Если бы разработчик должен был писать отдельную процедуру для записи на компакт-диск, отдельную процедуру для записи на дискету, отдельную процедуру для записи на жесткий диск и т. д., то каждый раз, когда появлялся бы новый вид устройств ввода/вывода, разработчику приходилось бы обновлять все свои приложения, чтобы они могли его поддерживать.

Существуют различные способы выполнения процедур ввода / вывода операционной системой. Мы рассмотрим следующие:

- Программируемый ввод/вывод
- Ввод/вывод, управляемый прерываниями
- Ввод/вывод с использованием DMA
- Ввод/вывод с частичным отображением
- Ввод/вывод с полным отображением

Программируемый ввод/вывод (programmable I/O). Если операционная система использует программируемый ввод/вывод, процессор посылает данные на устройство ввода/вывода, а затем опрашивает это устройство, чтобы узнать, когда оно будет готово принять новый блок данных. Пока устройство не готово к приему новых данных, процессор простаивает, ожидая пока устройство освободится. Это очень медленный метод работы, бесполезно расходующий драгоценное процессорное время. Поэтому умные люди придумали другой метод: ввод/вывод, управляемый прерываниями.

Ввод/вывод, управляемый прерываниями (interrupt-driven I/O). Если операционная система использует ввод/вывод, управляемый прерываниями, процессор отправляет данные на устройство и переключается на запросы других процессов. Когда устройство закончило обработку полученных данных и готово принять следующий блок данных, оно отправляет прерывание процессору. Процессор останавливает свою работу, отправляет устройству следующий блок данных, а затем снова переключается на другую работу. Это продолжается пока все данные не будут отправлены на устройство. Хотя при этом процессор не ждет выполнения каждой задачи устройством, он все равно расходует много времени на прерывания. Поэтому другие умные люди придумали ввод/вывод с использованием DMA.

Ввод/вывод с использованием DMA (I/O using DMA). Прямой доступ в память (DMA – Direct Memory Access) – это способ передачи данных между устройством ввода/вывода и системной памятью без использования процессора. Скорость передачи данных при этом намного больше. При использовании в операциях ввода/вывода, контроллер DMA передает данные на устройство ввода/вывода, не беспокоя для этого процессор. Этот метод иногда называют вводом/выводом без отображения (unmapped I/O).

Ввод/вывод с частичным отображением (remapped I/O). Ввод/вывод с частичным отображением и полным отображением (описанный ниже) не направлены на увеличение производительности, как предыдущий метод. Это два подхода, напрямую влияющие на безопасность. В системе, использующей ввод/вывод с частичным отображением, процессор отправляет физический адрес памяти запрашивающего процесса устройству ввода/вывода, а устройство ввода/вывода является достаточно доверенным для того, чтобы взаимодействовать с содержимым памяти напрямую. Операционная система доверяет устройству, надеясь, что оно будет вести себя правильно.

Ввод/вывод с полным отображением (fully mapped I/O). При использовании ввода/вывода с полным отображением операционная система не доверяет устройству ввода/вывода полностью – физический адрес устройству не передается. Вместо этого устройство работает с логическим адресом и в рамках контекста безопасности (от имени) запрашивающего процесса. Таким образом, операционная система не доверяет устройству прямое взаимодействие с памятью, она работает в качестве посредника для управления взаимодействием между процессом и устройством.

Ссылки по теме:

- Chapter 12, “I/O Management and Disk Scheduling,” by Joseph Kee-Yin

- Introduction to Operating Systems, by B. Ramamurthy (1/25/2002)

2. Архитектура системы

Проектирование операционной системы «с нуля» – это очень трудная задача со множеством запутанных и абстрактных целей, которые должны быть достигнуты с использованием математики, логики, дизайна, программирования и внедрения. Фундаментальные решения в отношении операционной системы должны быть приняты до начала фактического программирования. Безопасность – это только одна из целей операционной системы, но часто об этой цели беспокоятся только специалисты по безопасности.

Защитные меры, обеспечивающие доступность, целостность и конфиденциальность могут быть реализованы в различных местах. Например, компания может хранить информацию кредитных карт клиентов в базе данных, к которой имеют доступ многие пользователи. Эта информация явно требует защиты, обеспечивающей невозможность несанкционированного доступа и изменения. Мы должны начать с основополагающих, высокоуровневых вопросов, а затем двигаться вниз, в детали. Итак, где должна быть размещена защита? Следует применять средства управления доступом и присваивать права доступа пользователям на этапе их регистрации в системе? Следует ли обеспечивать защиту файлов данных, содержащих информацию кредитных карт, на уровне файловой системы? Следует ли обеспечивать защиту с помощью ограничения для пользователей возможных операций и действий? Или следует использовать комбинацию всего этого? Первый и наиболее глобальный вопрос – это «Где должна быть размещена защита: на стороне пользователей или в месте хранения данных, либо обеспечивать защиту путем ограничения действий пользователей в рамках среды?». Это проиллюстрировано на рисунке 3-14.

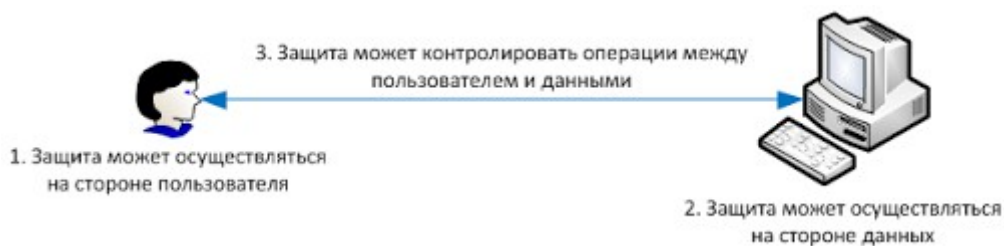


Рисунок 3-14. Безопасность может находиться в трех основных областях

Такие же вопросы следует задавать при построении операционной системы. Когда ответы на эти основополагающие вопросы будут получены, нужно учесть размещение механизмов. Механизмы безопасности могут быть размещены в аппаратных средствах, ядре, операционной системе, службах и на уровне приложений. На каком уровне (уровнях) следует реализовывать механизмы безопасности? Если защита реализована на аппаратном уровне, это обеспечивает широкий диапазон защиты, поскольку на нем базируется безопасность всех компонентов, работающих выше аппаратного уровня. Если механизмы защиты расположены в непосредственной близости от пользователя (на самом высоком уровне архитектуры операционной системы), эти механизмы являются более детальными и узконаправленными, чем механизмы, работающие на более низких уровнях архитектуры.

Как правило, более сложные механизмы защиты обеспечивают меньше гарантий, независимо от их местонахождения в архитектуре операционной системы. Это связано с тем, что более сложные механизмы требуют большей технической квалификации от специалистов, которые их устанавливают, тестируют, поддерживают и эксплуатируют. Более сложные механизмы безопасности сложнее протестировать полностью и во всех возможных условиях. С другой стороны, простые механизмы не всегда могут предоставить богатую функциональность и гибкие настройки, хотя их проще установить, поддерживать, эксплуатировать и тестировать. Таким образом, необходимо хорошо понимать компромисс между функциональностью и гарантиями, чтобы правильно выбрать механизмы безопасности при проектировании

системы.

Когда у проектировщики системы поняли, на чем должны фокусироваться механизмы безопасности (пользователи, операции или данные), на каком уровне должны быть размещены эти механизмы (на аппаратном уровне, в ядре, на уровне операционной системы, служб или на уровне приложений), насколько сложен должен быть каждый механизм, им нужно построить и интегрировать эти механизмы таким образом, чтобы они правильно взаимодействовали с другими частями системы.

Сначала команда проектировщиков должна решить, какие системные механизмы будут считаться доверенными и будут размещены на кольце 0. Затем они должны определить, как эти различные части кода будут безопасно взаимодействовать между собой. Хотя, казалось бы, предпочтительно было бы доверять всем компонентам системы, это может быть крайне сложно реализовать, кроме того это может стать «узким местом» с точки зрения производительности. Чтобы быть доверенным, механизм должен работать предсказуемым и безопасным образом, не оказывая неблагоприятного воздействия на другие доверенные и недоверенные механизмы. Доверенные компоненты имеют доступ к привилегированным службам и прямой доступ в память, обычно они имеют высокий приоритет при запросе компьютерного времени, а также обладают большим контролем над системными ресурсами. Поэтому доверенные субъекты и объекты должны быть надлежащим образом идентифицированы, отделены от недоверенных и размещены в определенных подмножествах (subset).

2.1. Определенные подмножества субъектов и объектов

Как было сказано ранее, не все компоненты должны быть доверенными и поэтому не все компоненты попадают в рамки *доверенной компьютерной базы* (TCB – Trusted Computing Base). TCB определяется как сочетание всех защитных механизмов в рамках компьютерной системы. TCB включает в себя аппаратное и программное обеспечение, а также прошивки (firmware), поскольку все эти компоненты реализуют политику безопасности и не должны нарушать ее.

Компоненты, которые попадают в рамки TCB, должны быть идентифицированы, должны быть определены их стандартные возможности. Например, система, которая не требует высокого уровня доверия, может разрешить всем аутентифицированным пользователям доступ с правом внесения изменений в любые файлы на компьютере. Это широкое подмножество субъектов и объектов, которые свободно взаимодействуют внутри него. Однако система, от которой требуется высокий уровень доверия, может позволить, например, только двум субъектам иметь доступ на чтение ко всем файлам на компьютере, и только одному из них – доступ на изменение любых файлов. Такое подмножество будет гораздо меньше, а правила взаимодействия внутри него будут более строгими и детальными.

Архитектура безопасности – это один из компонентов архитектуры продукта в целом, она предоставляет правила, используемые при проектировании продукта. Архитектура безопасности должна описывать требуемый уровень гарантий и потенциальные воздействия различных этапов разработки продукта на его безопасность. Поскольку проект разработки программного обеспечения переходит от архитектуры к проектированию, затем к техническим требованиям, а затем к разработке кода, архитектура безопасности и требования безопасности становятся более детальными на каждом шаге.

Если разработчики хотят разработать систему, соответствующую рейтингу D (очень низкий) классификации уровня гарантий Оранжевой книги, в рамки TCB попадет немного компонентов, т.к. от системы не ожидается очень высокого уровня безопасности. Однако если разработчики хотят разработать систему с рейтингом по Оранжевой книге B1 или B2 (гораздо более высокий, чем D), в рамки TCB попадет гораздо больше компонентов, а разработчики должны будут обеспечить правильное выполнение своих задач всеми этими компонентами. Эти компоненты TCB должны реализовывать жесткие правила,

определяющие порядок взаимодействия субъектов и объектов. Разработчикам также нужно убедиться, что все эти компоненты идентифицированы, проконтролированы и работают предсказуемым образом, поскольку все они будут тщательно исследоваться и тестироваться в процессе оценки, прежде чем продукту будет дан рейтинг B2 или B1. (Оранжевая книга будет рассматриваться далее в этом Домене).

2.2. Доверенная компьютерная база

Термин «доверенная компьютерная база» (ТСВ), появившийся в Оранжевой Книге, не учитывает обеспечиваемый системой уровень безопасности, он учитывает уровень доверия к системе с точки зрения ее безопасности. Это связано с невозможностью защитить компьютерную систему полностью. Со временем меняются, развиваются и появляются новые виды атак и уязвимостей, поэтому при достаточном количестве времени и объеме ресурсов атака почти на любую систему будет успешной. Однако если система удовлетворяет определенным критериям, считается, что она обеспечивает определенный уровень доверия и будет реагировать предсказуемо в различных ситуациях.

ТСВ учитывает не только компоненты операционной системы, поскольку компьютер состоит не только из операционной системы. ТСВ учитывает аппаратное обеспечение, программное обеспечение и прошивки. Каждый из этих компонентов может оказать положительное или отрицательное воздействие на компьютерную среду, каждый из них обеспечивает поддержку и реализацию политики безопасности соответствующей системы. Некоторые компоненты и механизмы напрямую отвечают за поддержку политики безопасности (например, прошивка, которая не позволяет пользователю загрузиться с дискеты, или менеджер памяти, не позволяющий одному процессу напрямую записывать информацию в память другого процесса). Существуют и другие компоненты, которые не реализуют политику безопасности, но они должны соответствовать ей, чтобы не нарушить доверие к системе. Примерами способов нарушения компонентом политики безопасности системы могут быть попытки приложений напрямую обратиться к аппаратному устройству вместо использования соответствующего системного вызова с помощью механизмов операционной системы; попытки программ обратиться к памяти за пределами выделенного им пространства памяти; неправильное освобождение ресурсов программами после окончания их использования.

ТСВ для многих кажется абстрактной и непонятной концепцией, а множество изданных по этой теме книг и документов не упрощают понимание этого термина. Если операционная система использует ТСВ, это означает, что она имеет защищенное ядро, отделяющее системные процессы. Ядро состоит из аппаратного обеспечения, программного обеспечения и прошивок – оно, по сути, и является ТСВ. Но в рамки ТСВ могут быть включены и другие компоненты, такие как доверенные команды, программы, конфигурационные файлы, напрямую взаимодействующие с ядром. Например, при установке системы Unix администратор может выбрать установку ТСВ, в результате чего будет установлен доверенный канал, доверенная оболочка, а также средства контроля целостности системы.

Доверенный канал (trusted path) – это коммуникационный канал взаимодействия между пользователями (или программами) и ядром. ТСВ обеспечивает защиту ресурсов, гарантируя невозможность компрометации этого канала. **Доверенная оболочка** (trusted shell) гарантирует, что работающий в ней пользователь не сможет выйти за ее пределы, и никто другой (никакой другой процесс) не попадет внутрь нее.

Некоторые команды Unix являются частью ТСВ, например, `setuid` (установка ID процесса) и `setgid` (установка группового ID процесса). Только привилегированный пользователь (такой как `root`) может менять информацию об ID процесса.

Ранние операционные системы (например, MS-DOS, Windows 3.11, Novell Netware версии 3 и т.п.) не имели ТСВ. Windows 95 имел ТСВ, но он мог использоваться только при работе в

32-битном режиме. Windows NT был первой версией Windows, в которой реально была реализована идея TCB. Microsoft часто использует слова «Доверенные вычисления» (Trustworthy Computing), но это концепция не принадлежит Microsoft. Многие производители разрабатывают все лучшие и лучшие TCB, создавая новые и совершенствуя существующие методы защиты программного обеспечения от компрометации. Просто Microsoft является первым производителем, внедрившим эти методы в свои продукты Windows 2003. В основном Microsoft строит свои продукты на основе текущей TCB, которую она назвала «Безопасной компьютерной базой нового поколения» (NGSCB – Next-Generation Secure Computing Base). Microsoft переименовала свое ядро безопасности, назвав его «пехус». Организация работы NGSCB выходит за рамки данной книги и экзамена CISSP, в основном операционная система просто делает то, что от нее ожидается, улучшает изоляцию между доверенными и недоверенными процессами, улучшает управление памятью, лучше защищает операции ввода/вывода, улучшает программные механизмы аутентификации.

ПРИМЕЧАНИЕ. Чтобы понять ядро Microsoft Windows 7, посетите страницу <http://channel9.msdn.com/shows/Going+Deep/Mark-Russinovich-Inside-Windows-7/>

Каждая операционная система имеет ряд компонентов, компрометация которых может подвергнуть систему серьезной опасности. TCB предоставляет дополнительные уровни защиты вокруг таких компонентов, чтобы гарантировать невозможность их компрометации и обеспечить постоянную работу системы безопасным и предсказуемым образом.

Как TCB делает все это? Процессы в рамках TCB являются компонентами, защищающими систему в целом. Поэтому разработчики операционной системы должны позаботиться, чтобы эти процессы имели собственный **домен выполнения**. Это означает, что они размещаются на кольце 0, их команды выполняются в защищенном режиме, а менее доверенные процессы не могут напрямую взаимодействовать с ними. Разработчики должны обеспечить поддержку операционной системой изолированного домена выполнения, чтобы эти процессы не могли быть скомпрометированы или взломаны. Ресурсы, используемые процессами TCB, должны быть также изолированы, для чего должно быть реализовано строгое управление доступом и все запросы на доступ и операции с ресурсами должны надлежащим образом контролироваться.

Четырьмя основными функциями TCB являются: активация процессов, переключение доменов выполнения, защита памяти и операции ввода/вывода. Мы уже рассматривали все эти функции в предыдущих разделах без использования точной терминологии, поэтому подведем здесь итог. **Активация процесса** относится к деятельности, выполняющейся в тот момент, когда процессу нужно передать на обработку процессору свои команды и данные. Как было сказано ранее, процессор заполняет свои регистры информацией запрашивающего процесса (счетчик команд, базовый адрес и адрес границы области памяти, реальный или защищенный режим и т.д.). Процесс «активирован», когда вызвано его прерывание, позволяющее ему взаимодействовать с процессором. Процесс «деактивирован», когда его команды полностью выполнены процессором или когда другой процесс с более высоким приоритетом обратился к процессору. После деактивации процесса, регистры процессора должны быть заполнены новой информацией, относящейся к новому запрашивающему процессу. Данные, которые загружаются в регистры процессора и выгружаются из них, имеют критическое значение, поэтому компоненты TCB должны гарантировать их безопасность.

Переключение домена выполнения производится, когда процессу нужно вызвать другой, более доверенный, процесс, находящийся на более низком кольце защиты. Как объяснялось ранее, менее доверенные процессы работают в реальном режиме и не могут выполнять действия, такие как взаимодействие с аппаратными устройствами или прямая отправка запросов ядру. Процесс, работающий в реальном режиме (кольцо 3) должен делать запрос к службе операционной системы, которая работает на кольце 1. Информация менее доверенного процесса будет загружена в регистры процессора, и затем, когда процессор

увидит, что была вызвана служба операционной системы, он переключит домены и контекст безопасности. Соответственно, информация процесса службы операционной системы будет загружена в регистры процессора, и процессор начнет выполнять ее команды в защищенном режиме. Таким образом, переключение домена выполнения – это, по сути, переключение процессора из выполнения команд в реальном режиме в защищенный режим и обратно. Все это должно происходить надлежащим образом, иначе менее доверенный процесс сможет выполняться в защищенном режиме и иметь прямой доступ к системным ресурсам.

Защита памяти и операции ввода/вывода обсуждались в предыдущих разделах, поэтому просто укажем, что за эти операции отвечают компоненты, находящиеся в рамках ТСВ. Защита достигается путем разделения этой деятельности на отдельные модули, фактически являющиеся процессами, из которых состоит ядро. При этом даже компрометация одного из процессов ядра не означает, что атакующий получит контроль над всеми процессами.

Не каждая часть системы должна быть доверенной. Частью процесса оценки уровня доверия к системе является определение ее архитектуры, служб безопасности и механизмов обеспечения гарантий, которые составляют ТСВ. В процессе оценки проводится тестирование, которое должно показать, насколько ТСВ защищен от умышленного и неумышленного воздействия и попыток компрометации. Чтобы достичь высокого рейтинга уровня доверия, системы должны удовлетворять строгим и четким требованиям ТСВ, а детали их рабочих состояний, этапов разработки, процедур тестирования и документация должны быть проанализированы более детально, чем при оценке систем, пытающихся достичь менее высокого рейтинга уровня доверия.

Использование определенных критериев безопасности позволяет встроить доверие в систему, оценить его и подтвердить. Такой подход дает клиенту метрики, позволяющие ему сравнить один продукт с другим. А производителям это дает четкие инструкции, говорящее им о том, что ожидается от их систем для присвоения им того или иного рейтинга уровня гарантий. Так, если кто-то говорит про систему с рейтингом C2, все остальные понимают, что конкретно под этим подразумевается.

Оранжевая книга – это один из критериев оценки. Она определяет доверенную систему как аппаратное и программное обеспечение, которое в определенной мере защищает классифицированные и неклассифицированные данные для определенного круга пользователей без нарушения прав доступа и политики безопасности. Она рассматривает все защитные механизмы системы, которые реализуют политику безопасности, и обеспечивает предсказуемое поведение среды. Это означает, что каждый следующий уровень системы должен доверять нижестоящим уровням, быть уверенным, что они будут выполнять ожидаемые функции, обеспечивать ожидаемый уровень защиты и работать предсказуемым образом во множестве различных ситуаций. Когда операционная система обращается к аппаратному устройству, она ожидает, что данные будут возвращены им в определенном формате, будут непротиворечивы и предсказуемы. Приложение, которое работает над операционной системой, ожидает наличия возможности делать системные вызовы, получать запрошенные данные, работать в надежной среде. Пользователи ожидают, что аппаратное обеспечение, операционная система и приложения работают одинаковым образом и предоставляют определенный уровень функциональности. Для того, чтобы все эти действия выполнялись предсказуемым образом, требования к системе должны быть учтены на этапе планирования, а не позднее.

2.3. Периметр безопасности

Как было сказано ранее, не каждый процесс и ресурс попадает в рамки ТСВ, поэтому некоторые из этих компонентов находятся вне воображаемых границ, называемых периметром безопасности. *Периметр безопасности* (security perimeter) – это границы, которые отделяют доверенное от недоверенного. Чтобы система оставалась в безопасном и доверенном состоянии, должны быть разработаны четкие стандарты взаимодействия,

гарантирующие, что любое взаимодействие компонента ТСВ с компонентом вне ТСВ не может привести к неожиданному нарушению безопасности системы. Такое взаимодействие производится и управляется посредством интерфейсов.

Например, ресурс, находящийся в рамках границ ТСВ или периметра безопасности, не должен позволить менее доверенным компонентам использовать критичные системные ресурсы. Процесс в рамках ТСВ также должен заботиться о командах и информации, которые он принимает от менее доверенных ресурсов. Эти ограничения встраиваются в интерфейсы, которые обеспечивают такой вид взаимодействия и являются механизмами, реализующими периметр безопасности. Взаимодействие между доверенными и недоверенными компонентами должно контролироваться для гарантий сохранения системы в стабильном и безопасном состоянии.

ПРИМЕЧАНИЕ. ТСВ и периметр безопасности являются не физическими сущностями, а концептуальными конструкциями, используемыми разработчиками систем для разделения доверенных и недоверенных компонентов.

2.4. Монитор обращений и ядро безопасности

Итак, к данному моменту наши разработчики архитектуры компьютерной системы сделали многие вещи. Они определили размещение механизмов безопасности (аппаратное обеспечение, ядро, операционная система, службы или программы), процессы, находящиеся в рамках ТСВ, а также взаимодействие механизмов безопасности и процессов друг с другом. Они определили периметр безопасности, который разделяет доверенные и недоверенные компоненты. Они разработали интерфейсы для безопасного взаимодействия всех этих сущностей. Теперь им нужно разработать и внедрить механизм, гарантирующий, что субъекты, которые обращаются к объектам, имеют необходимые для этого права. Это означает, что разработчики должны разработать монитор обращений и ядро безопасности.

Монитор обращений (reference monitor) – это абстрактная машина, являющаяся посредником, через которого проходят все обращения субъектов к объектам. Монитор обращений проверяет, что субъекты имеют необходимые права доступа, защищая объекты от несанкционированного доступа и изменения. Чтобы система достигла высокого уровня доверия, необходимо, чтобы субъекты (программы, пользователи и процессы) были полностью авторизованы перед их доступом к объектам (файлам, программам и ресурсам). Субъекту не должно быть позволено использовать запрошенный ресурс, пока ему не будут предоставлены соответствующие привилегии доступа. Монитор обращений – это концепция управления доступом, а не реальный физический компонент, поэтому его часто называют «концепцией монитора обращений» или «абстрактной машиной».

Ядро безопасности (security kernel) состоит из компонентов аппаратного обеспечения, программного обеспечения и прошивок, попадающих в рамки ТСВ и реализующих концепцию монитора обращений. Ядро безопасности осуществляет посредничество при доступе и работе субъектов с объектами. Ядро безопасности – это ядро ТСВ, это наиболее часто используемый подход к построению доверенных компьютерных систем (trusted computing system). Есть три основных требования к ядру безопасности:

- Оно должно обеспечивать изоляцию для процессов, реализующих концепцию монитора обращений, и процессов, которые должны быть защищены от внешних воздействий.
- Оно должно вызываться при каждой попытке доступа, не должно существовать возможностей обойти его.
- Оно должно быть полностью и всесторонне протестировано и проверено.

Это является требованиями монитора обращений и, соответственно, требованиями компонентов, обеспечивающих и реализующих концепцию монитора обращений – ядра

безопасности.

Хотя эти вопросы и являются абстрактными, они реализуются в реальном мире аппаратных устройств и программного кода. Компоненты, реализуя абстрактную идею монитора обращений, обеспечивают гарантии посредством тестирования и функциональности.

ПРИМЕЧАНИЕ. Монитор обращений – это концепция абстрактной машины, являющейся посредником во всех операциях доступа субъектов к объектам. Ядро безопасности – это аппаратное обеспечение, прошивки и программное обеспечение ТСВ, реализующее эту концепцию. ТСВ – это совокупность защитных механизмов компьютера, которые работают совместно для реализации политики безопасности. ТСВ включает в себя ядро безопасности и все остальные защитные механизмы.

Приведем аналогию, показывающую взаимоотношения между процессами ядра, самим ядром и концептуальным монитором обращений. Общество состоит из людей. Представим, что люди – это процессы, а общество – ядро. Члены общества должны общаться между собой определенным образом, чтобы иметь определенные стандарты жизни. Для этого существуют законы. Законы – это монитор обращений, они обеспечивают правильность деятельности. Ожидается, что каждый человек останется в рамках законов и будет действовать определенным образом, чтобы не оказать неблагоприятного воздействия на общество в целом и не поставить под угрозу стандарты жизни. Компоненты в рамках системы должны оставаться в рамках законов монитора обращений, чтобы они не оказали неблагоприятного влияния на другие компоненты и не поставили под угрозу безопасность всей системы.

Ссылки по теме:

- “Rationale Behind the Evaluation Classes”
- The Reference Monitor Concept
- Implementing Complete Mediation

2.5. Политика безопасности

Как было указано ранее, ТСВ состоит из компонентов, непосредственно реализующих политику безопасности. Но что такое политика безопасности? Политика безопасности – это набор правил и практик, диктующих порядок управления критичной информацией и ресурсами, их защиты и распространения. Политика безопасности точно определяет целевой уровень безопасности и указывает, какие механизмы безопасности предполагается реализовать для его достижения. Это важный элемент, который играет ключевую роль при определении структуры системы. Политика безопасности – это основа для технических требований к системе, она предоставляет базис для оценки системы.

Домен 01 глубоко рассматривает политики безопасности, но эти политики задают направление для всей компании. Политики безопасности, рассматриваемые в настоящем Домене, относятся к операционным системам, устройствам и приложениям. Хотя эти различные политики похожи, они имеют различные цели: цели компании в первом случае и цели отдельной компьютерной системы – во втором.

Система обеспечивает доверие, выполняя и реализуя политику безопасности, контролируя взаимодействие между субъектами и объектами. Политика должна указывать, какие субъекты могут иметь доступ к определенным объектам, какие действия являются приемлемыми, а какие – нет. Политика безопасности дает основу для архитектуры безопасности системы.

Для обеспечения приемлемого уровня доверия, система должна быть основана на архитектуре, способной защитить себя от недоверенных процессов, случайных и преднамеренных нарушений безопасности, атак на разных уровнях системы. Большинство

рейтингов доверия, определенных посредством проведения формальных оценок, требуют определения подмножеств субъектов и объектов, явно заданных доменов, изоляции процессов, а также управления их доступом и ведения аудита их действий.

Итак, мы знаем, что доверие к системе определяется тем, как она реализует свою собственную политику безопасности. Когда система протестирована на соответствие определенным критериям, системе присваивается рейтинг, и в дальнейшем этот рейтинг используется покупателями, производителями и компьютерным сообществом в целом. Критерии определяют, обеспечена ли должным образом реализация и поддержка политики безопасности. Политика безопасности основана на правилах и практиках, относящихся к тому, как осуществляется управление системой, ее защита и разрешение доступа к критичным ресурсам. Монитор обращений – это концепция, которая говорит, что все субъекты должны быть надлежащим образом авторизованы для доступа к объектам, эта концепция реализуется ядром безопасности. Ядро безопасности включает в себя все ресурсы, которые контролируют деятельность системы в соответствии с ее политикой безопасности и являются частью операционной системы, управляющей доступом к системным ресурсам. Чтобы ядро безопасности работало правильно, отдельные процессы должны быть изолированы друг от друга, а домены должны быть определены для указания того, какие объекты каким субъектам доступны.

Политики безопасности, которые предотвращают утечку информации с высокого уровня безопасности на более низкий, называются **многоуровневыми политиками безопасности** (multilevel security policy). Политики такого типа позволяют субъекту получить доступ к объекту, только если уровень безопасности субъекта выше или равен классификации объекта.

Как было указано ранее, рассмотренные в предыдущих разделах концепции являются абстрактными идеями, которые используются физическими аппаратными компонентами, прошивками, кодом программного обеспечения, а также в процессе деятельности на этапе проектирования, построения и реализации системы. Эти идеи похожи на абстрактные цели и мечты, которые неплохо бы реализовать, путем сложной физической работы и дисциплины.

2.6. Принцип наименьших привилегий

После обеспечения надлежащей изоляции процессов и ресурсов, необходимо внедрить **принцип наименьших привилегий** (least privilege). Он означает, что процессы должны иметь не больше привилегий, чем им необходимо для выполнения своих функций. Только процессам, которым необходимо выполнять критичные системные функции, должен быть разрешен доступ к критичным системным ресурсам, а другим, менее привилегированным процессам, при необходимости выполнения таких действий следует обращаться к более привилегированным процессам. Такой вид непрямого взаимодействия защищает систему от плохо написанного и неправильно работающего кода. Процессы должны владеть повышенным уровнем привилегий только в течение того времени, пока действительно существует такая необходимость. Например, если процессу нужно повысить свой статус, чтобы напрямую взаимодействовать с системными ресурсами, то, как только эта задача будет выполнена, статус процесса должен быть обратно понижен, чтобы другие механизмы не могли воспользоваться им для оказания негативного воздействия на систему. В ядре размещаются только процессы, которым необходим полный набор системных привилегий, другие, менее привилегированные процессы, обращаются к ним для выполнения критичных или деликатных операций.

Приведем пример управления доступом на основе принципа наименьших привилегий. Программа резервного копирования должна иметь доступ к файлам только на чтение, доступ на изменение этих файлов ей предоставляться не должен. Аналогично, программе восстановления должна быть позволена только запись в файлы на диске, но не их чтение.

3. Модели безопасности

Важной концепцией в проектировании и анализе безопасных систем является модель безопасности, поскольку она включает в себя политику безопасности, которая должна быть реализована в системе. Модель – это символическое представление политики. Она преобразует желания создателей политики в набор правил, которым должна следовать компьютерная система.

В этом Домене часто упоминается политика безопасности и ее важность. Это связано с тем, что политика является абстрактным понятием, представляющим цели и задачи, которые должна соблюдать и реализовывать система для того, чтобы считаться безопасной и приемлемой. Как нам перейти от абстрактной политики безопасности к моменту, когда администратор запрещает Дэвиду доступ к конфигурационным файлам системы, снимая отметку с соответствующего пункта в графическом интерфейсе? Этот путь состоит из множества сложных шагов, предпринимаемых на протяжении проектирования и разработки системы.

Модель безопасности преобразует абстрактные цели политики в термины информационных систем, точно описывая структуры данных и средства (методы), необходимые для реализации политики безопасности. Модель безопасности обычно представляется в виде математических и аналитических идей, которые затем преобразуются в технические требования к системе, а затем разрабатывается программистами в коде программы. Таким образом, мы имеем политику, реализующую цели безопасности, типа «каждый субъект должен быть авторизован для доступа к каждому объекту». Модель безопасности берет эти требования и предоставляет необходимые математические формулы, взаимоотношения и структуру, которым необходимо следовать для достижения целей безопасности. Исходя из этого, разработаны технические требования для каждого типа операционной системы (Unix, Windows, Macintosh и т.д.), и отдельные производители могут решать, как им реализовывать механизмы, которые будут соблюдать эти технические требования.

Приведем очень общий и упрощенный пример. Если политика безопасности утверждает, что субъекты должны быть авторизованы для доступа к объектам, модель безопасности должна предоставить математические взаимоотношения и формулы, объясняющие, как x может получить доступ к y только посредством определенных и описанных методов. Затем разрабатываются технические требования, являющиеся мостом между тем, что это означает в компьютерной среде и тем, как это связано с компонентами и механизмами, которые должны быть разработаны. После этого разработчики пишут программный код для реализации механизмов, позволяющих использовать ACL и предоставляющих администраторам определенную степень управления. Эти механизмы представляют сетевому администратору графический интерфейс, в котором он может настроить разграничение доступа в рамках операционной системы, выбирая, какие субъекты могут иметь доступ к каким объектам. Это элементарный пример, поскольку модель безопасности может быть очень сложной. Этот пример используется для демонстрации взаимоотношений между политикой безопасности и моделью безопасности.

Некоторые модели безопасности, такие как Bell-LaPadula, реализуют правила, обеспечивающие защиту конфиденциальности. Другие модели, как, например, Biba, реализуют правила, обеспечивающие защиту целостности. Формальные модели безопасности, такие как Bell-LaPadula и Biba, используются для предоставления высокого уровня гарантий безопасности. Неформальные модели, такие как Clark-Wilson, больше используются в качестве основы для описания того, как политики безопасности должны выражаться и выполняться.

Политика безопасности описывает цели без конкретизации того, как они должны быть достигнуты. Модель – это платформа, которая придает политике форму и решает проблемы безопасного доступа к информации в конкретных ситуациях. Многие модели безопасности

разработаны для реализации политик безопасности. Следующие разделы предоставляют обзор каждой из моделей.

Взаимоотношения между Политикой безопасности и Моделью безопасности. Если кто-то говорит вам, что нужно вести правильный и здоровый образ жизни, это очень широкое, общее, абстрактное понятие. Когда вы спрашиваете этого человека – как это сделать, он описывает вам, что вам нужно делать и что не нужно (не наносить другим вреда, не говорить неправду, есть овощи, чистить зубы и т.д.). Аналогично, Политика безопасности предоставляет абстрактные цели, а Модель безопасности говорит, что нужно и что не нужно делать, чтобы достичь эти цели.

3.1. Модель конечных автоматов

В *моделях конечных автоматов* (state machine model) для проверки безопасности системы используется состояние (state), которое является мгновенным «снимком» всех текущих разрешений и всех текущих экземпляров субъектов, использующихся объектами. Каждая связь субъекта с объектом имеет отношение к поддержанию состояния системы. Система безопасна, если субъекты могут получить доступ к объектам только с помощью средств, соответствующих политике безопасности. Конечные автоматы предоставляют основу для важнейших моделей безопасности. Состояние системы – это ее моментальный снимок. Множество различных действий могут изменить состояние системы. Это называется **переходами состояния** (state transition). Разработчики операционной системы при реализации модели конечных автоматов должны учесть все возможные переходы состояний и оценить, не могут ли они привести систему в небезопасное состояние. Если все действия, которые могут произойти в системе, не приводят ее в небезопасное состояние, это означает, что система выполняется в безопасной модели конечных автоматов.

Формальные модели. Использование моделей при разработке программного обеспечения не стало так популярно, как предполагалось. Это связано в первую очередь с тем, что производители всегда старались вывести продукт на рынок как можно быстрее. А при использовании формальных моделей больше времени занимает архитектурная фаза разработки, что разработчики часто не могут себе позволить. В обязательном порядке формальные модели используются при разработке систем, для которых недопустимы ошибки или проблемы безопасности, таких, как системы управления движением воздушного транспорта, программное обеспечение космических кораблей, железнодорожные сигнальные системы, военные классифицированные системы и медицинские системы управления. Это не означает, что эти модели или их части не используются в коммерческих продуктах. Просто производители коммерческих систем не всегда в полной мере следуют этим моделям.

Модели конечных автоматов используются для описания поведения системы при различных входных данных. Это дает математические конструкции, которые представляют собой множества (субъектов и объектов), и последовательности. Когда объект принимает входные данные, это изменяет состояние переменной. Упрощенный пример состояния переменной – это (Имя, Значение), как показано на Рисунке 3-15. Эта переменная является частью набора команд операционной системы. Когда эта переменная вызывается для использования, в нее может быть записано (Цвет, Красный) из входных данных, введенных пользователем программы. Скажем, пользователь вводит другое значение и теперь в переменную записывается (Цвет, Синий). Это упрощенный пример перехода состояния. Некоторые переходы состояний также являются простыми. Сложность появляется, когда системе нужно решать, следует ли разрешать тот или иной переход. Для принятия решения о возможности перехода, операционной системой должны быть проанализированы атрибуты безопасности объекта и права доступа субъекта.

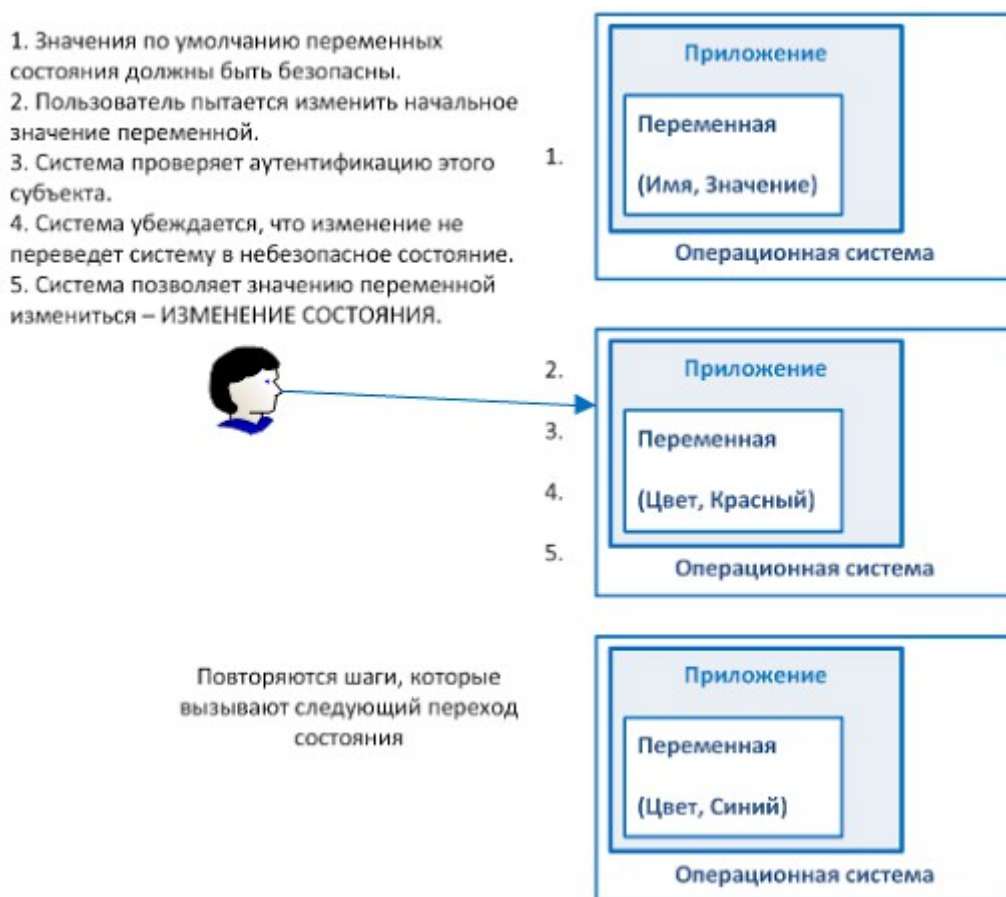


Рисунок 3-15. Упрощенный пример изменения состояния

Разработчики, которые реализуют модель конечных автоматов, должны идентифицировать все начальные состояния (значения переменных по умолчанию) и описать, как эти значения могут быть изменены (входные данные, которые будут приняты), чтобы любые конечные состояния (результатирующие значения) сохраняли уверенность в том, что система остается безопасной. Описание процесса возможных изменений этих значений часто реализуется посредством условных утверждений, типа «ЕСЛИ условие ТОГДА обновление».

Система, использующая модель конечных автоматов, будет оставаться в безопасном состоянии в любой момент. Она будет загружаться в безопасном состоянии, безопасно выполнять команды и транзакции, позволит субъектам использовать ресурсы только в безопасном состоянии, будет сохранять безопасное состояние при выключении и в случае сбоя. Сохранение безопасного состояния в случае сбоя крайне важно – система должна быть в состоянии «сохранить себя» и не сделать себя уязвимой. Когда операционная система показывает пользователю сообщение об ошибке, перезагружается или зависает, это является следствием выполнения защитных мер. Если операционная система считает, что происходит что-то недопустимое, но при этом она сама не может противостоять этому, она не позволяет перевести себя в небезопасное состояние и реагирует подобным образом. Поэтому, если приложение или система зависла, знайте, что система просто попыталась защитить себя и ваши данные.

Многие вопросы должны быть учтены при разработке продукта, использующего модель конечных автоматов. Изначально разработчик должен определить, что и где является переменными состояния. В компьютерной среде все переменные с данными могут считаться независимыми переменными состояния, а неправильные их изменения предположительно могут изменить или нарушить работу системы или отдельных процессов. Затем разработчик должен определить безопасное состояние для каждой переменной состояния. После этого нужно определить все приемлемые функции перехода состояний. Эти функции должны

описывать допустимые изменения, которые могут быть произведены в отношении переменных состояния.

После того, как все функции перехода будут определены, они должны быть протестированы, чтобы убедиться, что конечный автомат в целом не будет скомпрометирован и что эти функции переходов сохраняют целостность системы (компьютера, данных, программы или процесса) в течение всего времени.

3.2. Модель Bell-LaPadula

В 1970-е годы американские военные использовали мейнфреймы, работающие в режиме разделения времени, и были заинтересованы в безопасности этих систем и защите от утечки классифицированной информации. **Модель Bell-LaPadula** разработана для таких систем. Это была первая математическая модель многоуровневой политики безопасности, использованная для определения понятия безопасного конечного автомата, режимов доступа и описания правил доступа. Ее разработка была профинансирована американским правительством, для обеспечения платформы для компьютерных систем, используемых для хранения и обработки критичной информации. Основной целью этой модели было предотвращение несанкционированного доступа к секретной информации.

Системы, использующие модель Bell-LaPadula, называют *многоуровневыми системами безопасности*, т.к. систему используют пользователи с разным допуском, а система обрабатывает данные с различной классификацией. Уровень классификации информации определяет, какие процедуры могут быть использованы для ее обработки. Модель Bell-LaPadula – это модель конечных автоматов, которая реализует аспект *конфиденциальности* при управлении доступом. Для принятия решения о доступе субъекта к объекту используются матрицы и уровни безопасности. Допуск субъекта сравнивается с классификацией объекта, затем применяются специальные правила для определения возможных вариантов взаимодействия субъекта и объекта.

Эта модель использует субъекты, объекты, операции доступа (чтение, запись и чтение/запись) и уровни безопасности. Субъекты и объекты могут находиться на различных уровнях безопасности, при этом будут использоваться взаимоотношения и правила, указывающие приемлемые действия между ними. Эта модель, при надлежащей реализации и внедрении, математически доказывает обеспечение очень безопасной и эффективной операционной системы. Она также является моделью безопасности информационных потоков, что означает невозможность передачи информации небезопасным способом.

Модель Bell-LaPadula – это модель субъект-объект (например, как пользователь (субъект) может читать элемент данных (объект) из определенной базы данных и записывать данные в эту базу данных). Модель Bell-LaPadula направлена на обеспечение гарантий того, что субъекты надлежащим образом аутентифицированы, имеют необходимый допуск безопасности, категорию «должен знать» и формальное разрешение доступа перед непосредственным доступом к объекту.

Есть три основных правила, используемых и реализованных в модели Bell-LaPadula: простое правило безопасности, правило *-свойства и строгое правило *-свойства. **Простое правило безопасности** (simple security rule) говорит о том, что субъект не может читать данные, находящиеся на более высоком уровне безопасности, чем его допуск. Например, если Боб имеет допуск безопасности «секретно», это правило говорит, что он не может читать данные, классифицированные как «совершенно секретно». Если компания хочет, чтобы Боб имел возможность читать совершенно секретные данные, она должна сначала дать Бобу соответствующий допуск.

Правило *-свойства (*-property rule, star property rule) говорит о том, что субъект не может записывать данные на меньший уровень безопасности, чем его допуск. Простое правило безопасности называют правилом «не читать сверху» (no read up), а правило *-свойства –

правилом «не записывать вниз» (no write down), как показано на Рисунке 3-16. Третье правило – **строгое правило *-свойства** (strong *-property rule) говорит о том, что субъект может выполнять функции чтения и записи только на том же уровне безопасности, что и его допуск – не выше и не ниже. Таким образом, чтобы субъект имел возможность чтения и/или записи объекта, его уровень допуска должен совпадать с классификацией объекта.

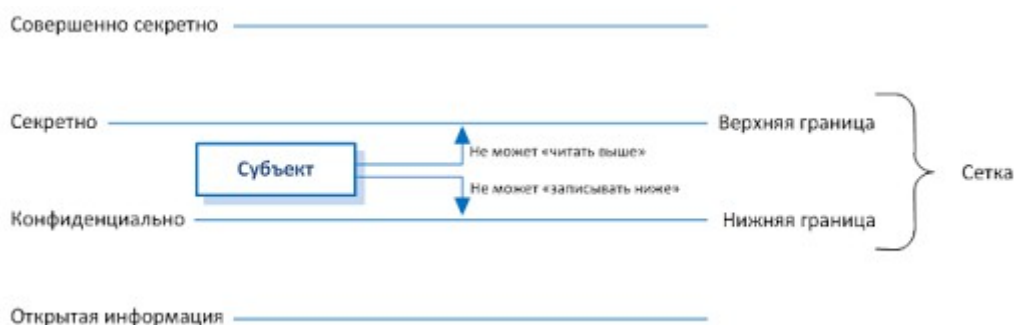


Рисунок 3-16. В модели Bell-LaPadula каждый субъект имеет сетку прав

Эти три правила указывают, в какие состояния может перейти система. Помните, что состояние – это мгновенный снимок значений переменных в программной системе. Если субъект выполняет операцию чтения объекта, находящегося на более низком уровне безопасности, этот субъект теперь имеет переменную, в которую записаны прочитанные данные. Если субъект записал данные в объект, находящийся на более высоком уровне безопасности, субъект изменил переменную в рамках домена объекта.

ПРИМЕЧАНИЕ. В терминах управления доступом, слово «доминировать» (dominate) означает быть выше или равным. Таким образом, если вы видите утверждение, например, «субъект может выполнять операцию чтения только в том случае, если его класс доступа доминирует над классом доступа объекта», это означает просто, что субъект должен иметь допуск выше или равный классификации объекта. В модели Bell-LaPadula это называют отношением доминирования, являющимся взаимоотношением допуска субъекта и классификации объекта.

Состояние системы изменяется по мере выполнения различных операций. Модель Bell-LaPadula определяет безопасное состояние, означающее безопасную вычислительную среду и допустимые действия, являющиеся операциями, не нарушающими безопасное состояние системы. Это означает, что эта модель обеспечивает безопасное состояние и разрешает только те операции, которые сохраняют систему в безопасном состоянии и не дают ей перейти в небезопасное состояние. Так, если 100 человек ежедневно с помощью одной этой системы использует 2000 объектов, эта система выполняет достаточно большой объем сложной работы. Однако в конце дня система также безопасна, как была в начале дня. Это является определением **Основной теоремы безопасности системы** (Basic Security Theorem), используемой в компьютерной науке, которая гласит, что если система была инициализирована в безопасном состоянии, и она допускает только безопасные изменения состояния, она безопасна в любой момент, независимо от входных данных.

ПРИМЕЧАНИЕ. Принцип равновесия (tranquility principle), также используемый в этой модели, гласит, что субъекты и объекты не могут изменить свой уровень безопасности после своего создания.

Важно отметить, что модель Bell-LaPadula была разработана для гарантии сохранения секретности секретной информации, поэтому она обеспечивает и учитывает только конфиденциальность. Эта модель не учитывает целостность данных в системе — только кто может, а кто не может получить доступ к данным, и какие операции может выполнять с ними.

ПРИМЕЧАНИЕ. Обеспечение того, что информация не может переходить с более высокого уровня безопасности на более низкий, называется *контролем несанкционированного понижения статуса информации* (controlling unauthorized downgrading of information). Это может произойти при выполнении операции «запись вниз» (write down). При этом реальная компрометация

произойдет только в том случае и в тот момент, когда пользователь на более низком уровне безопасности прочитает эти данные.

Так что же это означает и зачем это нужно? В Домене 02 рассматривались системы мандатного управления доступом (MAC) в сравнении с системами дискреционного управления доступом (DAC). Все системы MAC основаны на модели Bell-LaPadula, т.к. она позволяет интегрировать многоуровневую безопасность в код. Субъектам и объектам присваиваются метки. Метка субъекта содержит отметку о его допуске («совершенно секретно», «секретно» или «конфиденциально»), а метка объекта содержит его уровень классификации («совершенно секретно», «секретно» или «конфиденциально»). Когда субъект пытается получить доступ к объекту, система сравнивает метку допуска субъекта с меткой классификации объекта, чтобы выяснить, является ли такой доступ допустимым и безопасным. В нашем сценарии, это разрешенное действие и субъект получает доступ к объекту. Теперь, если метка допуска субъекта – «совершенно секретно», а классификация объекта – «секретно», субъект не может записывать в этот объект из-за правила *-свойства, которое гарантирует, что субъект не сможет намеренно или случайно предоставить общий доступ к конфиденциальной информации, записав ее на более низкий уровень безопасности. В качестве примера представьте, что занятой и беспечный армейский генерал (который имеет доступ к совершенно секретной информации) открыл письмо с информацией о целях (которому была присвоена классификация «секретно»), которое затем будет отправлено всем секретарям на всех базах по всему миру. Он попытался дополнить письмо информацией, что США нападет на Кубу. При этом модель Bell-LaPadula будет приведена в действие и не позволит этому генералу записать такую информацию в это письмо, поскольку его доступ выше классификации письма.

Точно также, если секретарь попытается прочитать письмо, которое доступно только генералам и выше, модель Bell-LaPadula остановит эту попытку. Доступ секретаря ниже классификации объекта (письма), что нарушает простое правило безопасности модели. Все это относится к сохранению секретности.

ПРИМЕЧАНИЕ. Очень важно, чтобы MAC-операционная система и MAC-база данных следовали этим правилам. В Домене 09 мы рассмотрим, как база данных должна следовать этим правилам, используя многоэкземплятность (polyinstantiation).

ПРЕДУПРЕЖДЕНИЕ. Вы должны понимать правило Bell-LaPadula, называемое *Дискреционным свойством безопасности* (Discretionary Security Property - ds-property), которое является еще одним правилом этой модели. Это правило основано на именовании субъектов и объектов. Оно указывает, что определенное разрешение позволяет субъекту передавать разрешения на свой страх и риск. Эти разрешения хранятся в матрице доступа. Это просто означает, что механизмы мандатного и дискреционного управления доступом могут быть реализованы в одной операционной системе одновременно.

Правила, которые нужно знать. Основными правилами модели Bell-LaPadula, которые вы должны понимать, являются следующие:

- **Простое правило безопасности.** Субъект не может читать данные из объекта, находящегося на более высоком уровне безопасности (правило «Не читать сверху»).
- **Правило *- свойства.** Субъект не может записывать данные в объект, находящийся на более низком уровне безопасности (правило «Не записывать вниз»).
- **Строгое правило *-свойства.** Чтобы субъект мог читать и записывать данные в объект, его доступ и классификация объекта должны совпадать.

3.3. Модель Biba

Модель Biba была разработана позднее модели Bell-LaPadula. Она также является моделью конечных автоматов и очень похожа на Bell-LaPadula. Biba учитывает *целостность* данных в рамках приложений. Модель Bell-LaPadula использует сетку уровней безопасности («совершенно секретно», «секретно», «конфиденциально» и т.п.). Эти уровни безопасности были разработаны в основном для гарантии того, что критичные данные будут доступны

только уполномоченным лицам. Модель Biba не связана с уровнями безопасности и конфиденциальностью, поэтому она не основана на решениях о доступе в рамках такой сетки. Модель Biba использует сетку уровней целостности.

В случае правильного внедрения и реализации, модель Biba предотвращает переход данных с любого уровня *целостности* на более высокий уровень целостности. Biba имеет три основных правила для обеспечения такой защиты:

- **Аксиома *-целостности** (*-integrity axiom). Субъект не может записывать данные в объект, находящийся на более высоком уровне целостности (это называют «не записывать вверх» (no write up));
- **Простая аксиома целостности** (simple integrity axiom). Субъект не может читать данные, находящиеся на более низком уровне целостности (это называют «не читать снизу» (no read down)).
- **Свойство вызова** (Invocation property). Субъект не может запрашивать обслуживание (вызов) у другого субъекта, находящегося на более высоком уровне целостности.

Название «простая аксиома целостности» может звучать немного глупо, но это правило защищает субъекта и данные на более высоком уровне целостности от повреждения данными более низкого уровня целостности. Это касается доверия к источнику информации. «Грязные» (недоверенные) данные не должны быть смешаны с «чистыми» (доверенными) данными.

Простая аксиома целостности применяется не только к субъектам, создающим данные, но также и к процессам. Процесс на более низком уровне целостности не должен осуществлять запись в доверенные данные на более высоком уровне целостности. Области с различными уровнями целостности разделяются в приложениях, основанных на модели Biba.

Например, вы пишете статью для Нью-Йорк Таймс о трендах в безопасности за последний год, суммах потерь бизнеса, соотношении затрат/преимуществ внедрения межсетевых экранов, систем IDS и сканеров уязвимостей. Вы не хотите использовать данные и цифры со старых и безымянных веб-сайтов, не зная, как они были посчитаны и на основе каких источников информации. Ваша статья (данные на более высоком уровне целостности) может быть скомпрометирована безосновательной информацией, полученной из недостоверных источников (данные на более низком уровне целостности).

Модели Bell-LaPadula и Biba могут показаться очень похожими, а причины их разделения могут вызывать недоумение. Модель Bell-LaPadula была написана для американского правительства, а правительство просто параноидально относится к утечке секретной информации. В модели Bell-LaPadula пользователь не может записывать данные на более низкий уровень безопасности, поскольку это может привести к утечке секретной информации. Точно также, пользователь на более низком уровне безопасности не может читать что-либо на более высоком уровне безопасности, поскольку этот пользователь не уполномочен на получение доступа к этим секретам. Правительство и военные заботятся о конфиденциальности и придают большое значение защите секретов. Коммерческие организации в большей степени сосредоточены на *целостности* своих данных, поэтому многие из них используют приложения, реализующие модель Biba. Конечно, эти компании не ищут слово Biba на коробках с программными продуктами, которые они покупают. Какую модель использовать, решается и реализуется на этапе проектирования приложения. Покупатели используют рейтинг уровня гарантий для определения того, подходит им этот продукт или нет.

Bell-LaPadula vs. Biba. Модель Bell-LaPadula используется для обеспечения конфиденциальности. Модель Biba используется для обеспечения целостности. Модели Bell-LaPadula и Biba являются моделями информационных потоков, поскольку они в основном рассматривают процессы перехода данных с одного уровня на другой. Bell-LaPadula использует уровни безопасности, а Biba

использует уровни целостности. Для экзамена CISSP важно знать правила Biba и Bell-LaPadula. Эти правила звучат очень просто и их просто запомнить – слово «простое» используется в отношении чтения, а «*» - в отношении записи. Поэтому вам нужно помнить только направления чтения и записи в каждой модели.

Свойство вызова в модели Biba говорит, что субъект не может вызвать другой субъект, находящийся на более высоком уровне целостности. Прекрасно, но чем это отличается от двух других правил Biba? Аксиома *-целостности (не записывать вверх) указывает, как субъекты могут *записывать* информацию в объекты. Простая аксиома целостности (не читать снизу) указывает, как субъекты могут *читать* объекты. Свойство вызова указывает, как один субъект может взаимодействовать и инициализировать другой субъект во время выполнения. Примером вызова одним субъектом другого субъекта является отправка процессом запроса процедуре для выполнения определенной задачи. Субъектам разрешается запрашивать только инструменты на более низком уровне целостности. С использованием свойства вызова, система гарантирует, что «грязные» субъекты не могут запрашивать «чистые» инструменты для загрязнения чистых объектов.

Ссылки по теме:

- Security Models
- Course study materials for Introduction to Security, University of Cambridge, Dr. Markus Kuhn, principal lecturer (academic year 2003–2004)
- Chapter 3.3, “Models of OS Protection,” by Fred Cohen

3.4. Модель Clark-Wilson

Модель Clark-Wilson была разработана позднее модели Biba и использует несколько иной подход для защиты целостности информации. Эта модель использует следующие элементы:

- **Пользователи** (user) – активные агенты
- **Процедуры преобразования** (TP – transformation procedure) – запрограммированные абстрактные операции, такие как чтение, запись и изменение
- **Ограниченные элементы данных** (CDI – constrained data item) – могут управляться только TP
- **Неограниченные элементы данных** (UDI – unconstrained data item) – могут управляться пользователями посредством команд чтения и записи
- **Процедуры проверки целостности** (IVP – Integrity verification procedure) – запускаются периодически для проверки непротиворечивости CDI внешней действительности.

Когда приложение использует модель Clark-Wilson, оно разделяет данные на одно подмножество, которое должно быть максимально защищено (CDI), и подмножества, для которых такая защита не требуется (UDI). Пользователи не могут напрямую изменять критичные данные CDI. Вместо этого пользователи (субъекты) должны пройти аутентификацию в соответствующей части программного обеспечения, а программные процедуры (TP) выполняют необходимые операции от имени этих пользователей. Например, если Кэти нужно обновить информацию, содержащуюся в базе данных компании, она не может сделать этого без использования специального программного обеспечения, управляющего этими действиями. Сначала Кэти должна аутентифицироваться в соответствующей программе, которая является интерфейсом к базе данных, а затем программа будет управлять тем, что Кэти может и что не может делать с информацией в этой базе данных. Это называют *тройкой доступа* (access triple): субъект (пользователь), программа (TP) и объект (CDI). Пользователь не может вносить изменения в CDI без

использования TP.

Таким образом, Кэти собирается ввести данные, которые предположительно заменят исходные данные в базе данных. Программное обеспечение (TP) должно гарантировать, что эта деятельность безопасна, и выполнить процедуру записи для Кэти. Кэти (как и любой другой субъект) не достаточно доверена для управления объектами напрямую.

Целостность CDI должна быть защищена с помощью TP. UDI не требует такого уровня защиты. Например, если Кэти работает со своим интернет-банкингом, данные на сервере и в базе данных ее банка разделены на категории UDI и CDI. Категория CDI содержит информацию о ее банковском счете, которая должна быть максимально защищена. Данными UDI могут быть ее клиентский профиль, который она может обновлять при необходимости. TP не требуется, если Кэти нужно обновить свою информацию UDI.

В некоторых случаях системе может потребоваться перенести данные UDI в данные CDI. Например, когда Кэти обновляет свой клиентский профиль через веб-сайт, чтобы указать свой новый адрес проживания, эта информация должна быть отправлена банковскому программному обеспечению, ответственному за почтовую информацию в банковских счетах. Банк не хочет, чтобы Кэти напрямую взаимодействовала с банковским программным обеспечением, поэтому применяется компонент программного обеспечения (TP), ответственный за копирование таких данных и обновление почтового адреса клиента. На этом этапе TP меняет статус данных с UDI на CDI. Эта концепция показана на Рисунке 3-17.

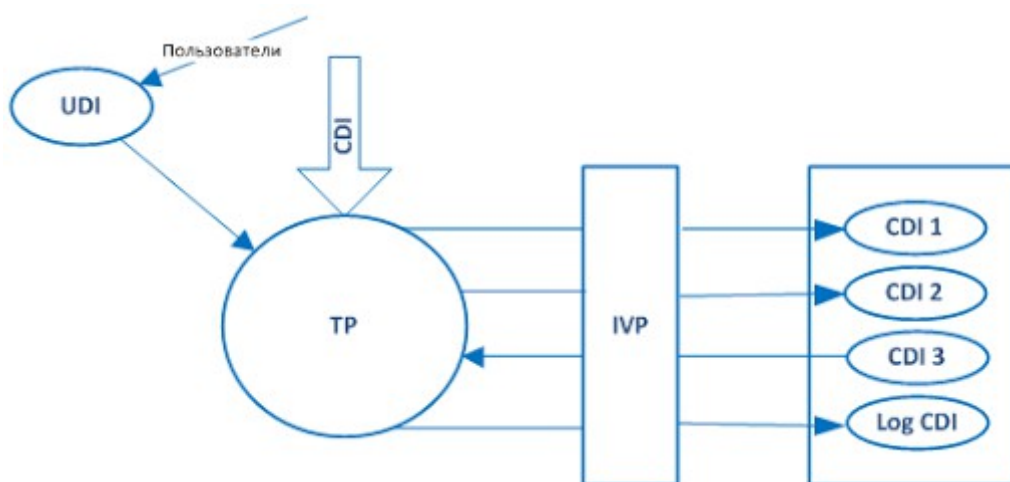


Рисунок 3-17. Субъект не может внести изменения в CDI без использования TP

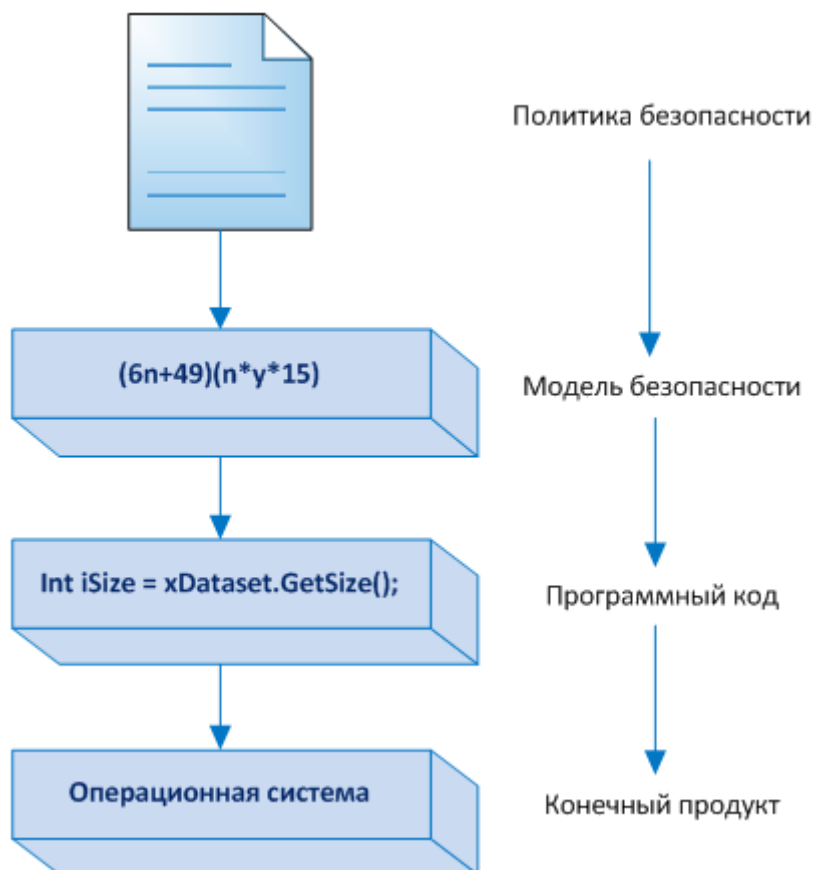
Помните, что это модель целостности, поэтому так много обсуждения идет в отношении правил целостности. Это задача IVP. IVP гарантирует, что все критичные данные (CDI) следуют определенным правилам целостности приложений.

Модель состоит из конструкций, математических формул и т.п. Модель предоставляет платформу, которая может использоваться для обеспечения конкретных характеристик в программном обеспечении (конфиденциальность, целостность и т.д.). Таким образом, модель не обуславливает, какие правила целостности должно реализовывать IVP; она просто предоставляет платформу, а производители выбирают правила целостности. Производители реализуют те правила целостности, которые нужны большинству их заказчиков. Поэтому если производитель разрабатывает приложение для финансовой компании, в состав UDI могут входить профили клиентов, которые позволено обновлять, а в состав CDI – информация о банковском счете, обычно хранящаяся на мейнфрейме. Данные UDI не требуется очень сильно защищать, они могут быть размещены на той же системе или на другой системе. Пользователь может иметь доступ к данным UDI без использования TP, однако когда пользователю нужно получить доступ к CDI, он обязан использовать TP. Таким образом, разработчики продукта определяют, какой тип данных будет рассматриваться как

UDI, а какой тип данных является CDI, и разрабатывают ТР для управления и дирижирования тем, как программное обеспечение будет обеспечивать целостность значений CDI.

В банковских приложениях IVP может гарантировать, что CDI содержит правильное значение. Например, если Кэти имеет 200 000 руб. на своем счете и кладет на него еще 50 000 руб., CDI для ее счета должен теперь иметь значение 250 000 руб. IVP гарантирует согласованность данных. Поэтому после того, как Кэти выполнит эту транзакцию и IVP проверит целостность CDI (новое значение остатка на банковском счете является правильным), тогда CDI будет считаться находящимся в согласованном (правильном) состоянии. Только ТР является компонентом, позволяющим изменить состояние CDI. В нашем примере ТР могут быть процедуры программного обеспечения, выполняющие операции пополнения счета, снятия денег со счета и их перевода. Использование ТР для изменения CDI называют правильной транзакцией.

Правильная транзакция (well-formed transaction) – это последовательность операций, которые выполняются для перевода данных из одного согласованного состояния в другое. Если Кэти переводит деньги со своего текущего счета на свой депозитный счет, эта транзакция состоит из двух частей – списания суммы с одного счета и зачисления на другой. Гарантируя правильность и целостность нового значения остатка на текущем и депозитном счетах, IVP поддерживает внутреннюю и внешнюю согласованность. Модель Clark-Wilson также описывает, как реализовать **разделение обязанностей** в архитектуре приложения. Возвращаясь к нашему примеру с банковским программным обеспечением, если клиенту нужно снять более 1 000 000 руб., приложение должно требовать авторизации этой операции супервизором (например, начальником операционного отдела). Это является защитной мерой против потенциального мошенничества. Эта модель предоставляет правила, которым должны следовать разработчики для правильной реализации разделения обязанностей в процедурах своего программного обеспечения.



Цели Моделей Целостности

Ниже приведены три основных цели моделей целостности:

- Предотвращение выполнения изменений неуполномоченными пользователями
- Предотвращение выполнения некорректных изменений уполномоченными пользователями (разделение обязанностей)
- Обеспечение внутренней и внешней согласованности (правильные транзакции)

Модель Clark-Wilson учитывает все три эти цели (посредством тройки доступа, разделения обязанностей и аудита). Эта модель обеспечивает целостность, используя правильные транзакции (посредством тройки доступа) и разделение обязанностей. Модель Biba учитывает только первую цель.

Внутренняя и внешняя согласованность обеспечивается IVP, гарантирующими значения всех CDI согласуются со входными значениями, изменившими их состояние. Так, если Кэти имеет 250 000 руб. на своем счете и снимает 200 000 руб., то результирующее значение в CDI будет 50 000 руб.

ПРИМЕЧАНИЕ. Матрица контроля доступа рассматривалась в Домене 02. Это еще одна модель, часто используемая в операционных системах и приложениях.

3.5. Модель информационных потоков

Модель Bell-LaPadula сосредоточена на предотвращении утечки информации с высокого уровня безопасности на более низкий. Модель Biba сосредоточена на предотвращении утечки информации с низкого уровня целостности на более высокий. Обе эти модели построены на **модели информационных потоков** (information flow model). Модель информационных потоков может учитывать любые информационные потоки, а не только переходы информации с одного уровня безопасности (целостности) на другой.

В модели информационных потоков данные рассматриваются как хранящиеся отдельно или разделенные на группы. В модели Bell-LaPadula эти группы основаны на уровнях безопасности. Помните, что системы MAC основаны на модели Bell-LaPadula. Системы MAC используют метки для каждого субъекта и объекта. Метка субъекта указывает на уровень допуска субъекта и его категории «должен знать». Метка объекта указывает на его классификацию и категорию. Если вы в армии и имеете доступ к совершенно секретной информации, это не означает, что вы можете получить доступ ко всей военной совершенно секретной информации. Информация разделяется по двум факторам – классификация и категории «должен знать». Ваш доступ должен «доминировать» над классификацией объекта, а ваш профиль безопасности должен содержать одну из категорий, перечисленных в метке объекта, что реализует принцип «должен знать». Таким образом, Bell-LaPadula – это модель информационных потоков, которая гарантирует, что информация не может перейти из одной группы в другую способом, угрожающим конфиденциальности данных. Biba делит данные на основе уровней целостности. Это также модель информационных потоков, которая управляет потоком информации способами, обеспечивающими защиту целостности наиболее доверенной информации.

Как может перемещаться информация в рамках всей компании? Ответить на этот вопрос можно разными способами. Субъекты могут использовать доступ к файлам. Процессы могут использовать доступ к сегментам памяти. Информационным потоком является перемещение данных из файла подкачки в память. Загрузка и выгрузка данных из регистров процессора. Перемещение данных в другую кэш-память. Запись данных на жесткий диск, флеш-накопитель, компакт-диск и т.п. Полноценный контроль всех этих потоков данных может быть очень сложной задачей. Именно поэтому существует модель информационных потоков, которая помогает архитекторам и разработчикам гарантировать, что их программное обеспечение не позволит информации перемещаться способами, которые могут представлять

опасность для системы или данных. Один из способов, которым модель информационных потоков обеспечивает такую защиту, является обеспечение отсутствия в коде скрытых каналов.

Скрытые каналы

Скрытые каналы (covert channel) – это способ несанкционированного получения информации. Это потоки информации, не контролируемые механизмами безопасности. Это такой информационный путь, который не был предназначен для передачи информации, поэтому системы не защищают его надлежащим образом, в большинстве случаев разработчики даже не представляли, что информация может передаваться таким образом. Получение информации посредством скрытых каналов является явным нарушением политики безопасности.

Скрытый канал для несанкционированной передачи данных является следствием одной из следующих причин:

- Недостаточный контроль при разработке продукта
- Некорректная реализация управления доступом в программном обеспечении
- Существование общих ресурсов между двумя сущностями
- Заражение троянской программой

Существует два типа скрытых каналов: по памяти и по времени. В **скрытом канале по памяти** (covert storage channel) процессы могут взаимодействовать через некое пространство хранения информации в системе. При этом один процесс записывает данные в определенное место (любой вид памяти, хранилища информации), а другой – напрямую или не напрямую – читает их. Например, система А заражена троянской программой, которая установила программное обеспечение, способное взаимодействовать с другим процессом ограниченным способом. Система А имеет очень критичный файл (File 2), который представляет большой интерес для конкретного атакующего. Программное обеспечение, установленное троянской программой, имеет доступ к этому файлу на чтение и ему нужно отправить его содержимое атакующему, причем это будет происходить только по одному биту за раз. Это программное обеспечение будет взаимодействовать с атакующим посредством блокировки определенного файла (File 3). Когда атакующий попытается получить доступ к File 3 и выявит, что он заблокирован этой программой, он сделает вывод, что первый бит критичного файла – «1». Когда в следующий раз атакующий попытается получить доступ к File 3, он окажется незаблокирован. Атакующий интерпретирует это как следующее значение, являющееся «0». Это будет продолжаться до тех пор, пока все данные критичного файла не будут отправлены атакующему. В этом примере программное обеспечение, установленное троянской программой, является «передатчиком». Оно может получить доступ к критичному файлу и воспользоваться другим файлом на жестком диске для отправки сигналов атакующему.

Другие типы скрытых каналов. Мы упомянули скрытые каналы в рамках программного кода, но скрытые каналы могут находиться и во внешнем мире. Скажем, вы собираетесь посетить одну из лекций. Перед началом лекции вы с лектором договариваетесь о способе взаимодействия, который не поймет никто в аудитории. Лектор говорит вам, что если он будет вертеть ручку между пальцами правой руки, это значит, что в конце занятия будет экзамен. Если он будет вертеть ручку между пальцами левой руки, значит, что экзамена не будет. Это тоже скрытый канал, поскольку это не нормальный способ взаимодействия, а секретный.

Другой способ атаки через скрытый канал по памяти может осуществляться посредством создания файла. Система может быть скомпрометирована и на нее может быть установлено программное обеспечение, которое может создавать и удалять файлы в определенной директории и иметь доступ на чтение критичного файла. Когда это программное обеспечение увидит, что первый бит данных критичного файла является «1», оно создаст файл с именем Temp в определенной директории. Атакующий будет пытаться создать в этой

же директории файл с таким же именем. Если атакующий получит сообщение об ошибке, говорящее, что такой файл уже существует в этой директории, он будет знать, что первый бит критичного файла – «1». Атакующий будет пытаться создать такой же файл снова и, если система позволит ему сделать это, значит вредоносное программное обеспечение, установленное в системе, удалило этот файл, говоря, что следующим битом является «0».

Модели информационных потоков предоставляют правила, позволяющие гарантировать отсутствие скрытых каналов. Но существует множество различных вариантов информационных потоков в системе, поэтому выявление и искоренение скрытых каналов обычно является более сложной задачей, чем может показаться на первый взгляд.

ПРИМЕЧАНИЕ. Открытый канал – это коммуникационный канал, который был разработан специально для коммуникационных целей. Процессам следует взаимодействовать через открытые каналы, а не через скрытые.

В скрытых каналах по времени (covert timing channel) один процесс передает информацию другому, получив сигнал в виде использования системных ресурсов. Два процесса взаимодействуют друг с другом, используя один и тот же общий ресурс, которым является время. Например, если один процесс обратился к диску 30 раз за 30 секунд, это является сигналом для другого процесса выполнить определенные вредоносные действия, на которые он был заранее запрограммирован. Еще один пример. Предположим, что процесс А является частью вредоносного программного обеспечения, установленного троянской программой. В многозадачной системе каждый процесс имеет возможность взаимодействия с процессором. Когда эта возможность предоставляется процессу А, он отвергает ее, что соответствует «1» для атакующего. В следующий раз процесс А использует доступ к процессору, что соответствует «0» для атакующего. Представьте, что это разновидность азбуки Морзе, использующая некоторый вид системных ресурсов.

Контрмеры

Поскольку все операционные системы имеют некоторые виды скрытых каналов, не всегда реально освободиться от всех из них. Число допустимых скрытых каналов обычно зависит от рейтинга уровня гарантий системы. Система, которая имеет рейтинг EAL 6 по Общим критериям, содержит меньше скрытых каналов, чем система с рейтингом EAL 3, поскольку рейтинг EAL 6 представляет собой более высокий уровень гарантий обеспечения конкретного уровня защиты, по сравнению с рейтингом EAL 3. Немногие пользователи могут обеспечить противодействие таким каналам, более правильно, чтобы такие каналы были учтены при проектировании и разработке системы.

ПРИМЕЧАНИЕ. В Оранжевой книге скрытые каналы в операционных системах не учитываются, пока уровень безопасности не опускается ниже уровня B2. Системы на уровнях ниже B2 хранят и обрабатывают достаточно критичную информацию, чтобы атакующие нашли любые пути для доступа к этой информации, в том числе посредством скрытых каналов.

Ссылки по теме:

- “Secure Databases: An Analysis of Clark-Wilson Model in a Database Environment,” by Xiaocheng Ge, Fiona Polack, and R gine Laleau
- “Access Control: Theory and Practice”
- “New Thinking About Information Technology Security,” by Marshall D. Abrams, PhD and Michael V. Joyce (first published in Computers & Security, Vol. 14, No. 1, pp. 57–68)

3.6. Модель невлиания

Свойства многоуровневой безопасности могут быть выражены многими способами, одним из них является «*невлиание*». Эта концепция реализована для обеспечения гарантий того, что любые действия, происходящие на высоком уровне безопасности, не воздействуют (не влияют) на действия, происходящие на более низком уровне безопасности. Эта модель не

имеет отношения к потокам данных, она относится к тому, что субъект знает о состоянии системы. Так, если сущность на высоком уровне безопасности выполняет некоторое действие, оно не может изменить состояние для сущности, находящейся на более низком уровне.

Если сущность на низком уровне безопасности узнает о том, что сущность на высоком уровне безопасности произвела определенные действия, из-за которых изменилось состояние системы для этой сущности на низком уровне, она (эта сущность) может догадаться (предположить) о дополнительной информации, относящейся к этим действиям на высоком уровне, что является разновидностью утечки информации.

Пользователи на низком уровне безопасности не должны знать о действиях, выполненных пользователями на высоком уровне безопасности, эти действия не должны влиять на пользователей на низком уровне безопасности.

Скажем, Том и Кэти одновременно работают на мейнфрейме, реализующем многоуровневую безопасность. Том имеет допуск уровня «секретно», а Кэти – «совершенно секретно». Соответственно терминал Тома работает в контексте «секретно», а терминал Кэти – «совершенно секретно». Эта модель утверждает, что никакие из действий Кэти на своем терминале не окажут прямого или косвенного воздействия на домен Тома (доступные ресурсы и рабочая среда). Независимо от выполняемых ею команд и используемых ресурсов, это никоим образом не повлияет на работу Тома на этом мейнфрейме.

То, что выполняемые Кэти команды не должны влиять на терминал Тома, выглядит вполне логично. Но реальный смысл этой модели состоит в учете скрытых каналов и атак посредством предположений о действиях. Эта модель рассматривает общие ресурсы, которые будут использовать различные пользователи системы, и пытается идентифицировать способы, посредством которых информация может переходить от процессов, работающих с высоким уровнем допуска, к процессам с меньшим уровнем допуска. Т.к. Том и Кэти работают в одной и той же системе в одно и то же время, они, скорее всего, будут иметь некоторые общие ресурсы. Данная модель содержит правила, гарантирующие, что Кэти не сможет отправить данные Тому посредством скрытых каналов (по памяти или по времени).

Другим недостатком безопасности, учтенным данной моделью, являются **атаки посредством предположений о действиях** (inference attack). Такая атака происходит, когда человек имеет доступ к некоему типу информации и может на основании нее сделать вывод (или догадаться) о чем-то, что ему не положено знать в соответствии с его уровнем допуска или полномочиями. Например, Том работал с файлом, содержащим информацию о России. Он закрыл этот файл, а когда через час попытался открыть его снова, получил отказ в доступе к нему, т.к. в это время кто-то повысил уровень секретности данного файла до уровня «совершенно секретно». Том может предположить, что готовится некая совершенно секретная миссия в отношении России, но к информации об этом Том не был допущен. Это является атакой посредством предположения и фактом «утечки информации» (Атаки посредством предположений будут рассматриваться позднее в Домене 09).

3.7. Сетчатая модель

Сетка (lattice) – это математическая конструкция, построенная на основе понятия группы. Стандартное определение **сетчатой модели** (lattice model) звучит следующим образом: это «структура, состоящая из конечного, частично упорядоченного множества с определенной самой верхней границей и самой нижней границами операторов множества».

В этом определении есть существенные недостатки. Во-первых, такая формулировка может быть понятна только тем, кто хорошо понимает эту модель. Это похоже на определение метаданных: «данные о данных». Только *после* того, как вы действительно поймете, что такое метаданные, это определение обретет смысл для вас. Таким образом, приведенное

выше определение сетчатой модели практически бесполезно. Кроме того, еще одна проблема с таким математическим объяснением состоит в том, что только доктор математических наук сможет его понять. Эта модель должна объясняться обычным, общепонятным языком. Давайте попробуем сделать это.

Вспомним модель МАС, которая рассматривалась в Домене 02, а затем затрагивалась и в этом Домене. В этой модели субъекты и объекты имеют метки. Метка каждого субъекта содержит допуск и категории «должен знать», к которым этот субъект имеет доступ. Предположим, что Кэти имеет уровень допуска «совершенно секретно» и она формально имеет доступ к разделам «Ирак» и «Корея» в соответствии со своими категориями «должен знать». Таким образом, ее метка – **СС {Ирак, Корея}**. Таблица 3-1 показывает различные файлы в системе в соответствии с этим сценарием. Эта система основана на модели МАС, т.е. операционная система принимает решения о возможности доступа на основе содержимого меток безопасности.

Метка безопасности Кэти	Метка безопасности Файла В	Метка безопасности Файла С	Метка безопасности Файла D
Совершенно Секретно {Ирак, Корея}	Секретно {Ирак}	Совершенно Секретно {Ирак, Корея}	Секретно {Ирак, Корея, Иран}

Таблица 3-1. Элементы управления доступом

Кэти пытается получить доступ к файлу В, поскольку ее допуск выше классификации файла В, она может читать этот файл, но не записывать в него (помните, в Bell-LaPadula субъект с более высоким уровнем может «читать снизу», но не «записывать вверх»). Это относится к *«частично упорядоченному множеству с определенной самой верхней границей и самой нижней границами операторов множества»*. Множество – это субъект (Кэти) и объект (файл). Это частично упорядоченное множество, т.к. не все атрибуты управления доступом полностью эквивалентны. Система должна выбрать между чтением, записью, полным доступом, изменением и всеми другими видами разрешений доступа, используемыми этой операционной системой. Таким образом, «частично упорядоченное» означает, что система должна применять самый ограниченный вариант доступа к этому множеству. В качестве «самой верхней границы» система берет одно утверждение управления доступом (Кэти может читать файл), а другое утверждение управления доступом (Кэти не может записывать в файл) считается «самой нижней границей». Поскольку запрет записи – это более ограниченный вариант, чем разрешение чтения, самой *верхней* границей Кэти для этого файла будет чтение, а самой *нижней* границей – запрет записи. Рисунок 3-18 иллюстрирует границы доступа. Это просто более сложный способ сказать, что «самое большее, что Кэти может делать с этим файлом, это читать его. А самое меньшее – она не может записывать в него».

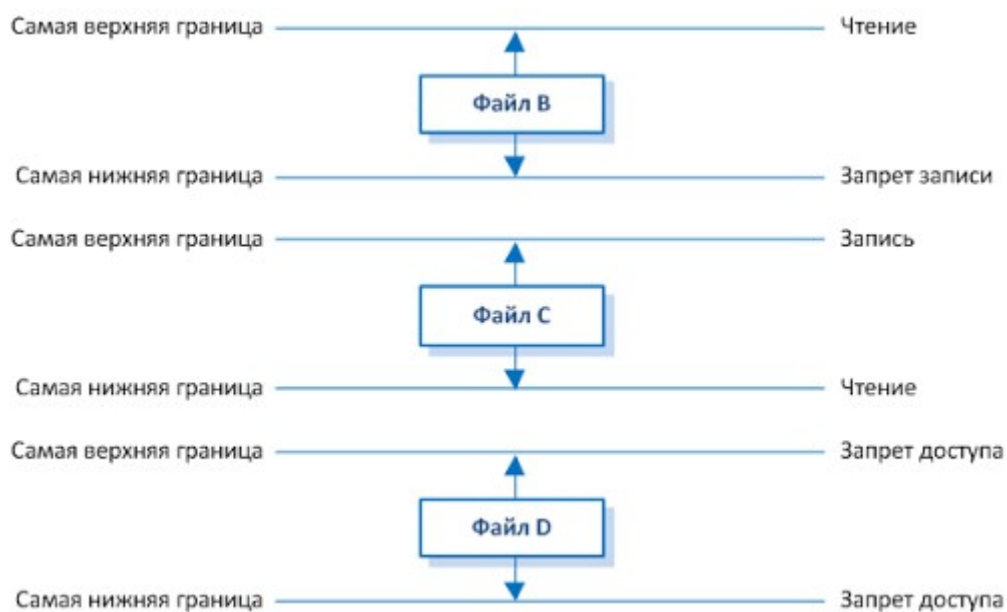


Рисунок 3-18. Границы доступа в сетчатой модели

Давайте изобразим самую верхнюю и самую нижнюю границы уровня доступа для Кэти и файла С. Допуск Кэти совпадает с классификацией файла С. В соответствии с моделью Bell-LaPadula в этом случае нужно применять строгое правило *-свойства (субъект имеет доступ на чтение и запись к объекту на том же уровне безопасности). Поэтому самая верхняя граница – это запись, а самая нижняя – чтение.

Посмотрим на метку безопасности файла D – она имеет категорию **{Иран}**, которая отсутствует в метке безопасности Кэти. Это означает, что у Кэти нет необходимой категории «должен знать» для доступа к этому файлу. Самая верхняя и самая нижняя границы разрешений доступа Кэти в этом случае будут соответствовать уровню «Нет доступа».

Так зачем же описывать эту достаточно простую концепцию таким сложным образом? Во-первых, здесь эта модель описана самыми простыми из возможных терминов, чтобы вам было проще понять ее назначение. Сложность в этой концепции появляется, если представить все взаимодействия субъект-объект, происходящие в рамках операционной системы каждую секунду. К тому же это формальная модель, т.е. она может доказать математически обеспечение определенного уровня защиты, при правильном следовании всем ее правилам. Изучение этой модели похоже на изучение основ химии. Студент сначала изучает, из чего состоит атом (протоны, нейтроны и электроны) и как эти элементы взаимодействуют друг с другом. Это легкая часть. Затем студент переходит к органической химии и должен понять, как все эти элементы работают вместе в сложной органической системе. Затем студент переходит к квантовой физике и изучает, что отдельные атомы в действительности имеют несколько различных субатомных частиц (кварки, лептоны и мезоны). В этой книге, вы изучаете только базовые компоненты этой модели.

3.8. Модель Brewer and Nash

Модель Brewer and Nash, также называемая **моделью «Китайской стены»**, создана для обеспечения управления доступом, который может динамически изменяться в зависимости от предыдущих действий пользователя. Основной целью модели является защита от конфликтов интересов, вызванных попытками получения доступа пользователями. Например, большая маркетинговая компания реализует продвижение и предоставляет маркетинговые материалы двум банкам. При этом сотрудники этой компании, работающие над проектом для банка А не должны видеть маркетинговую информацию, относящуюся к банку Б, поскольку это может привести к конфликту интересов, т.к. банки конкурируют между собой. Если менеджер проекта для банка А сможет увидеть маркетинговые материалы

для банка В, он сможет обеспечить наилучшее продвижение банка А, напрямую привлекая его клиентов. Это может повредить репутации маркетинговой компании, в которой сотрудники ведут себя настолько безответственно.

Этой компании следует внедрить продукт, который отслеживает доступ к информации представителей обеих этих групп и предотвращает конфликт интересов. На Рисунке 3-19 мы видим, что когда пользователь получает доступ к информации банка А, система автоматически запрещает ему доступ к информации банка В. Права доступа изменяются динамически на основе авторизации пользователей, их деятельности, предыдущих запросов на доступ.

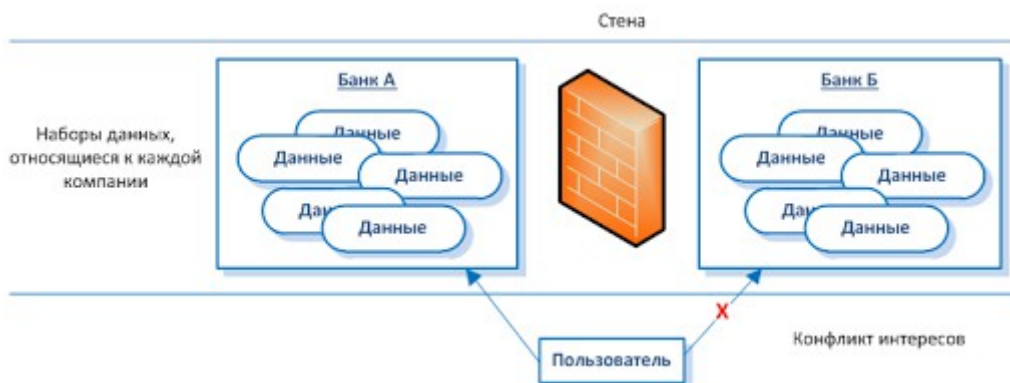


Рисунок 3-19. Модель «Китайская стена» реализует динамическое управление доступом

Модель «Китайская стена» также основана на модели информационных потоков. Информация может передаваться между субъектом и объектом только такими способами, которые не могут привести к конфликту интересов. Модель гласит, что субъект может записывать информацию в объект только в том случае, если субъект не может читать другой объект, находящийся в другом наборе данных. Таким образом, возвращаясь к нашему примеру, менеджер проекта не сможет записать информацию в любой объект в рамках набора данных банка А, если он в настоящее время имеет доступ на чтение к любому из объектов в рамках набора данных банка В.

3.9. Модель Graham-Denning

Помните, что все вышеописанное является моделями, поэтому они не очень конкретны. Каждый производитель должен сам принимать решения как выполнять то или иное правило, определенное в выбранной модели. Модели Bell-LaPadula и Biba не определяют, как оцениваются и изменяются рейтинги безопасности и целостности, а также не предоставляют способа делегирования и передачи прав доступа. **Модель Graham-Denning** учитывает эти вопросы и определяет набор базовых прав в терминах команд, которые определенный субъект может выполнять над объектом. Эта модель имеет восемь команд защиты прав или правил, определяющих каким образом эта функциональность должна выполняться безопасным образом.

- Как безопасно создать объект
- Как безопасно создать субъект
- Как безопасно удалить объект
- Как безопасно удалить субъект
- Как безопасно прочесть права доступа
- Как безопасно предоставить права доступа
- Как безопасно удалить права доступа

- Как безопасно передать права доступа

Эти вещи могут звучать как незначительные, однако при построении безопасной системы они играют критически важную роль.

3.10. Модель Harrison-Ruzzo-Ulman

Модель Harrison-Ruzzo-Ulman связана с правами доступа субъектов и целостностью этих прав. Субъект может выполнять только конечный набор операций над объектом. Пока безопасность достаточно проста, системе не сложно разрешить или запретить выполнение операций при условии, что одна команда ограничена одной операцией. Например, если субъект отправляет команду X, которая требует выполнения только операции Y, это позволяет системе достаточно просто принять решение о разрешении или запрете этой операции. Но если субъект отправляет команду M, для выполнения которой нужно выполнить операции N, B, W и P — системе гораздо более сложно решить, следует ли разрешать эту команду.

Обобщение информации по моделям. Для упрощения понимания всех этих моделей, ниже приведены их ключевые концепции.

- **Модель Bell-LaPadula.** Это модель конфиденциальности, описывающая допустимые информационные потоки и формализующая военную политику безопасности. Это первая математическая модель многоуровневой политики безопасности, которая определяет концепцию безопасного состояния и необходимые режимы доступа.
 - **Простое правило безопасности.** Субъект не может читать данные более высокого уровня безопасности, чем его допуск («не читать сверху»);
 - **Правило *-свойства.** Субъект не может записывать данные в объект меньшего уровня безопасности, чем его допуск («не записывать вниз»);
 - **Строгое правило *-свойства.** Субъект может выполнять функции чтения-записи только в отношении объекта, находящегося на том же уровне безопасности.
- **Модель Biba.** Эта модель защищает целостность информации в рамках системы и происходящих в ней действий. Она учитывает первую цель целостности.
 - **Простая аксиома целостности.** Субъект не может читать данные с более низкого уровня целостности («не читать снизу»).
 - **Аксиома *-целостности.** Субъект не может изменять объект на более высоком уровне целостности («не записывать вверх»).
- **Модель Clark-Wilson.** Это модель целостности, реализованная для защиты целостности данных и обеспечения того, чтобы выполнялись только корректные транзакции. Она учитывает все три цели целостности.
 - Субъекты могут получить доступ к объектам только посредством авторизованных программ (тройка доступа).
 - Реализация разделения обязанностей.
 - Требуется ведение аудита.
- **Модель матрицы контроля доступа.** Это модель, которая принимает решения о предоставлении доступа на основе ACL объектов и таблиц разрешений субъектов.
- **Модель информационных потоков.** Это модель, в которой информация ограничена в своих потоках и может передаваться между сущностями только способами, которые не могут нарушить политику безопасности.
- **Модель невливания.** Эта модель гласит, что команды и действия, выполняющиеся на одном уровне безопасности, не могут быть замечены и не могут оказать влияние на субъекты или объекты на другом уровне безопасности.
- **Модель Brewer and Nash.** Эта модель позволяет динамически изменять права доступа для защиты от конфликта интересов. Также известна как модель «Китайская стена».
- **Модель Graham-Denning.** Эта модель показывает, как следует создавать и удалять

субъекты и объекты. Она также учитывает, как назначать отдельные права доступа.

Ссылки по теме:

- Various Papers on Models
- “A Lattice Model of Secure Information Flow,” by Dorothy E. Denning (first published in Communications of the ACM, Vol. 19, No. 5, pp. 236–243 (May 1976))
- Course syllabus for Security, University of Cambridge, Dr. Ross Anderson, principal lecturer (Jan. 1999)

4. Режимы безопасности функционирования

Система может работать в различных режимах в зависимости от критичности обрабатываемых данных, уровня допуска пользователей и того, на выполнение каких действий пользователи авторизованы. Режим функционирования описывает условия безопасности, в которых система реально функционирует.

Эти режимы используются в системах МАС, в которых хранятся данные одного или нескольких уровней классификации. Несколько вещей играют роль при определении режима функционирования, в котором должна работать операционная система:

- Типы пользователей, которые напрямую или не напрямую подключаются к системе
- Типы данных (уровни классификации, разделы и категории), которые обрабатываются в системе
- Уровни допуска, категории «должен знать», формальные разрешения доступа, которые имеют пользователи

Следующие разделы описывают различные режимы безопасности функционирования, для работы в которых может быть разработана и настроена операционная система.

4.1. Специальный режим безопасности

Система функционирует в **специальном режиме безопасности** (dedicated security mode), если все пользователи имеют допуск и категории «должен знать» для *всех* данных, обрабатываемых в системе. Все пользователи имеют формальные разрешения доступа ко *всей* информации в системе, с ними заключено соглашение о неразглашении конфиденциальной информации (NDA - Nondisclosure agreement) в отношении этой информации. Система может работать только с одним уровнем классификации информации.

Многие военные системы разработаны для работы только с одним уровнем безопасности, они работают в специальном режиме безопасности. Это требует, чтобы все, кто использует систему, имели максимальный уровень допуска, достаточный для доступа к любым данным в системе. Если система содержит совершенно секретные данные, только пользователи с допуском к совершенно секретной информации могут иметь доступ к ней. Другие военные системы работают с несколькими уровнями безопасности, что реализуется посредством разделения данных. Такие виды систем могут поддерживать работу пользователей с высоким и низким уровнем допуска одновременно.

4.2. Режим повышенной безопасности системы

Система функционирует в **режиме повышенной безопасности** (system high-security mode), когда все пользователи имеют допуск к информации, но не ко всей обрабатываемой в системе информации они имеют разрешенную категорию «должен знать». Т.е. в режиме повышенной безопасности пользователи имеют категории «должен знать» только для *некоторых* данных.

Этот режим также требует, чтобы все пользователи имели наивысший уровень допуска ко

всем данным в системе. Однако при этом пользователи могут быть ограничены в доступе к отдельным объектам, если они не имеют соответствующую категорию «должен знать».

4.3. Раздельный режим безопасности

Система функционирует в *раздельном режиме безопасности* (compartment security mode), когда все пользователи имеют доступ ко всей информации, обрабатываемой в системе с конфигурацией повышенной безопасности, но могут не иметь отдельных категорий «должен знать» и формальных разрешений доступа. Это означает, что если система хранит секретные и совершенно секретные данные, все пользователи должны иметь доступ не ниже совершенно секретного, чтобы получить доступ к системе. В этом заключается различие раздельного и многоуровневого режимов безопасности. Оба режима требуют, чтобы пользователь имел правильные категории "должен знать", NDA и формальное разрешение, но в раздельном режиме безопасности требуется, чтобы пользователь имел доминирующий (большой или равный) уровень допуска над любыми данными в системе, тогда как в многоуровневом режиме безопасности просто требуется, чтобы пользователь имел доступ к данным, с которыми он собирается работать.

В раздельном режиме безопасности пользователи имеют ограничения в доступе к некоторой информации, т.к. она не нужна им для выполнения своих обязанностей и они не имеют формального разрешения доступа к ней. Это обеспечивается наличием у всех объектов меток безопасности, отражающих уровень их критичности (уровень классификации, категория классификации, процедуры обработки) информации. В этом режиме пользователи могут получить доступ к разделам данных только посредством мандатного управления доступом.

Целью является гарантия того, что минимально возможное число людей знают информацию на каждом уровне. Разделы являются категориями данных с ограниченным числом субъектов, допущенных к данным на каждом уровне. *Рабочие станции раздельного режима* (CMW – compartment mode workstation) позволяют пользователям обрабатывать различные разделы данных одновременно, если они имеют необходимый доступ.

4.4. Многоуровневый режим безопасности

Система функционирует в *многоуровневом режиме безопасности* (multilevel security mode), если разрешена одновременная обработка информации двух или более уровней классификации, когда не у всех пользователей есть доступ или формальное разрешение на доступ ко всей информации, обрабатываемой в системе. Поэтому все пользователи должны иметь формальное разрешение, NDA, категорию "должен знать" и необходимый уровень допуска для доступа к данным, которые нужны им для выполнения своих должностных обязанностей. В этом режиме пользователь не может получить доступ ко всем данным в системе только на основе своего допуска.

Модель Bell-LaPadula является примером многоуровневой модели безопасности, поскольку она одновременно обрабатывает информацию с различной классификацией, на различных уровнях безопасности в рамках одной системы.

Охранники

Программные и аппаратные охранники (guard) позволяют обмениваться данными между доверенными (с высоким уровнем гарантий) и менее доверенными (с низким уровнем гарантий) системами и средами. Скажем, вы работаете в системе MAC (работающей в специальном режиме безопасности с секретными данными) и вам нужно взаимодействовать с базой данных MAC (работающей в режиме многоуровневой безопасности, классификация данных в которой доходит до уровня совершенно секретно). Эти две системы обеспечивают различные уровни защиты. Как было сказано ранее, если система с меньшими гарантиями может напрямую взаимодействовать с системой с высокими гарантиями, это может привести к появлению уязвимостей и нарушению безопасности. Чтобы избежать этого, можно

внедрить программного охранника, который в действительности является просто интерфейсным продуктом, позволяющим осуществлять взаимодействие между системами, работающими на разных уровнях безопасности. (Существуют различные типы охранников, которые могут выполнять фильтрацию, обработку запросов, блокировку и антивирусную обработку данных). Либо может быть внедрен аппаратный охранник, который является системой с двумя сетевыми картами, каждая из которых подключена к одной из систем, которым нужно взаимодействовать друг с другом. Охранник - это дополнительный компонент, который обеспечивает строгое управление доступом при взаимодействии между различными системами.

Охранник принимает запросы от системы с низким уровнем гарантий, просматривает их, чтобы убедиться в их допустимости, и направляет эти запросы в более гарантированную систему. Целью является обеспечение того, чтобы информация не переходила с высокого уровня безопасности на низкий уровень безопасности несанкционированным способом.

Охранники могут использоваться для подключения различных систем МАС, работающих в разных режимах безопасности, и для соединения различных сетей, работающих на разных уровнях безопасности. Во многих случаях, менее доверенная система может отправить сообщения более доверенной системе, но при этом она может получить в ответ только подтверждение о получении. Это обычная практика, когда нужно отправить сообщение с менее доверенной системы на более доверенную, классифицированную систему.

Обобщение информации по режимам безопасности. Значительно проще понять эти различные режимы, когда они описаны в простой и понятной форме. Обратите внимание на слова, выделенные курсивом, поскольку они отражают различия между различными режимами.

Специальный режим безопасности. Все пользователи должны иметь:

- Надлежащий допуск ко всей информации в системе
- Формальное разрешение на доступ ко всей информации в системе
- Подписанное NDA для всей информации в системе
- Правильные категории "должен знать" для *всей* информации в системе
- Все пользователи имеют доступ ко всем данным

Режим повышенной безопасности системы. Все пользователи должны иметь:

- Надлежащий допуск ко всей информации в системе
- Формальное разрешение на доступ ко всей информации в системе
- Подписанное NDA для всей информации в системе
- Правильные категории "должен знать" для *некоторой* информации в системе
- Пользователи могут получить доступ к некоторым данным на основе своих категорий "должен знать"

Раздельный режим безопасности. Все пользователи должны иметь:

- Надлежащий допуск к наивысшему классу информации, среди содержащихся в системе
- Формальное разрешение доступа ко всей информации в системе, которую они будут использовать
- Подписанное NDA для всей информации в системе, которую они будут использовать
- Правильные категории "должен знать" для *некоторой* информации в системе
- Все пользователи имеют доступ к некоторым данным на основе их категорий "должен знать" и формальных разрешений доступа

Многоуровневый режим безопасности. Все пользователи должны иметь:

- Надлежащий допуск к необходимому ему классу информации
- Формальное разрешение доступа ко всей информации в системе, которую они будут

использовать

- Подписанное NDA для всей информации в системе, которую они будут использовать
- Правильные категории "должен знать" для *некоторой* информации в системе
- Все пользователи могут использовать некоторые данные на основе своих категорий "должен знать", уровней допуска и формальных разрешений доступа

Ссылки по теме:

- "Data Protection Measures"
- "Physical Model of Operations"
- National Industrial Security Program Operating Manual (NISPOM), U.S. Department of Defense

4.5. Доверие и гарантии

Как уже говорилось ранее в разделе « Доверенная компьютерная база», в действительности не существует полностью защищенных систем и при наличии достаточных ресурсов у злоумышленника он может так или иначе скомпрометировать практически любую систему, однако системы могут предоставлять уровень доверия. Уровень доверия говорит покупателю, насколько надежную защиту он может ожидать от этой системы, и насколько он может быть **уверен** в том, что система будет функционировать правильным и предсказуемым образом в любой ситуации.

ТСВ включает в себя все механизмы защиты системы (аппаратное обеспечение, программное обеспечение и прошивки). Все эти механизмы должны работать согласованно для реализации всех требований политики безопасности. В процессе оценки эти механизмы тестируются, их структура анализируется, просматривается и оценивается документация по их сопровождению. При оценке уровня доверия к системе учитывается все - как система была разработана, как она поддерживалась, и даже как она была доставлена клиенту. Каждый компонент проходит процедуру оценки и каждому выставляется соответствующий рейтинг гарантий, представляющий собой степень доверия и гарантий, которые вызвал продукт у команды тестировщиков. В дальнейшем заказчики используют этот рейтинг для определения того, насколько эта система подойдет их требованиям безопасности.

Доверие (trust) и гарантии (assurance) похожи, но немного отличаются в отношении рейтингов продуктов. В доверенной системе все защитные механизмы работают совместно для обеспечения защиты критичных данных при их обработке различными способами и обеспечения необходимого уровня защиты в соответствии с уровнем классификации. Гарантия смотрит на те же самые вопросы, но более глубоко и детально. Системы, обеспечивающие высокий уровень гарантий, детально протестированы, их структура глубоко проанализирована, рассмотрены этапы их разработки, оценены их технические спецификации и планы тестирования. Вы можете купить машину и доверять ей, но вы будете иметь значительно более глубокое чувство уверенности, если вы будете знать, как эта машина была создана, кто ее изготовил, какие тесты она прошла и как она ведет себя в различных ситуациях.

В стандарте TCSEC (Trusted Computer System Evaluation Criteria), также известном как « Оранжевая книга», для систем, претендующих на низкий рейтинг гарантий, для присвоения рейтинга анализируются механизмы безопасности и результаты тестирования, а для систем, претендующих на высокий уровень гарантий, рассматривают также структуру системы, спецификации, процедуры разработки, документацию по сопровождению и результаты тестирования. Механизмы защиты в системах с высоким уровнем гарантий могут не иметь существенных отличий от механизмов защиты в системах с более низким уровнем гарантий, однако для них они должны быть спроектированы и построены гораздо более внимательно. Эта более тщательная проверка дает возможность присвоить системе более высокий уровень

гарантий.

5. Методы оценки систем

При *оценке безопасности* системы проверяются части системы, имеющие отношение к ее безопасности: ТСВ, механизмы управления доступом, монитор обращений, ядро и механизмы защиты. Взаимоотношения и взаимодействие между этими компонентами также подлежат оценке. Существуют различные методы оценки и присвоения уровней гарантии системам. Но почему недостаточно только одного метода? Это связано с тем, что методы и идеология со временем развиваются, а различные страны смотрят на компьютерную безопасность по-разному и по-разному оценивают некоторые аспекты безопасности. Далее будет описан каждый из методов, и все методы будут сравнены между собой.

5.1. Зачем проводить оценку продукта?

Оценка продукта на соответствие Оранжевой книге, ITSEC или Общим критериям – это далеко не прогулка в парке для производителя. В действительности, это длительный и сложный процесс. Прежде чем перейти к этим критериям, давайте посмотрим, зачем вообще нужно проходить через этот процесс.

Если вы идете в магазин, чтобы купить межсетевой экран, как вы узнаете уровень защиты, который предоставляет каждый из имеющихся в магазине межсетевых экранов? Какой из них лучше всего подходит для вашей среды? Вы можете послушать речи маркетологов производителя или поверить менеджеру магазина, который расскажет вам, что все ваши проблемы будут решены определенным продуктом всего за неделю. Либо вы можете послушать совет независимой третьей стороны, которая всесторонне проверила продукт и не имеет к нему никаких замечаний. Если вы выбираете последний вариант, вам нужно присоединиться к миру людей, которые используют рейтинги гарантий.

В США Национальный центр компьютерной безопасности (NCSC – National Computer Security Center), входящий в Агентства национальной безопасности (NSA – National Security Agency), является организацией, которая отвечает за оценку компьютерных систем и продуктов. NCSC использует Программу оценки доверия к продукту (TPEP – Trusted Product Evaluation Program) при проведении тестирования коммерческих продуктов на соответствие определенному набору критериев для присвоения им рейтинга.

Таким образом, производители создают продукт и передают его на оценку, которая является проверкой соответствия требованиям TPEP. Оценивающая организация имеет группу тестирующих, которые следуют набору критериев (многие годы этими критериями была Оранжевая книга, но сейчас для этих целей используются Общие критерии) для тестирования продукта, предоставленного производителем на оценку. После прохождения оценки, продукту присваивается рейтинг уровня гарантий. Успешно оцененные продукты размещаются в Списке оцененных продуктов (EPL – Evaluated Products List) с указанием присвоенного им рейтинга. Поэтому, вместо того, чтобы просто доверять маркетологам производителя, вы, как покупатель, можете поверить словам независимой третьей стороны, полностью протестировавшей продукт. Для этого вы можете воспользоваться EPL.

Процесс оценки является очень трудоемким и дорогостоящим для производителя. Не каждый производитель проводит свои продукты через это, поскольку это дорого и, к тому же, отодвигает дату выпуска продукта на рынок. Обычно, производитель проводит свои продукты через этот процесс, только если основная часть его потенциальных покупателей ориентируется на рейтинг гарантий при выборе продукта. В США Министерство обороны является самым крупным покупателем, поэтому многие производители проводят свои основные продукты через процесс оценки, в надежде, что их купит Министерство обороны (или кто-то другой).

5.2. Оранжевая Книга

Министерством обороны США был разработан стандарт TCSEC (Trusted Computer System Evaluation Criteria), называемый «Оранжевой книгой», который используется для оценки операционных систем, приложений и других продуктов. Покупатели используют рейтинг гарантий в качестве критерия, являющегося метрикой при сравнении различных продуктов. Он также предоставляет конкретные требования для разработчиков, чтобы они могли учесть их при разработке своих продуктов.

Оранжевая книга используется для оценки продукта – действительно ли он имеет заявленные производителем свойства безопасности, и подходит ли продукт для выполнения определенных функций и определенных вариантов применения. Оранжевая книга используется для анализа функциональности, эффективности и гарантий продукта в процессе оценки. Она использует классы, содержащие типовые шаблоны требований безопасности.

TCSEC предоставляет систему классификации, иерархически разделенную на следующие уровни гарантий:

- **A** – Проверенная защита (verified protection)
- **B** – Мандатная защита (mandatory protection)
- **C** – Дискреционная защита (discretionary protection)
- **D** – Минимальная безопасность (minimal security)

Уровень классификации «A» представляет собой максимальный уровень гарантий, «D» – минимальный.

Каждый уровень может иметь один или несколько пронумерованных классов с соответствующими наборами требований, которые должны выполняться системой для достижения этого конкретного рейтинга. Классы с более высокими номерами предоставляют более высокую степень доверия и гарантий. Так, например, класс «B2» более доверенный, чем класс «B1».

Критерии включают в себя 4 основных раздела: политика безопасности, подотчетность, гарантии и документация. Но в действительности они делятся на 7 различных областей:

- **Политика безопасности.** Политика должна быть ясной, однозначной и реализованной механизмами системы.
- **Идентификация.** Отдельные субъекты должны быть уникально идентифицированы.
- **Метки.** Метки управления доступом должны быть надлежащим образом связаны с объектами.
- **Документация.** Должна быть предоставлена документация по тестированию, конструкторская документация, технические требования, руководства для пользователя, учебная документация.
- **Подотчетность.** Данные аудита должны сохраняться и защищаться для обеспечения подотчетности.
- **Гарантии жизненного цикла.** Должна существовать возможность протестировать по отдельности программное обеспечение, аппаратное обеспечение и прошивки, чтобы убедиться, что каждый из этих компонентов эффективно реализует политику безопасности в течение своего жизненного цикла.
- **Непрерывная защита.** Механизмы безопасности и система в целом должны всегда работать предсказуемо и приемлемо в различных ситуациях.

Эти области оцениваются независимо, но присваиваемый в конечном итоге рейтинг не учитывает все эти различные области по отдельности. Рейтинг – это их общая сумма.

Каждый уровень и класс включает в себя требования предыдущего уровня или класса. Это означает, что «C2» должен соответствовать требованиям своих критериев, а также всем требованиям критериев «C1», а «B3» помимо своих требований должен удовлетворять требованиям «C1», «C2», «B1» и «B2». Каждый уровень или класс вносит свой вклад в требования безопасности и ожидает выполнения всех требований всех предыдущих классов и разделов.

Разве Оранжевая книга еще актуальна? Мы перешли от Оранжевой книги к Общим критериям, поэтому возникает резонный вопрос – зачем изучать Оранжевую книгу? Сам по себе факт перехода не сделал Оранжевую книгу неважной. Это была первая эволюция критериев, и она использовалась 20 лет. Многие основные термины и концепции произошли из Оранжевой книги. И у нас все еще есть много продуктов с рейтингами Оранжевой книги, которые, в конечном счете, пройдут оценку по Общим критериям. Экзамен CISSP неуклонно переходит от Оранжевой книги к Общим критериям, но этот переход еще не завершен.

Уровень D: Минимальная защита

На уровне «D» есть только один класс. Он зарезервирован для систем, которые проходили процесс оценки, но не смогли удовлетворить критериям и требованиям более высоких уровней.

Уровень C: Дискреционная защита

Уровень «C» содержит два отдельных класса рейтинга гарантий, которые описаны далее. Более высокий номер рейтинга гарантий означает более высокий уровень защиты.

C1: Дискреционное обеспечение безопасности (Discretionary Security Protection).

Дискреционное управление доступом основано на отдельных людях и/или группах. Оно требует разделения пользователей и информации, проведения идентификации и аутентификации отдельных сущностей. Управление доступом необходимо, чтобы пользователи могли быть уверены, что их данные не могут быть получены или повреждены неуполномоченными лицами. Архитектура системы должна обеспечивать защищенный домен выполнения, обеспечивающий невозможность негативного воздействия на привилегированные системные процессы со стороны менее привилегированных процессов. Должны существовать определенные способы проверки функциональной целостности системы. Требования к документации включают обязательное наличие конструкторской документации, показывающей, как защитные механизмы встроены в систему, документации по тестированию (планы и результаты тестирования), описания возможностей системы, позволяющие покупателю понять, как правильно установить и настроить систему, а также учебную документацию для пользователей.

Этот рейтинг требуется для среды, в которой пользователи обрабатывают информацию одного уровня критичности, поэтому строгое управление доступом и средства аудита в ней не требуются. Это доверенная среда с низким уровнем безопасности.

C2: Управляемая защита доступа (Controlled Access Protection). Пользователи должны быть индивидуально идентифицированы для обеспечения более точного управления доступом и функций аудита. Механизмы логического управления доступом используются для выполнения аутентификации и однозначной идентификации каждого пользователя. Существенные с точки зрения безопасности события журналируются, записи о них защищены от несанкционированной модификации. Архитектура должна обеспечивать изоляцию ресурсов (или объектов), предоставляя надежную защиту, а также надлежащим образом журналировать любые производимые с ними действия. Должна применяться концепция обеспечения безопасности при повторном использовании объектов,

предусматривающая, что в любом хранилище данных не должно оставаться остаточной информации после его освобождения для возможности использования другим субъектом. Например, если субъект использует сегмент памяти, то после окончания его использования, этот сегмент не должен содержать никакой информации. Это справедливо для носителей информации, используемых объектов, создаваемых временных файлов – все данные должны эффективно затираться, когда субъект перестает использовать это хранилище.

Этот класс требует применения более детализированных методов, обеспечивающих управление доступом. Система должна реализовывать строгие процедуры входа в систему и предоставлять возможности принятия решений по запросам субъектов на доступ к объектам. Система «C2» не может гарантировать невозможность своей компрометации, но она предоставляет уровень защиты, который существенно усложняет попытки взлома.

Системы с рейтингом «C2» требуются для среды, в которой пользователи являются доверенными, но требуется определенный уровень подотчетности. «C2», в целом, выглядит наиболее разумно для коммерческих приложений, однако уровень его защиты сравнительно невелик.

Уровень В: Мандатная защита

Мандатное управление доступом реализуется посредством использования меток безопасности. Архитектура основана на модели безопасности Bell-LaPadula, должны быть доступны доказательства реализации монитора обращений.

B1: Маркированная безопасность (Labeled Security). Каждый объект данных должен содержать метку классификации, а каждый субъект должен иметь метку допуска. Когда субъект пытается получить доступ к объекту, система должна сравнить метки безопасности субъекта и объекта, чтобы убедиться, что запрошенные действия допустимы. Выходящие из системы данные также должны содержать правильную метку безопасности. Политика безопасности основана на неформальных утверждениях, а технические условия и конструкторская документация проанализированы и проверены.

Этот рейтинг безопасности предназначен для сред, в которых обрабатываются классифицированные данные.

ПРИМЕЧАНИЕ. Метки безопасности не требуются при рейтинге безопасности ниже «В». Поэтому например, «C2» не требует меток безопасности, а «B1» – требует.

B2: Структурированная защита (Structured Protection). Политика безопасности явно определена и документирована, структура системы и ее реализация подвергнуты исчерпывающему анализу и детальным тестовым процедурам. Этот класс требует наличия более строгих механизмов аутентификации и строго определенных интерфейсов между уровнями. Субъекты и устройства требуют наличия меток, а система не должна допускать наличия скрытых каналов. Должен существовать доверенный способ (trusted path) входа в систему и прохождения аутентификации, который обеспечивает прямое взаимодействие субъектов с приложениями или операционной системой, а также отсутствием недокументированных способов доступа (backdoor, trapdoor). Отсутствуют способы обхода или компрометации этого доверенного коммуникационного канала. Существует разделение функций администратора и оператора в рамках системы для обеспечения более доверенного и защищенного функционирования. Должно быть обеспечено отдельное адресное пространство для изоляции процессов и проведен анализ скрытых каналов. Этот класс увеличивает гарантии, расширяя требования к структуре системы.

Системы «B2» требуются для среды, в которой обрабатываются критичные данные, требующие высокого уровня безопасности. Такая среда требует использования систем, относительно стойких к попыткам проникновения и компрометации.

В3: Домены безопасности (Security Domains). В этом классе обеспечена еще большая детализация для каждого защитного механизма, а программный код, не требующийся для поддержки политики безопасности, исключен. Структура и реализация системы не должны быть очень сложными, поскольку сложность системы увеличивает требования к уровню подготовки людей, которые должны будут ее тестировать, поддерживать и настраивать, в связи с чем безопасность системы в целом может быть подвержена дополнительным угрозам. Компоненты монитора обращений должны быть достаточно небольшими, чтобы они могли быть надлежащим образом протестированы, также они должны быть устойчивы к внешним воздействиям. Роль администратора безопасности должна быть явно определена, система должна быть способна восстанавливаться после сбоев без снижения уровня безопасности. Процесс запуска системы и загрузки ее операционной системы и компонентов должен выполняться в изначально безопасном состоянии, чтобы обеспечить невозможность использования слабостей системы на этом этапе.

Системы «В3» требуются для высоко защищенной среды, которая обрабатывает очень критичную информацию. Системы должны быть высоко устойчивы к попыткам проникновения.

Уровень А: Проверенная защита

Формальные методы используются для обеспечения того, что все субъекты и объекты управляются средствами дискреционного и мандатного управления доступом. Проектирование, разработка, внедрение и документация рассматриваются формально и детально. Механизмы безопасности систем «В3» и «А1» отличаются несильно, но в «А1» применяются гораздо более структурированные и строгие процедуры оценки способов проектирования и разработки системы.

А1: Проверенная структура (Verified Design). Архитектура и возможности защиты не сильно отличаются от систем с рейтингом «В3», но гарантии систем «А1» выше, чем систем «В3», из-за более формального подхода к проектированию системы «А1», разработки ее технических требований и более детальной техники проведения проверки. Для проверки эквивалентности между техническими требованиями ТСВ и моделью политики безопасности используются формальные техники. При разработке систем «А1» реализуется более строгое изменение конфигурации, может быть проверена структура в целом. Часто даже способ доставки системы заказчику внимательно исследуется, чтобы убедиться в отсутствии компрометации системы до ее внедрения в работу.

Тип среды, который требует систем «А1», является самым безопасным среди безопасных сред. Этот тип среды пригоден для совершенно секретной информации и гарантирует невозможность использования системы, минуя строгую аутентификацию, ограничения и аудит.

ПРИМЕЧАНИЕ. TCSEC учитывает конфиденциальность, но не целостность. Функциональность механизмов безопасности и гарантии этих механизмов не оцениваются отдельно, они комбинируются и оцениваются как одно целое.

Ссылки по теме:

- Trusted Computer System Evaluation Criteria (Orange Book), U.S. Department of Defense (Dec. 26, 1985)
- Lecture notes for Cryptography and Network Security, Part 7, “Trusted Computer Systems,” William Stallings, lecturer, Dr. Lawrie Brown, author

5.3. Оранжевая книга и Радужная серия

Оранжевая книга в основном учитывает правительственные и военные требования и ожидания для их компьютерных систем. Многие люди видят определенные недостатки в

Оранжевой книге в части безопасности, особенно когда применяют эти требования в коммерческой области, а не военной организации. Ниже перечислены основные проблемы, возникающие у практиков при использовании Оранжевой книги:

- Она применима к операционным системам, но не другим системам, таким как сети, базы данных и т.д.
- Она фокусируется в основном на одном атрибуте безопасности – конфиденциальности, но не учитывает целостность и доступность.
- Она работает с правительственной классификацией, а не с классификацией, применяемой в коммерческих компаниях.
- Она имеет относительно небольшое количество рейтингов, в связи с чем множество различных аспектов безопасности не оцениваются независимо.

Оранжевая книга концентрируется на контроле того, как пользователи получают доступ к информации, но игнорирует контроль того, что они делают с ней после авторизации, хотя авторизованный пользователь имеет возможность нанести (и обычно наносит) больше вреда, чем внешний атакующий. Коммерческие компании больше заботятся о целостности и доступности своих данных, а военные организации ставят во главу угла конфиденциальность. В связи с этим различием целей Оранжевая книга идеально подходит только для оценки средств для правительственных и военных систем.

Поскольку Оранжевая книга фокусируется на операционных системах, множество других областей безопасности она не учитывает. Оранжевая книга предоставляет широкую платформу для построения и оценки доверенных систем, но оставляет без ответа множество вопросов, не касающихся операционных систем. Много книг было написано для расширения границ Оранжевой книги в другие области безопасности. Эти книги предоставляют детальную информацию и интерпретацию определенных требований Оранжевой книги и описывают процесс оценки. Все эти книги вместе называются *Радужной серией* (т.к. их обложки имеют различные цвета).

Для пояснения каждой книги и ее использования, пожалуйста, ознакомьтесь со следующими ссылками.

Ссылки по теме:

- Rainbow Series
- Rainbow Series and related documents from the Federation of American Scientists

5.4. Красная книга

Оранжевая книга учитывает односистемную безопасность, но сети являются комбинацией систем, и каждая сеть нуждается в обеспечении безопасности, не требуя при этом полного доверия от всех и каждой подключенных к ней систем. *Трактовка доверенных сетей* (TNI – Trusted Network Interpretation), также называемая *Красной книгой*, учитывает аспекты оценки безопасности для сети и сетевых компонентов. Она учитывает изолированные локальные (LAN) сети и глобальные сети (WAN).

Как и Оранжевая книга, Красная книга не содержит конкретных деталей по вопросам реализации механизмов безопасности, вместо этого она предоставляет основу для обеспечения безопасности различных типов сетей. Сеть имеет политику безопасности, архитектуру и дизайн – как и операционная система. Субъекты, использующие доступ к объектам в сети, должны управляться, отслеживаться и проверяться. В сети субъектами могут быть рабочие станции, а объектами могут быть сетевые сервисы на сервере.

Красная книга оценивает степень конфиденциальности данных и операций, происходящих в

сети и сетевых системах. Данные и метки должны быть защищены от несанкционированного изменения, должна обеспечиваться целостность передаваемой информации. Исходные и целевые механизмы, используемые для обмена сообщениями, должны быть оценены и протестированы, чтобы обеспечить невозможность модификации.

Безопасность сети обеспечивается такими средствами, как шифрование и протоколы, и Красная книга измеряет их функциональность, стойкость и гарантии.

Ниже приведен перечень вопросов безопасности, учтенных в Красной книге:

- **Целостность коммуникаций**

- **Аутентификация.** Защищает от атак маскарадинга и повторного воспроизведения (playback attack). Механизмы включают цифровую подпись, шифрование, штампы времени и пароли.
- **Целостность сообщений.** Защищает заголовок протокола, информацию маршрутизации и содержимое пакета от модификации. Механизмы включают аутентификацию и шифрование сообщений.
- **Невозможность отказа от авторства.** Гарантирует, что отправитель не может отказаться от факта отправки сообщения. Механизмы включают шифрование, цифровую подпись и подтверждение подлинности (notarization).

- **Предотвращение отказа в обслуживании**

- **Непрерывность деятельности.** Обеспечивает доступность сети даже в случае атаки. Механизмы включают применение отказоустойчивых и избыточных систем, а также средства перенастройки сетевых параметров в случае аварийной ситуации.
- **Управление сетью.** Мониторинг производительности сети, выявление атак и сбоев. Механизмы включают компоненты, позволяющие сетевому администратору отслеживать и ограничивать доступ к ресурсам.

- **Защита от компрометации**

- **Конфиденциальность данных.** Защищает данные от неавторизованного доступа в процессе передачи. Механизмы включают управление доступом, шифрование и физическую защиту кабелей.
- **Конфиденциальность потока трафика.** Предотвращает неавторизованный доступ к информации о маршрутизации или частоте выполнения коммуникаций посредством анализа трафика. Механизмы включают «разбавление» (padding) сообщений, отправку «шума» и «ложных» сообщений.
- **Избирательная маршрутизация.** Маршрутизирует сообщения способом, исключая некоторые специфические угрозы. Механизмы включают конфигурацию сети и таблицы маршрутизации.

Оценка гарантий производится путем сравнения реальной работы продукта с теоретической – как он должен работать, а также путем тестирования конфигураций во множестве различных сценариев, оценки практик создания системы, проверки заявлений безопасности и их обоснованности.

TCSEC введен в 1985 году и отменен в декабре 2000 года. Это был первый методический и логичный набор стандартов разработки безопасных компьютерных систем. Он оказал большое влияние на многие страны, которые разработали свои стандарты оценки на основе рекомендаций TCSEC. В конечном итоге TCSEC был заменен Общими критериями.

5.5. ITSEC

ITSEC (Information Technology Security Evaluation Criteria) – это первая попытка создания единого стандарта для оценки атрибутов безопасности компьютерных систем и продуктов рядом европейских стран. США использовали Оранжевую книгу и Радужную серию, а Европа применяла ITSEC для оценки и установки рейтингов компьютерных систем. (Сегодня все мигрировали на Общие критерии, описанные в следующем разделе).

ITSEC оценивает два основных атрибута защитных механизмов систем: функциональность и гарантии. При оценке функциональности защитных механизмов системы, проводится проверка и оценка предоставляемых субъектам сервисов (механизмов управления доступом, средств аудита, аутентификации и т.д.). Функциональность защитных механизмов может сильно различаться, так как системы разрабатываются по-разному, чтобы предоставить различную функциональность пользователям. После оценки функциональности она тестируется, чтобы понять, обеспечивают ли эти функции защиту системы в той мере, в которой утверждает производитель. Гарантии, с другой стороны, это степень доверия защитным механизмам, их эффективность и возможность работы согласованным образом. Гарантии в основном проверяются путем анализа практик разработки, документации, управления настройками, тестирования механизмов.

Возможно существование двух защитных механизмов систем, предоставляющих одинаковый тип функциональности, но имеющих существенно различающийся уровень гарантий. Это связано с тем, что предоставляющие функциональность механизмы были разработаны, спроектированы и внедрены по-разному. Например, системы А и В могут иметь защитные механизмы, предоставляющие одинаковую функциональность аутентификации, и в этих условиях оба продукта имеют одинаковый рейтинг по функциональности. Но разработчики системы А были неряшливы и беззаботны при разработке механизма аутентификации и поэтому их продукт получил меньший рейтинг по уровню гарантий. ITSEC разделяет эти два атрибута (функциональность и гарантии) и присваивает им рейтинг по-отдельности, тогда как TCSEC рассматривает их совместно и присваивает им один рейтинг (от «D» до «A1»).

Следующий список показывает различные аспекты функциональности и гарантий, которые тестируются в процессе оценки.

- Требования к функционалу безопасности
- Идентификация и аутентификация
- Аудит
- Использование ресурсов
- Доверенные пути/каналы
- Защита пользовательских данных
- Управление безопасностью
- Доступ к продукту
- Коммуникации
- Защита персональных данных
- Защита функций безопасности продукта
- Поддержка криптографии
- Требования к гарантиям безопасности
- Инструкции и руководства

- Управление конфигурациями
- Оценка уязвимостей
- Доставка и функционирование
- Поддержка жизненного цикла
- Обеспечение гарантий
- Разработка
- Тестирование

Рассмотрим снова наш пример, когда две системы предоставляют одинаковую функциональность (в отношении механизмов защиты), но имеют различный уровень гарантий. Используя подход TCSEC, различия в уровне гарантий сложно увидеть, т.к. функциональность и гарантии рассматриваются совместно. В подходе ITSEC рейтинг функциональности присваивается отдельно от гарантий, делая отличия в уровне гарантий более заметными. В критериях ITSEC классы рейтинга «F1» – «F10» относятся к функциональности механизмов безопасности, а «E0» – «E6» – к гарантиям этих механизмов.

Таким образом, отличие между ITSEC и TCSEC заключается в том, что TCSEC оценивает функциональность и гарантии совместно, а ITSEC рассматривает два этих атрибута раздельно. Другим отличием является тот факт, что ITSEC разработан для предоставления большей гибкости, чем TCSEC, а также то, что ITSEC учитывает целостность, доступность и конфиденциальность, в то время как TCSEC – только конфиденциальность. ITSEC также учитывает сетевые системы, а TCSEC – только автономные системы.

Таблица 3-2 показывает взаимосвязь между двумя схемами оценки.

ITSEC	TCSEC
E0	= D
F1 + E1	= C1
F2 + E2	= C2
F3 + E3	= B1
F4 + E4	= B2
F5 + E5	= B3
F5 + E6	= A1
F6	= Системы, предоставляющие высокий уровень целостности
F7	= Системы, предоставляющие высокий уровень доступности
F8	= Системы, предоставляющие целостность данных в процессе коммуникаций
F9	= Системы, предоставляющие высокий уровень конфиденциальности (например, криптографические устройства)
F10	= Сети с высокими запросами к конфиденциальности и целостности

Таблица 3-2. Связь между ITSEC и TCSEC

Как вы могли заметить, большинство рейтингов ITSEC связаны с рейтингами TCSEC, но в ITSEC добавлены рейтинги «F6» – «F10» для специфических нужд потребителей, которые TCSEC не учитывает.

ITSEC – это критерии для операционных систем и других продуктов, которые рассматриваются как отдельные объекты оценки (TOE – Target of Evaluation). Поэтому, если вы читаете литературу, обсуждающую ITSEC-рейтинги продуктов, и в ней заявлено, что TOE имеет рейтинг «F1» и «E5», вы знаете, что TOE – это продукт, который был оценен и что он указывает на низкий рейтинг функциональности и высокий рейтинг гарантий.

Рейтинги относятся к гарантиям, которые являются корректностью и эффективностью механизмов безопасности, а также к функциональности. Анализ корректности

рассматривает, как ТОВ был построен и внедрен, при этом анализируется архитектура, реализация политики безопасности механизмами безопасности, документация по эксплуатации, среда. Анализ эффективности рассматривает недостатки процесса разработки и эксплуатации, простоту их использования, чтобы корректные настройки безопасности не мешали производительности. Функциональность рассматривается в терминах целей безопасности системы, функций и механизмов безопасности. Функциональностью может быть, например, идентификация и аутентификация, управление доступом, подотчетность, журналирование, повторное использование объектов, точность, надежность сервисов и обмен данными.

Ссылки по теме:

- Criteria and Methods of Evaluations of Information Systems

5.6. Общие критерии

Оранжевая книга и Радужная серия предоставляют схемы оценки, крайне негибкие для бизнеса. ITSEC пытается предоставить более гибкий подход, разделяя атрибуты функциональности и гарантий и учитывая оценку всей системы в целом. Однако эта гибкость добавляет сложности, так как оценщики могут смешивать и сочетать рейтинги функциональности и гарантий, что приведет к слишком большому количеству классификаций и их ненадежности. Следующей попыткой сделать все правильно стало создание более эффективных и приемлемых критериев оценки – **Общих критериев** (Common Criteria).

В 1990 году Международная организация по стандартизации (ISO – International Organization for Standardization) выявила наличие потребности в международном стандарте критериев оценки, который мог бы использоваться в глобальных масштабах. Проект по созданию Общих критериев стартовал в 1993 году, когда несколько организаций собрались для объединения и согласования существующих и разрабатываемых критериев оценки (TCSEC, ITSEC, STCPEC и Федеральных критериев). В результате сотрудничества между национальными организациями по стандартизации США, Канады, Франции, Германии, Великобритании и Нидерландов были разработаны Общие критерии.

Выгода от получения глобального, принятого на мировом уровне набора критериев заключается в уменьшении сложности рейтингов для покупателей, исключении необходимости понимания определения и смысла различных рейтингов в рамках различных схем оценки. Это также помогает производителям, т.к. теперь они могут обеспечивать соответствие одному конкретному набору требований, даже если они хотят продавать свой продукт на международном рынке, а не пытаться обеспечить соответствие нескольким различным рейтингам с различными требованиями и правилами.

Оранжевая книга оценивает все системы на соответствие модели Bell-LaPadula. Общие критерии предоставляют значительно большую гибкость, оценивая продукты на соответствие профилям защиты, которые структурированы для учета потребностей безопасности в реальном мире. Так, если Оранжевая книга говорит: *"Все марш в этом направлении, в этой форме и по этому пути!"*, Общие критерии спрашивают: *"Так, какие угрозы сегодня стоят перед нами и каков лучший вариант борьбы с ними?"*.

В модели Общих критериев выполняется оценка продукта, после чего ему присваивается **Оценочный уровень гарантий** (EAL – Evaluation Assurance Level). Исчерпывающее и строгое тестирование, основанное на ориентированных на детали задачах, повышает уровень гарантий. Общие критерии имеют семь уровней гарантий в диапазоне от «EAL1», где проводится тестирование функциональности, до «EAL7», где выполняется исчерпывающее тестирование и проверяется структура системы. Различные пакеты EAL приведены в списке:

- **EAL 1.** Функционально протестировано
- **EAL 2.** Структурно протестировано
- **EAL 3.** Методически протестировано и сверено
- **EAL 4.** Методически проработано, протестировано и проанализировано
- **EAL 5.** Полуформально проработано и протестировано
- **EAL 6.** Полуформально проработано и протестировано с подтверждением
- **EAL 7.** Формально проработано и протестировано с подтверждением

ПРИМЕЧАНИЕ. Если система является «формально проработанной», это означает, что она основана на модели, которая может быть математически доказана.

Общие критерии используют **профили защиты** в процессе оценки. Это механизм, который используется для описания потребностей реального мира в отношении продуктов, которые сейчас отсутствуют на рынке. Профиль защиты содержит набор требований по безопасности, их смысл и обоснования, а также соответствующий рейтинг EAL, который требуется для продукта. Профиль защиты описывает предположения о среде, цели, ожидаемый уровень функциональности и гарантий. Каждая существенная угроза указывается с пояснениями, как она должна контролироваться посредством конкретных задач. Профиль защиты также обосновывает уровень гарантий и требования к стойкости каждого защитного механизма.

Профиль защиты дает покупателю (или другим) возможность указать специфические потребности в безопасности для решения определенной проблемы безопасности. Если кто-то определил потребности в безопасности, но они не реализованы в имеющихся на рынке продуктах, этот человек может написать профиль защиты, описывающий продукт, решающий эту реальную проблему. Профиль защиты нужен для указания необходимых целей и защитных механизмов для достижения необходимого уровня безопасности, а также списка вещей, которые могут «пойти не так» в процессе разработки системы такого типа. Этот список используется инженерами, разрабатывающими систему, и затем оценщиками, чтобы убедиться, что инженеры выполнили свою работу в полном объеме.

Общие критерии были разработаны, чтобы придерживаться классов оценки, сохраняя при этом некоторую степень гибкости. Профили защиты были разработаны для описания функциональности, гарантий, описаний и рациональных требований к продукту.

Как и другие оценочные критерии до них, Общие критерии работают, чтобы ответить на два основных вопроса об оцениваемом продукте: что делают его механизмы безопасности (функциональность) и насколько в них можно быть уверенным (гарантии)? Эта система создает основу, позволяющую покупателям ясно описывать свои проблемы безопасности, разработчикам – описывать свои решения по безопасности для решения этих проблем, а оценщикам – для недвусмысленного определения того, что продукт в действительности выполняет.

Профиль защиты содержит следующие пять разделов:

- **Описательные элементы.** Указывает название профиля и описание проблемы безопасности, которая должна быть решена.
- **Логическое обоснование.** Обосновывает профиль и дает более детальное описание реальной проблемы, которая должна быть решена. Предполагаемые среда и порядок использования, а также угрозы, показываются вместе с описанием политик безопасности, которые должен поддерживать продукт или система, чтобы соответствовать данному профилю.
- **Функциональные требования.** Устанавливают границы защиты, означающие, что угрозы или компрометации в рамках этих границ должны быть предотвращены.

Продукт или система должны реализовать границы, установленные в этом разделе.

- **Требования к гарантиям разработки.** Идентифицирует конкретные требования, которым продукт или система должны удовлетворять на этапе разработки – от момента начала проектирования до момента внедрения.
- **Требования к оценке гарантий.** Устанавливают тип и интенсивность оценки.

Процесс оценки – это только один этап определения функциональности и гарантий продукта. Если продукт достиг некоторого рейтинга, это применимо только для этой конкретной версии и только для определенной конфигурации продукта. Таким образом, если компания покупает межсетевой экран, предоставляющий высокий рейтинг гарантий, она не имеет гарантий, что следующая версия его программного обеспечения будет иметь такой же рейтинг. Следующая версия должна будет пройти через процесс пересмотра оценки. С другой стороны, если компания купила межсетевой экран и установила его с конфигурацией, которая не была рекомендована, уровень безопасности, которого хотела достичь компания, может снизиться. Таким образом, все эти рейтинги являются формализованным методом обзора систем в процессе их оценки в лаборатории. Когда продукт внедрен в реальную среду, все факторы, отличающиеся от использовавшихся при установке рейтинга, должны быть учтены и оценены для обеспечения надлежащей защиты ресурсов и окружения.

ПРИМЕЧАНИЕ. Когда продукту присвоен рейтинг гарантий, это означает, что потенциально он может обеспечить этот уровень защиты. Покупатель должен правильно настроить продукт, чтобы реально получить этот уровень защиты. Производитель должен предоставить необходимую документацию по настройке и объяснить покупателю, как правильно настроить продукт и сохранить правильную настройку в течение всего времени его эксплуатации.

Различные компоненты Общих критериев перечислены и описаны ниже:



- **Профиль защиты.** Описание необходимого решения безопасности.
- **Объект оценки.** Продукт, который предлагается в качестве необходимого решения безопасности.
- **Цель безопасности.** Написанное разработчиком пояснение функциональности и гарантий безопасности, реализованных необходимым решением безопасности; другими словами: «это то, что наш продукт делает и как он это делает».
- **Пакеты EAL.** Требования в отношении гарантий и функциональности, объединенные в пакет для повторного использования. Этот компонент описывает, что должно быть учтено для достижения продуктом определенного рейтинга EAL.

Ссылки по теме:

- Computer Security Resource Center (CSRC) Common Criteria for IT Security Evaluation
- Common Criteria
- Common Criteria overview, Rycombe Consulting

6. Сертификация vs. Аккредитация

Мы рассмотрели различные типы критериев оценки, позволяющие оценить систему для присвоения ей соответствующего рейтинга. Это очень формализованный процесс, следуя которому оцениваемой системе или продукту присваивается определенный рейтинг, после чего сведения об этой системе (продукте) размещаются в EPL. Покупатель может проверить этот список и сравнить различные продукты и системы, чтобы увидеть, как соотносятся их

рейтинги возможностей защиты. Однако когда покупатель приобретает этот продукт и устанавливает его в свою среду, безопасность не гарантирована. Обеспечение безопасности включает в себя множество аспектов, среди которых администрирование системы, физическая безопасность, процесс установки и настройки механизмов непосредственно в среде, а также другие действия. Чтобы сказать, что система защищена, все эти аспекты должны быть учтены. Рейтинг – это только один из элементов в головоломке, называемой безопасностью.

6.1. Сертификация

Сертификация – это всесторонняя техническая оценка компонентов безопасности и их соответствия цели аккредитации. Процесс сертификации может использовать оценку мер безопасности, анализ рисков, проверку, тестирование и методы аудита для оценки пригодности конкретной системы. Предположим, что Дэн – офицер безопасности компании, который купил новые системы для использования в процессе обработки конфиденциальных данных. Он хочет знать, подходят ли эти системы для этих задач и предоставляют ли они необходимый уровень защиты. Он также хочет убедиться, что они совместимы с имеющимся окружением, не снизят производительность, не создадут новых угроз – по сути, он хочет знать, подходят ли эти системы его компании. Он может заплатить специализирующейся на этих вопросах компании, которая выполнит необходимые процедуры сертификации этих систем, либо он может это сделать своими силами. В рамках сертификации команда оценщиков анализирует оборудование, прошивки, тестирует конфигурации программного обеспечения, структуру, реализацию, системные процедуры, управление на физическом и коммуникационном уровнях.

Цель процесса сертификации – гарантировать, что сеть, продукт или система подходят задачам покупателя. Различные покупатели могут использовать один и тот же продукт по различным причинам, а их среда может иметь различный уровень угроз. Поэтому не каждый продукт хорошо подойдет каждому конкретному покупателю (хотя продавцы, конечно, будут пытаться убедить вас в обратном). Продукт предоставляет надлежащую функциональность и безопасность покупателям, цели которых соответствуют целям процесса сертификации.

Процесс сертификации и соответствующая документация могут показать продукт хорошим, плохим, ужасным, описать, как он работает в рамках данной среды. Далее, Дэн возьмет эти результаты и покажет своему руководству для проведения процесса аккредитации.

6.2. Аккредитация

Аккредитация – это формальное принятие руководством адекватности безопасности системы в целом и ее функциональности. Руководству (или ответственному лицу) представляется информация сертификации, ответы на их вопросы, обзоры отчетов и исследований, чтобы руководство решило – принять ли продукт, и нужны ли какие-либо корректирующие действия. Если безопасность системы в целом руководство устраивает, оно формально принимает ее. Делая это, руководство утверждает, что оно понимает уровень защиты, предоставляемый системой в текущей среде компании, и понимает риски безопасности, связанные с установкой и поддержкой этой системы.

Не нужно больше ломать карандаши. Многие компании стали относиться к процессу аккредитации более серьезно, чем раньше. К сожалению, иногда после завершения процесса сертификации и передачи документации руководству для анализа и принятия решения, руководство просто не глядя подписывает необходимые документы, не понимая в действительности, что оно подписывает. Аккредитация означает, что руководство принимает риск, связанный с разрешением внедрить новый продукт в среду компании. Когда происходит серьезная проблема с безопасностью, последней инстанцией является человек, который подписал документ. Как только такие руководители понимают, что они несут ответственность за то, что они подписали, а также узнают, как много законов устанавливают персональную ответственность

руководителя за безопасность, они гораздо реже ломают карандаши об аккредитационные бумаги. Хотя, увы, полностью исключить такие случаи все равно нельзя.

ПРИМЕЧАНИЕ. Сертификация – это технический обзор, обеспечивающий оценку механизмов безопасности и их эффективности. Аккредитация – это официальное принятие руководством информации, выявленной в процессе сертификации.

Поскольку программное обеспечение, системы и среды постоянно меняются и развиваются, сертификацию и аккредитацию следует также продолжать применять. При любом существенном дополнении в программном обеспечении, изменении в системе или изменении в среде следует инициировать новый цикл сертификации и аккредитации.

7. Открытые vs. Закрытые системы

Компьютерные системы могут разрабатываться для легкой интеграции с другими системами и продуктами (открытые системы), либо могут разрабатываться как более закрытые по своей природе и работающие только с определенным подмножеством других систем и продуктов (закрытые системы). Следующие разделы описывают различия между этими подходами.

7.1. Открытые системы

Системы, которые описываются как открытые, построены на основе стандартов, протоколов и интерфейсов, имеющих опубликованные спецификации, которые позволяют сторонним разработчикам разрабатывать дополнительные компоненты и устройства. Эта функциональная совместимость обеспечивается всеми разработчиками, которые следуют соответствующим стандартам и предоставляют интерфейсы, позволяющие любой системе без проблем взаимодействовать с другими системами, легко дополняя их.

Большинство систем, используемых в настоящее время, являются открытыми. Именно по этой причине администратор может обеспечить легкое взаимодействие в рамках одной сети компьютеров под управлением Windows NT 4, Windows 2000, Macintosh и Unix, так как их платформы являются открытыми. Если разработчик создает закрытую систему, это ограничивает ее потенциальные продажи, т.к. она пригодна только для проприетарных сред.

ПРИМЕЧАНИЕ. В Домене 09 мы рассмотрим стандарты взаимодействия, такие как CORBA, DCOM, J2EE и другие.

7.2. Закрытые системы

Системы, которые называют закрытыми, используют архитектуру не соблюдающую отраслевые стандарты. При этом функциональная совместимость не обеспечивается, стандартные интерфейсы не применяются, что не позволяет различными типам систем легко взаимодействовать между собой и дополнять возможности друг друга. Закрытая система проприетарна, что означает, что она может взаимодействовать только с такими же системами.

С другой стороны, закрытая архитектура может обеспечить более высокий уровень безопасности, т.к. она не имеет множества точек входа и функционирует в более ограниченной среде, чем открытые системы. Для закрытых систем не существует такого множества средств для обхода или нарушения механизмов безопасности, немногие люди понимают их структуру, язык, а также слабости защиты, чтобы воспользоваться ими. Большинство современных систем построено на открытой архитектуре, что позволяет им работать с другими типами систем, легко совместно использовать информацию и получать расширенную функциональность за счет дополнений, разработанных сторонними разработчиками. Однако это открытая дверь для взлома и атак.

8. Корпоративная архитектура

К настоящему моменту мы рассмотрели множество вопросов и концепций, относящихся к

операционным системам и приложениям. Мы начали с архитектуры системы, прошли через различные компоненты, которые должны быть безопасным образом встроены в систему для обеспечения необходимого (требуемого от них) уровня защиты. **Архитектура безопасности** системы – это высокоуровневый проект, использующийся в качестве основы. Архитектура необходима для обеспечения соблюдения политики безопасности системы и требуемого уровня безопасности. Прекрасно, но что такое корпоративная архитектура безопасности?

Компания всегда имеет выбор, пытаясь защитить свою среду в целом. Она может решить просто внедрить отдельные продукты в различных частях среды (это называется точечными решениями) и надеяться, что такой подход будет прекрасно работать и обеспечит равномерную безопасность среды, закроет все уязвимости компании. Либо компания может потратить время, чтобы проанализировать свою среду, понять требования бизнеса в отношении безопасности и разработать полноценную платформу и стратегию, которые являются отображением друг друга. Большинство компаний выбирают первый вариант, занимаясь «тушением локальных пожаров». Это приводит к постоянным стрессам, несоблюдению требований безопасности, делает хаос нормой жизни.

Второй подход может использоваться для определения корпоративной архитектуры безопасности, применяя ее в качестве руководства при реализации решений, чтобы убедиться, что учтено все необходимое, обеспечена стандартная защита всей среды, минимизировано количество возможных сюрпризов безопасности для компании, когда она начнет реально использовать все это. Хотя реализация корпоративной архитектуры безопасности не обещает идеальную среду, она устраняет хаос, существенно увеличивает продуктивность работы сотрудников безопасности и компании в целом, позволяет мыслить более проактивно и зрело в отношении безопасности компании.

Корпоративная архитектура безопасности (enterprise security architecture) определяет стратегию информационной безопасности, которая состоит из уровней политики, стандартов, решений, процедур и способов, которыми они соединены стратегически, тактически и операционно. Это отличается от архитектуры инфраструктуры.

Инфраструктура – это лежащие в основе технологии и аппаратное обеспечение, необходимые для поддержки корпоративной архитектуры безопасности. Просто иметь инфраструктуру (коммутаторы, кабели, маршрутизаторы, узлы и т.п.) недостаточно. Чтобы иметь корпоративную архитектуру безопасности необходимо обеспечить совместную работу всех элементов инфраструктуры взаимосвязанным и безопасным образом, для чего требуется программное обеспечение, люди и процессы. Помимо безопасности, этот тип архитектуры позволяет компаниям достигать лучшей совместимости, интеграции, легкости использования, стандартизации и управляемости.

Как вы можете узнать, что компания не имеет корпоративной архитектуры безопасности? Если вы ответите «да» на большинство следующих вопросов, значит такой архитектуры не существует.

- Выявление новых уязвимостей и воздействий от них заняло более 15 дней?
- Когда потребности пользователя в доступе возрастают в соответствии с потребностями бизнеса, администратор безопасности или сетевой администратор просто изменяет его права доступа без документально оформленного руководителем пользователя разрешения?
- После внедрения нового продукта, неожиданно появляются проблемы несовместимости, которые требуют существенное количество времени и денег для своего исправления?
- При возникновении проблем безопасности применяются индивидуальные, одноразовые, точечные решения, вместо следования стандартизированным

процедурам?

- Руководители бизнес-подразделений компании не осведомлены о своих обязанностях в области безопасности, в том числе обязанностях установленных законодательством и требованиями регуляторов?
- Понятие «критичные данные» определено в политике, однако необходимые защитные меры не полностью внедрены, а их работа не отслеживается надлежащим образом?
- Реализуются точечные решения вместо решений корпоративного уровня?
- Продолжают происходить одни и те же дорогостоящие ошибки?
- Отсутствует постоянный контакт между высшим руководством и персоналом безопасности?
- Стратегическое управление безопасностью в настоящее время недоступно, поскольку безопасность компании не анализируется и не контролируется стандартными и всеобъемлющими способами?
- Бизнес-решения принимаются без учета безопасности?
- Персонал безопасности обычно «тушит пожары», не имея времени на разработку и внедрение стратегических подходов?
- Все больше и больше сотрудников безопасности «сохнут» на работе, пьют антидепрессанты и успокоительное?

Большинство компаний имеют перечисленные выше проблемы, и при этом они фокусируются на каждой из этих проблем в отдельности, как будто они не связаны между собой. Разве CSO, CISO и/или администратор безопасности не понимает, что это просто *симптомы* опасной болезни? «Лечением» была бы разработка поэтапного плана внедрения корпоративной архитектуры безопасности. Основной целью этого плана является переход от процессов безопасности, ориентированных на технологии, к процессам безопасности, ориентированным на бизнес, связывающим административные, технические и физические защитные меры для надлежащего управления рисками и интегрирования этих процессов в ИТ-инфраструктуру, бизнес-процессы, организационную культуру компании.

Основная причина, по которой компании не разрабатывают и не внедряют корпоративную архитектуру безопасности, заключается в том, что они не полностью понимают что это такое, и это выглядит для них очень сложной задачей. Борьба с точечными «пожарами» более понятна и прямолинейна, поэтому многие компании остаются с этим простым подходом.

Если вы помните, в Домене 01 мы рассматривали внедрение программы безопасности, в которой описаны следующие задачи:

- **Планирование и Организация**
 - Получение одобрения от руководства
 - Создание Руководящего комитета по надзору
 - Оценка бизнес-драйверов
 - Создание профиля угроз компании
 - Проведение оценки рисков
 - Разработка архитектуры безопасности на организационном, прикладном, сетевом и компонентном уровнях

- Определение решений на каждом уровне архитектуры
- Получение согласия руководства на дальнейшие действия
- **Реализация**
 - Распределение ролей и обязанностей
 - Разработка и внедрение политик безопасности, процедур, стандартов, базисов и руководств
 - Выявление критичных данных на этапах хранения и передачи
 - Реализация следующих проектов:
 - Идентификация и управление активами
 - Управление рисками
 - Управление уязвимостями
 - Соответствие требованиям
 - Управление идентификацией и доступом
 - Управление изменениями
 - Жизненный цикл разработки программного обеспечения
 - Планирование непрерывности бизнеса
 - Обучение и повышение осведомленности
 - Физическая безопасность
 - Реакция на инциденты
 - Внедрение решений (административных, технических, физических) по каждому проекту
 - Разработка решений по аудиту и мониторингу для каждого проекта
 - Установка целей, соглашений об уровне обслуживания (SLA) и метрик по каждому проекту
- **Функционирование и Поддержка**
 - Соблюдение установленных процедур для обеспечения соблюдения базисных уровней в каждом реализованном проекте
 - Проведение внутренних и внешних аудитов
 - Выполнение задач, намеченных в каждом проекте
 - Управление соглашениями об уровне обслуживания по каждому проекту
- **Мониторинг и Оценка**
 - Анализ лог-файлов, результатов аудита, собранных значений метрик и SLA по каждому проекту
 - Оценка достижения целей по каждому проекту
 - Проведение ежеквартальных встреч с руководящими комитетами
 - Совершенствование действий каждого этапа и их интеграция в фазу Планирования и Организации

Это является также и основными шагами внедрения корпоративной архитектуры безопасности, поскольку программа безопасности и архитектура безопасности должны следовать одной и той же модели. Программа безопасности часто основана на административных компонентах, также как и политики, стандарты, управление рисками, безопасность персонала, классификация данных и т.д. Обычно каждая основная концепция, рассмотренная в Домене 01, находит свое отражение в виде компонента программы безопасности. Корпоративная архитектура безопасности идет глубже, чем программа безопасности, и предоставляет больше деталей. Например, если политика безопасности требует, чтобы сетевое управление доступом было реализовано и внедрено, архитектура может рассматривать схему сети, отдельные зоны сети на основе уровня доверия и бизнес-потребностей, внешние подключения, механизмы безопасности, инструменты и роли, участвующие на каждом уровне. Архитектура работает от уровня политики до уровня компонентов.

В качестве аналогии скажем, что Шеннон говорит строителю, что он хочет дом в стиле ранчо с четырьмя спальнями общей площадью 2500 квадратных футов. Это высокоуровневое описание (политика) и строитель не собирается просто импровизировать и думать, что он сможет построить дом на основании такого наброска. Строитель будет следовать плану дома (архитектура), который предоставит более детализированные требования, которые должны быть учтены для выполнения запроса Шеннона.

Почти все хорошие корпоративные архитектуры безопасности работают тем или иным образом со структурой, предоставленной Моделью Захмана (Zachman Architecture Framework). Таблица 3-3 показывает состав этой модели (более полную информацию вы можете получить на сайте www.zifa.com). Модель Захмана на протяжении многих лет используется многими организациями для построения или лучшего определения своей бизнес-среды. Эта модель не является ориентированной на безопасность, но это хороший шаблон для использования, поскольку он дает направление, как понять реальное предприятие модульным способом.

№	Уровень	Что (данные)	Как (функции)	Где (сеть)	Кто (люди)	Когда (время)	Почему (мотивация)
1	Границы контекста - Планировщик	Список важных для бизнеса вещей	Список бизнес- процессов	Список мест, где выполняются операции бизнеса	Список организаций, важных для бизнеса	Список событий, важных для бизнеса	Список бизнес- целей / стратегий
2	Концепции бизнес-модели - Владелец	Например, семантичес- кая модель или модель взаимо- действия сущностей	Например, модель бизнес- процессов	Например, модель бизнес- логистики	Например, модель документо- оборота (workflow)	Например, мастер-план	Например, бизнес-план
3	Модель системной логики - Конструктор	Например, логическая модель данных	Например, архитектура приложений	Например, распреде- ленная системная архитектура	Например, архитектура пользова- тельского интерфейса	Например, структура процессов	Например, модель бизнес- правил
4	Технологическая (физическая) модель - Проектировщик	Например, физическая модель данных	Например, проект системы	Например, технологи- ческая архитектура	Например, архитектура презентации	Например, структура управления	Например, описание правил
5	Конфигурация компонентов - Разработчик	Например, описание структуры данных	Например, программа	Например, сетевая архитектура	Например, архитектура безопаснос- ти	Например, временные «привязки»	Например, специфика- ция правил
6	Функционирую- щие экземпляры в компании - Сотрудник	Например, данные	Например, функции	Например, сеть	Например, компания	Например, графики (планы) событий	Например, стратегия

Таблица 3-3. Модель Захмана для Корпоративной архитектуры

ПРИМЕЧАНИЕ. Модель Захмана используется в качестве модели для надежной архитектуры безопасности, т.е. работающей со многими компонентами всей организации. Многие люди знакомы с техническими архитектурами безопасности, которые имеют дело просто с сетями и системами в рамках этой сети. Это две совершенно разные вещи. Надежная архитектура безопасности включает в себя техническую архитектуру и многое другое, как вы можете видеть в Таблице 3-3.

Модель Захмана является двумерной, она использует шесть основных вопросов (communication interrogatives) (Что, Как, Где, Кто, Когда и Почему), пересекающихся с различными уровнями (Планировщик, Владелец, Конструктор (архитектор), Проектировщик, Разработчик и Сотрудник), чтобы дать целостное представление о предприятии. Эта модель была разработана в 1980-х годах, она основывается на принципах классической архитектуры бизнеса, которые содержат правила, управляющие упорядоченным множеством отношений.

Исходя из опыта автора, большинство технических людей негативно относятся к моделям, таким, как эта. Они считают, что здесь слишком много работы, очень много «воды», что это не имеет прямого отношения к их задачам и т.д. Они сфокусированы на технологиях и не понимают все остальные компоненты безопасности, которые являются не менее (а, возможно, более) важными, чем технологии.

SABSA. Независимо от Модели Захмана была разработана модель и методология SABSA (Sherwood Applied Business Security Architecture – Шервудская прикладная архитектура безопасности бизнеса). Она показана в следующей таблице. Вы можете посетить сайт www.sabsa-institute.org/home.aspx, чтобы узнать больше об этом подходе. На экзамене CISSP модель SABSA не рассматривается.

№	Уровень	Что (данные)	Как (функции)	Где (сеть)	Кто (люди)	Когда (время)	Почему (мотивация)
1	Границы контекста - Планировщик	Список важных для бизнеса вещей	Список бизнес- процессов	Список мест, где выполняются операции бизнеса	Список организаций, важных для бизнеса	Список событий, важных для бизнеса	Список бизнес- целей / стратегий
2	Концепции бизнес-модели - Владелец	Например, семантичес- кая модель или модель взаимо- действия сущностей	Например, модель бизнес- процессов	Например, модель бизнес- логистики	Например, модель документо- оборота (workflow)	Например, мастер-план	Например, бизнес-план
3	Модель системной логики - Конструктор	Например, логическая модель данных	Например, архитектура приложений	Например, распреде- ленная системная архитектура	Например, архитектура пользова- тельского интерфейса	Например, структура процессов	Например, модель бизнес- правил
4	Технологическая (физическая) модель - Проектировщик	Например, физическая модель данных	Например, проект системы	Например, технологи- ческая архитектура	Например, архитектура презентации	Например, структура управления	Например, описание правил
5	Конфигурация компонентов - Разработчик	Например, описание структуры данных	Например, программа	Например, сетевая архитектура	Например, архитектура безопаснос- ти	Например, временные «привязки»	Например, специфика- ция правил
6	Функционирую- щие экземпляры в компании - Сотрудник	Например, данные	Например, функции	Например, сеть	Например, компания	Например, графики (планы) событий	Например, стратегия

Работа на корпоративном уровне требует совершенно иного мышления, чем работа исключительно на системном или техническом уровне. Мало того, что решения должны применяться ко всей компании стандартизованным образом, они должны быть связаны с потребностями бизнеса. Например, когда вы думаете об управлении доступом, а не просто о Kerberos, контроллерах домена, ACL и полномочиях, вам нужно также принимать во внимание те требования регуляторов и законодательства, которые компания должна соблюдать. Если эта компания должна соблюдать требования SOX, она нуждается в том, чтобы многие из этих процессов (управления доступом) были документированы, а доступ предоставлялся только с разрешения руководителей. Это может потребовать внедрения полноценных решений по управлению идентификацией – доменной аутентификации, реализованной в продуктах Microsoft, может быть недостаточно.

В процессе разработки корпоративной архитектуры безопасности, должны быть поняты и учтены следующие вопросы: стратегическое выравнивание, обеспечение бизнеса, улучшение процессов, эффективность безопасности.

Стратегическое выравнивание (strategic alignment) относится к обеспечению соблюдения архитектурой безопасности требований бизнес-драйверов, регуляторов и законодательства. Текущее состояние безопасности должно быть понятно, для чего должна быть проведена оценка рисков, позволяющая компании понять свои актуальные угрозы, а также свои способности бороться с этими угрозами. Следует достичь консенсуса в отношении уязвимостей и угроз компании, приемлемый для компании уровень риска должен быть установлен на основе толерантности компании к рискам. Прекрасно, но что все это означает? Попробуем выразить это проще. Стратегическое выравнивание означает «Какие активы мы должны защищать?», «Насколько наши инициативы в области безопасности связаны с реальными потребностями бизнеса?», «Что подразумевается под "достаточной безопасностью"», и «Высшее руководство участвует во всем этом?».

Результатом этих усилий является выработка согласованного текущего профиля рисков и желаемого профиля. Следует разработать трехлетний план безопасности, в котором должно быть определено, каким образом компания будет двигаться и достигать желаемого профиля. Также в этом плане следует детализировать поэтапный подход к построению безопасности (в целом) компании.

ПРИМЕЧАНИЕ. Хотя приведенная выше информация может показаться очевидной, это действительно то, что должно быть предпринято компанией в первую очередь. К сожалению, многие компании просто начинают подключать межсетевые экраны, настраивать ACL и внедрять решения по шифрованию без создания общего плана. Получается, что все заняты, но зачастую все идет различными путями, не согласовав между собой цели, над которыми должен работать каждый.

Несколькими другими вещами, которые нужно учесть на этом стратегическом этапе, являются:

- Определить заинтересованных лиц и их потребности
- Определить владельцев и ответственных за хранение, распределить ответственность
- Определить и распределить обязанности, подотчетность и полномочия

Высшее руководство должно принимать активное участие на этом этапе и предоставлять необходимые ресурсы и поддержку. Без поддержки руководства, усилия будут вялыми, и никаких реальных успехов достичь не удастся.

Глядя на часть архитектуры, связанную с **обеспечением бизнеса** (business enablement), мы должны напомнить себе, что любая коммерческая компания находится в бизнесе, чтобы заработать деньги. Не существует коммерческих компаний, единственной целью которых является обеспечение своей безопасности. Безопасность не может стоять на пути бизнес-процессов, она должна быть реализована, чтобы обеспечивать потребности бизнеса, повышать его эффективность, участвовать в создании бизнес-продуктов. Многие из критически важных бизнес-процессов имеют дело с транзакциями, целостность которых должна быть обеспечена от начала до конца. В связи с этим безопасности уделяется повышенное внимание, т.к. именно обеспечение безопасности должно позволить осуществлять соответствующую бизнес-деятельность.

Обеспечение бизнеса означает, что ключевые бизнес-процессы интегрируются в операционную модель безопасности, они основываются на стандартах и учитывают критерии толерантности рисков. Что это означает в реальной жизни? Например, бухгалтерия компании рассчитала, что если компания организует работу на дому персонала обслуживания и поддержки клиентов, это позволит компании сэкономить существенные деньги на аренде офиса, коммунальных услугах и т.д., к тому же страхование работающих на дому сотрудников стоит дешевле. Компания может перейти в эту новую модель с использованием VPN, межсетевых экранов, контентной фильтрации и т.п. Другой пример – финансовая компания хочет позволить своим клиентам просматривать информацию о своих банковских счетах и осуществлять финансовые транзакции. Она может предложить такую услугу клиентам только в том случае, если будут внедрены все необходимые механизмы безопасности. Безопасность должна помогать компании процветать, предоставляя ей механизмы, позволяющие безопасно решать новые задачи.

Часть, связанная с **улучшением процессов** (process enhancement), может быть весьма полезной для компании, если она правильно использует эту возможность. Если компания серьезно относится к обеспечению безопасности своей среды, она должна пристально взглянуть на многие бизнес-процессы, выполняющиеся на постоянной основе. Хотя при этом бизнес-процессы рассматриваются через призму безопасности, это является также прекрасной возможностью для усовершенствования и улучшения этих процессов с целью повышения их эффективности. Если вы проанализируете бизнес-процессы, происходящие в

компаниях любого типа, часто вы найдете факты дублирования одинаковых действий, выполняющиеся вручную операции, которые можно легко автоматизировать и т.п. Помимо обеспечения безопасности этих процессов, вы можете предложить пути повышения их эффективности, сокращения времени и ресурсов, необходимых для их выполнения. Это обычно называют реинжинирингом процессов.

Когда компания разрабатывает проекты по обеспечению своей безопасности, эти проекты должны интегрироваться в бизнес-процессы, чтобы быть эффективными. Это позволит интегрировать безопасность в жизненные циклы систем и ежедневные операции. Также это может позволить компании параллельно усовершенствовать и откалибровать управление процессами.

Эффективность безопасности (security effectiveness) связана с метриками, соблюдением требований SLA, возвратом инвестиций (ROI – return on investment), соблюдением набора базисов и предоставлением руководству отчетов с указанием всех основных показателей безопасности. Это позволяет определить, насколько полезны текущие решения по безопасности и архитектура безопасности при их комплексной работе.

Многие компании уже дошли до этой точки в своей архитектуре, поскольку существует необходимость в том, чтобы убедиться, что внедренные контрмеры обеспечивают необходимый уровень защиты, а ограниченный бюджет используется надлежащим образом. После установки базисов можно разработать метрики для проверки соответствия их требованиям. Затем эти метрики предоставляются руководству в формате, позволяющим ему понять текущее состояние безопасности компании и уровень ее соответствия требованиям. Также это должно предоставлять руководству компании дополнительную информацию для принятия бизнес-решений. Безопасность сегодня затрагивает почти все в бизнесе, поэтому эта информация должна быть доступна высшему руководству в форме, которую руководство может реально использовать.

Рисунок 3-20 показывает, как компании обычно развивают зрелость своей безопасности.



Рисунок 3-20. Управление ИТ-безопасностью и Модель зрелости функционирования ИТ

Все эти элементы (стратегическое выравнивание, обеспечение бизнеса, улучшение процессов и эффективность безопасности) могут существовать только при наличии прочного фундамента поддерживающих проектов по безопасности. Проекты намечают актуальные механизмы безопасности, которые будут использоваться для обеспечения необходимого компании уровня защиты. Среди таких проектов может быть управление доступом, управление идентификацией, управление активами, реагирование на инциденты, инфраструктура безопасности, безопасность приложений и т.п. Без прочного фундамента безопасности ни одна из целей реально не может быть достигнута. Это все равно, что пытаться построить дом на песке.

ПРИМЕЧАНИЕ. Фундамент корпоративной безопасности не может быть построен без установленных *зон безопасности*, которые предоставляют основанные на доверии границы, определяющие, что на самом деле должно быть защищено. Это позволяет реализовывать защиту и управление рисками стандартизированным образом в рамках всей компании.

9. Анализ нескольких угроз

Итак, мы поговорили, как все это предположительно должно работать, теперь посмотрим на несколько вещей, в которых можно ошибиться, проектируя систему.

Программное обеспечение практически всегда имеет ошибки и уязвимости. Богатая функциональность, требуемая пользователями, ведет к повышению сложности, что в свою очередь открывает дверь проблемам компьютерного мира. Атакующие постоянно ищут способы использования различных функций систем для проведения атак и очень часто находят уязвимости. Всегда будут полицейские и грабители, всегда будут атакующие и специалисты по безопасности. Это игра, в которой один старается перехитрить другого, чтобы победить.

ПРИМЕЧАНИЕ. Университет Карнеги-Мелона установил, что на каждые 1000 строк кода приходится от 5 до 15 ошибок. Windows 2000 имеет 40-60 миллионов строк кода.

9.1. Закладки для поддержки

В мире программирования *закладки для поддержки* (maintenance hook) являются разновидностью черного хода (backdoor). Разработчик добавляет в программу команды, позволяющие получить простой доступ к коду, которые только он сам знает как вызвать. Это позволяет разработчику просматривать и редактировать код без прохождения через стандартную процедуру управления доступом. На этапе разработки программного обеспечения такие закладки могут быть очень полезны, но если они не были удалены перед передачей системы в промышленную эксплуатацию, они могут стать причиной серьезных проблем с безопасностью.

Закладки для поддержки обычно иницируются «случайной» последовательностью клавиш и позволяют получить доступ в программное обеспечение без прохождения через обычный процесс управления доступом, проверок и механизмов безопасности.

Приложение, которое имеет закладки для поддержки, позволяет разработчику выполнить определенные команды, используя определенные последовательности клавиш. При этом разработчик может работать внутри приложения, напрямую видеть код или конфигурационные файлы. Он может увидеть проблемные места в коде, проверить значения переменных, экспортировать дополнительный код в программу, исправить выявленные недостатки. Хотя это звучит прекрасно и здорово при использовании разработчиком, атакующий, найдя эти закладки, сможет выполнить крайне опасные действия. Поэтому все закладки должны быть удалены из программного обеспечения до начала его промышленной эксплуатации.

ПРИМЕЧАНИЕ. Многие полагают, что раз в настоящее время люди больше думают о безопасности, то и закладки для поддержки остались в прошлом. Но это не так. Разработчики продолжают использовать закладки для поддержки, потому что они недостаточно хорошо понимают и заботятся о вопросах безопасности. Множество закладок по-прежнему остается в старом программном обеспечении, которое используют компании.

Контрмеры:

Так как закладки для поддержки обычно вставляются программистами, они единственные, кто может удалить их до внедрения программы в промышленную эксплуатацию. Анализ кода и модулей, качественное тестирование должны выявить все черные ходы, пропущенные программистом. Поскольку закладки для поддержки расположены в коде приложения или системы, пользователь практически ничего не может сделать, чтобы предотвратить их присутствие. Когда производитель находит в коде своего продукта черный ход, обычно он выпускает патч, снижающий или устраняющий эту уязвимость. Т.к. большинство производителей продает свое программное обеспечение без исходных текстов, купившим его компаниям очень сложно выявить в нем черные ходы. Ниже представлен список некоторых превентивных мер против черных ходов:

- Используйте системы IDS уровня хоста, чтобы выявить факт использования черного хода атакующим;
- Используйте шифрующие файловые системы для защиты критичной информации;
- Внедрите аудит для выявления фактов использования черных ходов.

9.2. Атаки времени проверки / времени использования

Некоторые атаки могут использовать в своих интересах способы обработки системных вызовов и выполнения задач. Атаки *«времени проверки»* / *«времени использования»* (TOC/TOU – time-of-check/time-of-use) связаны с последовательностью шагов, выполняемых системой для завершения задачи. Это тип атак использует в своих целях зависимость от времени событий, происходящих в многозадачной операционной системе.

Как было сказано ранее, операционные системы и приложения в действительности являются просто множеством строк команд. Операционная система может выполнять команду 1, за ней команду 2, потом 3 и т.д. в зависимости от того, как написана программа. Если атакующий сможет вмешаться между командами 2 и 3 и оказать некоторое воздействие, он сможет получить контроль над результатом этой деятельности.

Приведем пример TOC/TOU-атаки. Процесс 1 проверяет авторизацию пользователя для открытия ему некритичного текстового файла, процесс 2 выполняет команду открытия. Если атакующий сможет подменить этот некритичный текстовый файл файлом, содержащим пароли, в то время, пока процесс 1 выполняет свою задачу, он сможет получить доступ к этому файлу. Возможность проведения такой атаки однозначно указывает на недостаток в коде программного обеспечения.

ПРИМЕЧАНИЕ. Этот тип атаки также называют асинхронной атакой. Асинхронным называется такой процесс, в котором время каждого шага может меняться. Атака происходит между этими шагами и изменяет что-то. Иногда TOC/TOU-атаки считают разновидностью атак соревнования (race conditions).

Атака соревнования (race condition) становится возможна, когда два различных процесса должны выполнить свои задачи с использованием одного и того же ресурса. Процессы должны работать в правильной последовательности. Процесс 1 должен выполнить свою работу до того момента, как процесс 2 получит доступ к тому же ресурсу и приступит к выполнению своей работы. Если процесс 2 начнет раньше процесса 1, результат может быть совсем другим. Если атакующий может управлять процессами, так, чтобы заставить процесс 2 начать первым, он может управлять результатом процедуры. Скажем, что командой процесса 1 является прибавление 3 к значению, а командой процесса 2 – деление значения на 15. Если процесс 2 выполнит свою команду первым, результат будет другим. Таким образом, если атакующий может заставить процесс 2 выполниться раньше процесса 1, он может управлять результатом.

Посмотрим на эту проблему с точки зрения безопасности. Существует несколько типов достаточно опасных атак соревнования. Если система разделяет шаги аутентификации и авторизации, атакующий может быть авторизован до прохождения аутентификации. Например, в нормальной последовательности, процесс 1 проводит аутентификацию до того как позволит пользователю получить доступ к ресурсу, а процесс 2 авторизует пользователя для доступа к ресурсу. Если атакующий заставит процесс 2 выполниться раньше процесса 1, он может получить доступ к ресурсу без прохождения процедуры аутентификации.

Хотя термины «атака соревнования» и «TOC/TOU-атака» иногда используют, подразумевая одно и то же, на самом деле это две разных вещи. В атаке соревнования атакующий заставляет процессы выполниться в неправильной последовательности для получения контроля над результатом. В TOC/TOU-атаке атакующий попадает между двух задач и изменяет что-то для контроля над результатом.

Контрмеры

Достаточно сложно обеспечить высокую точность при проведении этих типов атак, но это возможно и реализуется на практике. Для защиты от атак соревнования лучше всего не разделять критичные задачи, которые недопустимо выполнять в иной последовательности. Это означает, что система должна использовать атомарные операции, в рамках которых только один системный вызов должен использоваться и для проверки аутентификации, и для последующего предоставления доступа в рамках одной задачи. У процессора не должно быть возможности переключиться на другой процесс до окончания выполнения этих двух операций. К сожалению, использование соответствующих типов атомарных операций не всегда возможно.

Чтобы избежать ТОС/TOU-атаки лучше всего, если операционная система может применять программные блокировки к элементам, которые будут использоваться при выполнении задач «проверки». Так, если пользователь запрашивает доступ к файлу, система на время проведения его авторизации применяет программную блокировку к запрашиваемому файлу. Это гарантирует, что файл не может быть удален или заменен другим файлом. Применение таких блокировок легко реализовать для файлов, но применять их к компонентам баз данных и содержимому таблиц может быть значительно более сложной задачей.

9.3. Переполнение буфера

В настоящее время многие знакомы с термином «переполнение буфера» и его определением, но для специалистов по безопасности важно понимать его более глубоко.

Переполнение буфера (buffer overflow) происходит, когда приложением или операционной системой принимается слишком много входящих данных. Буфер – это некий выделенный сегмент памяти. Буфер может быть переполнен слишком большим объемом данных. Чтобы этим мог воспользоваться атакующий и запустить свой код на выполнение, вставляемый в буфер код должен иметь строго определенную длину. Таким образом, целью переполнения буфера может быть нарушение работы приложения путем записи произвольных данных в различные сегменты памяти; либо выполнение определенной задачи, путем введения в определенный сегмент памяти специально подготовленного набора данных, который выполнит эту задачу. Задача может заключаться, например, в открытии командной оболочки с административными полномочиями или выполнении вредоносного кода.

Давайте поглубже рассмотрим как это делается. Программное обеспечение может быть написано для получения данных от пользователя, веб-сайта, базы данных или другого приложения. Принятые данные требуют выполнения какого-либо действия, они принимаются для некоего типа манипуляций или расчетов, либо использования в качестве параметра процедуры. Процедура – это часть кода, которая может выполнять определенные действия над данными и возвращать результат вызвавшей ее программе, как это показано на Рисунке 3-21.

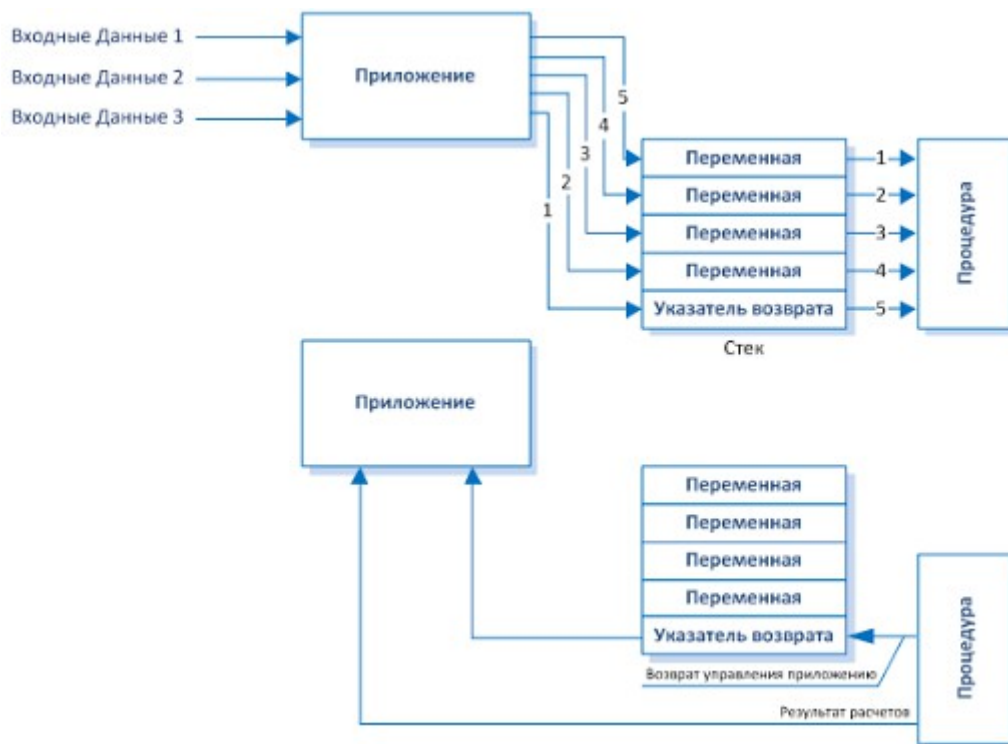


Рисунок 3-21. Стек использует отдельный буфер для хранения команд и данных

Когда программист пишет фрагмент кода, принимающий данные, он организует хранение этих данных в переменной. При вызове программой некой процедуры для выполнения определенной функции, она сохраняет необходимые инструкции и данные (параметры) в сегменте памяти, чтобы процедура смогла прочесть их и использовать в своей работе. (Стек памяти рассматривался ранее в этом Домене, но мы снова вернемся к нему в этом разделе).

Принятые от внешнего источника данные, размещаются в переменной. Эта переменная расположена в области памяти, которая называется буфером. Буфер похож на контейнер памяти для данных. Буфер должен быть достаточного размера, чтобы вместить все принятые данные. Так, если ожидается ввод одного символа, буфер должен иметь размер в 1 байт. Если программист не убедился, что получен только 1 байт, кто-то может ввести несколько символов и переполнить этот буфер.

Буфер хранит данные, размещающиеся в стеке памяти. Представьте, что буфер – это маленькое ведро с водой (данными). У нас есть несколько таких ведер, расположенных один над другим (стек). Если слишком много воды налить в верхнее ведро, вода перельется в стоящие ниже ведра (переполнение буфера) и перезапишет инструкции и данные в стеке памяти.

Что такое стек и как он работает?

Если вы работаете с приложением, которое считает процентную ставку по ипотеке, вы вводите в качестве параметров для расчета срок кредита и сумму кредита. Эти параметры будут сохранены в пустых переменных и размещены в некой линейной конструкции (стеке памяти), которая действует как очередь операций извлечения данных для выполнения расчета. Первая задача вашего приложения расчета ипотечной ставки – разместить указатель возврата (return pointer). Это указатель на адрес памяти запрашивающего приложения, который говорит процедуре, как вернуть управление запрашивающему приложению после обработки ей всех значений в стеке. Затем приложение переходит к адресу выше указателя возврата, и размещает там оставшиеся данные (введенные пользователем). После этого приложение отправляет запрос процедуре для выполнения необходимых вычислений, как показано на Рисунке 3-21. Процедура берет данные из стека, начиная сверху (первым пришел

– последним ушел, FILO – first in, last off). Процедура выполняет эти функции над всеми данными и возвращает результат и управление обратно приложению, используя указатель возврата в стеке.

Таким образом, стек – это просто сегмент памяти, который позволяет обмениваться информацией между запрашивающим приложением и подпрограммой (процедурой). Потенциальные проблемы появляются в случае, если запрашивающее приложение не выполняет надлежащую проверку границ, т.е. не убеждается, что введенные данные имеют приемлемую длину. Посмотрите на следующий код на Си, чтобы понять, как это происходит:

```
#include
int main(int argc, char **argv)
{
    char buf1 [5] = "1111";
    char buf2 [7] = "222222";
    strcpy (buf2, "3333333333");
    printf ("%s\n", buf2);
    printf ("%s\n", buf1);
    return 0;
}
```

ПРЕДУПРЕЖДЕНИЕ. Вам не обязательно знать язык Си, чтобы сдать экзамен CISSP. Мы углубились в этот вопрос потому, что переполнение буфера является общей и серьезной уязвимостью уже много лет. Для сдачи экзамена вам достаточно понимать общую концепцию переполнения буфера.

Итак, сначала мы установили длину буфера buf1 равной четырем символам, а буфера buf2 – шести символам. Для обоих буферов установлено начальное значение NULL (значение NULL указывает на конец буфера в памяти). Если мы сейчас посмотрим содержимое этих буферов, мы увидим следующее:

```
Buf2
\0 2 2 2 2 2
Buf1
\0 1 1 1 1
```

Затем наше приложение записывает десять троек в buf2, который может хранить только шесть символов. При этом шесть символов будет записано в buf2, а еще четыре символа – в buf1, затирая расположенные в нем ранее данные. Это происходит потому, что команда strcpy не проверяет, имеет ли буфер достаточную длину для хранения нужного количества символов. Так, теперь мы увидим следующее содержимое буферов:

```
Buf2
\0 3 3 3 3 3
Buf1
\0 3 3 3 3
```

Но результат будет еще интереснее, если будет перезаписан указатель возврата, как это показано на Рисунке 3-22. Тщательно подготовленная атака переполнения буфера обеспечивает перезапись указателя возврата на контролируемое значение с целью выполнения загруженных в стек вредоносных команд, вместо возврата управления запросившему приложению. Это позволяет выполнить вредоносные команды в контексте безопасности запросившего приложения. Если приложение работает в привилегированном режиме, атакующий получает более обширный доступ и привилегии для нанесения большего ущерба.

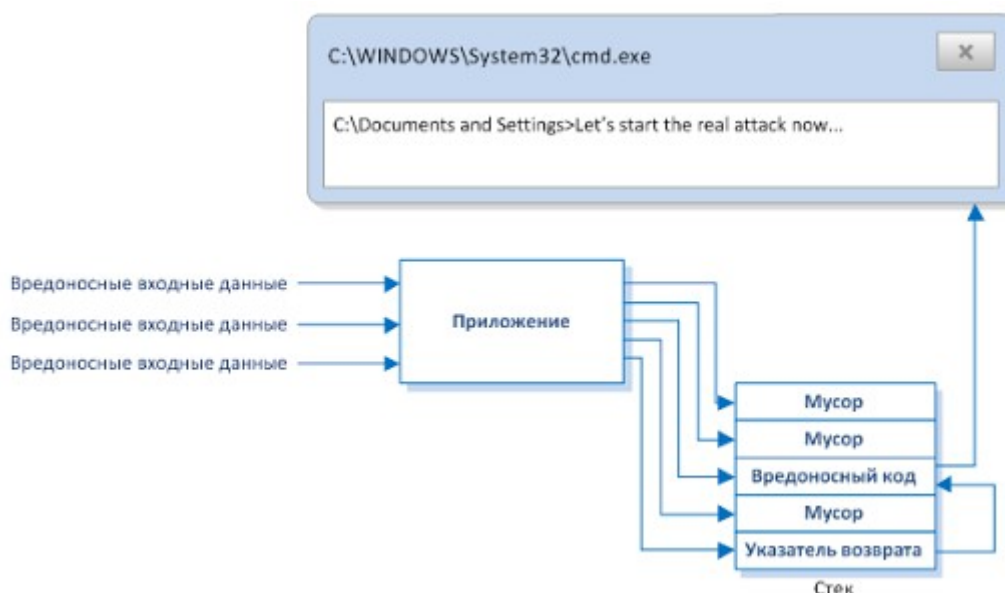


Рисунок 3-22. Атака переполнения буфера

Атакующий должен знать размер буфера, чтобы его переполнить, а также он должен знать связанные со стеком адреса. Без знания этих адресов он не сможет установить новый указатель возврата на свой вредоносный код. Атакующий должен сделать свое вредоносное вложение достаточно маленьким по размеру, чтобы его можно было поместить в данные, передаваемые от одной процедуры к другой.

Ядро Windows написано на языке Си и имеет несколько слоев объектно-ориентированного кода над собой. Когда процедуре нужно вызвать операционную систему для выполнения некой задачи, она обращается к системной службе посредством API-вызова. API работает как дверь к функциям операционной системы.

Язык Си чувствителен к атакам переполнения буфера, т.к. он позволяет использовать прямые операции с указателями. Отдельные команды могут предоставить доступ к низкоуровневым адресам памяти без проверки границ. Функции Си, которые выполняют необходимую проверку границ, это `sprintf()`, `strcat()`, `strcpy()` и `vsprintf()`.

Операционная система должна быть написана для работы с конкретными архитектурами процессоров. Эти архитектуры диктуют адресацию системной памяти, защитные механизмы, режимы выполнения, работу с определенными наборами команд. Это означает, что атака переполнения буфера, работающая на процессорах Intel, не обязательно будет работать на процессорах Motorola или SPARC. Эти различные процессоры используют различную адресацию памяти стека, поэтому атакующий должен иметь несколько вариантов кода переполнения буфера для различных платформ. Обычно это не является проблемой, т.к. в последнее время такой код уже написан и доступен на хакерских веб-сайтах.

Контрмеры

Недостатки, позволяющие провести переполнение буфера, содержатся в исходном коде различных приложений и операционных систем. Они были с тех пор, как программисты начали разрабатывать программное обеспечение. Пользователю очень сложно выявить и исправить их. Когда выявляется возможность переполнения буфера, производитель обычно рассылает патч. Таким образом, установка на систему всех актуальных исправлений, обновлений, патчей обычно является лучшей контрмерой. Существуют некоторые продукты, которые устанавливаются на систему и следят за входными значениями, выявляя попытки переполнения буфера. Однако идеальной контрмерой является качественное программирование, что означает надлежащее выполнение проверки границ. Если вводимое

значение должно иметь длину девять символов, приложение должно принимать только девять символов и не более. Некоторые языки программирования более чувствительны к переполнению буфера, чем другие. Программисты должны понимать это и выбирать правильные языки в соответствии с поставленными целями, выполнять анализ кода для выявления уязвимостей переполнения буфера.

10. Резюме

Архитектура компьютерных систем очень важна, она включает в себя множество различных аспектов.

Система должна обеспечить, чтобы память правильно разделялась и защищалась, чтобы только уполномоченные субъекты могли использовать объекты, чтобы недоверенные процессы не могли выполнять действий, которые могут подвергнуть риску другие процессы. Также, система должна обеспечить управление потоками информации и определить домены ресурсов для каждого субъекта. Если на компьютере произошел какой-либо сбой, система не должна перейти из-за этого в небезопасное состояние. Многие из этих вопросов учитываются в политике безопасности системы и в модели безопасности, созданной для поддержки требований этой политики.

После разработки политики безопасности, модели и архитектуры, операционная система (или другой продукт) должна быть разработана, протестирована, затем она должна пройти процедуру оценки и ей должен быть присвоен соответствующий рейтинг. Оценка реализуется путем сравнения системы с заранее определенными критериями. Рейтинг присваивается системе в зависимости от того, как она выполняет требования этих критериев. Покупатели используют этот рейтинг, чтобы понять, что они реально покупают, и насколько они могут доверять этому новому продукту. После покупки продукта, покупатель должен протестировать его в своей среде, чтобы быть уверенным, что он соответствует потребностям компании. Это выполняется в рамках процессов сертификации и аккредитации.

Тест

Вопросы экзамена CISSP являются концептуальными, поэтому они сформулированы соответствующим образом. Задачей кандидата является выбор наилучшего из всех представленных вариантов ответа. Среди вариантов ответа может не быть идеального ответа на поставленный вопрос - кандидат должен выбрать лучший ответ из имеющихся вариантов.

1. Какой недостаток является причиной уязвимости переполнения буфера?

- ☐ A. Приложение запускается в защищенном режиме
- ☐ B. Неправильная сегментация памяти
- ☐ C. Неправильное использование колец защиты
- ☐ D. Ненадлежащая проверка границ

2. Операционная система выполняет все, за исключением какой из перечисленных ниже задач?

- ☐ A. Распределение памяти
- ☐ B. Задачи ввода/вывода
- ☐ C. Распределение ресурсов
- ☐ D. Доступ пользователей к представлениям базы данных

3. Если операционная система позволяет следующему субъекту использовать некий объект без его предварительной очистки, какую проблему безопасности это вызывает?

- ☐ A. Раскрытие остаточных данных
- ☐ B. Несанкционированный доступ к привилегированным процессам
- ☐ C. Утечка данных по скрытым каналам
- ☐ D. Компрометация домена выполнения

4. Что является заключительным шагом разрешения использования системы в конкретной среде?

- ☐ A. Сертификация
- ☐ B. Оценка безопасности и присвоение рейтинга
- ☐ C. Аккредитация
- ☐ D. Проверка

5. Какая возможность позволяет выполнить некоторый код без прохождения обычной проверки безопасности?

- ☐ А. Временная изоляция
- ☐ В. Закладки для поддержки
- ☐ С. Атака соревнования (race conditions)
- ☐ D. Мультиплексирование процессов

6. Если в компоненте происходит сбой, что из приведенного ниже должна сделать система?

- ☐ А. Перейти в защищенный домен выполнения
- ☐ В. Перейти в проблемное состояние
- ☐ С. Перейти в более защищенное состояние
- ☐ D. Удалить все данные в энергозависимой памяти

7. Какое преимущество с точки зрения безопасности имеют прошивки над обычным программным обеспечением?

- ☐ А. Их трудно изменить без физического доступа
- ☐ В. Они требуют меньше памяти
- ☐ С. Они не нужны для реализации политики безопасности
- ☐ D. Их проще перепрограммировать

8. Какой уровень в Оранжевой книге является первым, в котором требуется установка меток классификации данных?

- ☐ А. B3
- ☐ В. B2
- ☐ С. B1
- ☐ D. C2

9. Какой из следующих пунктов лучше всего описывает ядро безопасности?

- ☐ А. Программный компонент, который отслеживает действия и записывает информацию о событиях безопасности в журнал регистрации событий
- ☐ В. Программный компонент, который определяет, имеет ли пользователь право выполнять запрошенную операцию
- ☐ С. Программный компонент, который изолирует процессы, разделяет защищенный и реальный режимы
- ☐ D. Программный компонент, который работает в самом центре колец защиты и обеспечивает интерфейсы между доверенными и недоверенными объектами

10. Для чего из перечисленного ниже был разработан ITSEC?

- ☐ А. Международного использования
- ☐ В. Использования в США
- ☐ С. Использования в Европе
- ☐ D. Использования в глобальных масштабах

11. Что из перечисленного ниже входит в состав ядра безопасности?

- ☐ А. Программное обеспечение, аппаратное обеспечение и прошивки
- ☐ В. Программное обеспечение, аппаратное обеспечение и структура системы
- ☐ С. Политика безопасности, механизмы защиты и программное обеспечение
- ☐ D. Политика безопасности, защитные механизмы и структура системы

12. Что является целью использования базового регистра и регистра границы области памяти?

- ☐ А. Контрмера против переполнения буфера
- ☐ В. Разделение времени использования ресурсов, в основном - процессора
- ☐ С. Изоляция процессов
- ☐ D. Реализация TCB

13. Охранники (guard) обычно используются с классифицированными системами. Что является основной целью внедрения и использования охранников?

- ☐ А. Обеспечить, чтобы менее доверенные системы могли получать только подтверждения, но не сообщения
- ☐ В. Обеспечить правильные потоки информации
- ☐ С. Обеспечить, чтобы менее доверенные и более доверенные системы имели открытую архитектуру и совместимость
- ☐ D. Позволить взаимодействовать системам с многоуровневой безопасностью и системам, работающим в специальном (dedicated) режиме

14. Чем из перечисленного ниже управляет TCB?

- ☐ А. Всеми доверенными процессами и программными компонентами
- ☐ В. Всеми доверенными политиками безопасности и реализацией механизмов
- ☐ С. Всем доверенным программным обеспечением и структурой механизмов
- ☐ D. Всем доверенным программным и аппаратным обеспечением

15. Что является воображаемыми границами, отделяющими поддерживающие безопасность компоненты от компонентов, не связанных с безопасностью?

- ☐ A. Монитор обращений
- ☐ B. Ядро безопасности
- ☐ C. Периметр безопасности
- ☐ D. Политика безопасности

16. Какая модель учитывает только конфиденциальность?

- ☐ A. Bell-LaPadula
- ☐ B. Clark-Wilson
- ☐ C. Biba
- ☐ D. Монитор обращений

17. Что из перечисленного ниже является лучшим описанием ядра безопасности с точки зрения безопасности?

- ☐ A. Монитор обращений
- ☐ B. Менеджер ресурсов
- ☐ C. Отображение памяти
- ☐ D. Периметр безопасности

18. При каких условиях безопасность системы является самой эффективной и экономичной?

- ☐ A. Если она спроектирована и реализована с самого начала разработки системы
- ☐ B. Если она спроектирована и реализована как безопасный и доверенный внешний интерфейс
- ☐ C. Если она специально спроектирована для противодействия определенным видам атак
- ☐ D. Если система была оптимизирована перед добавлением в нее безопасности

19. В безопасных компьютерных системах, зачем нужны логические формы разделения между процессами?

- ☐ A. Процессы заключены в свои собственные домены безопасности, поэтому ни один из них не делает несанкционированных запросов к другим процессам или их ресурсам
- ☐ B. Процессы заключены в свой собственный периметр безопасности, поэтому они могут обращаться только к вышестоящему уровню защиты
- ☐ C. Процессы заключены в свой собственный периметр безопасности, поэтому они могут обращаться только к эквивалентному им уровню защиты
- ☐ D. Разделение осуществляется на аппаратном уровне и не является логическим по своей природе

20. Какой вид атаки происходит, когда субъект на высоком уровне безопасности записывает данные в определенную область хранения информации, которую читает субъект на низком уровне безопасности?

- ☐ A. ТОС/TOU
- ☐ B. Атака посредством скрытых каналов по памяти
- ☐ C. Атака посредством скрытых каналов по времени
- ☐ D. Переполнение буфера

21. Какой вид рейтинга присваивают продуктам Общие критерии?

- ☐ A. PP
- ☐ B. EPL
- ☐ C. EAL
- ☐ D. A-D

22. Что лучше всего описывает аксиому *-целостности?

- ☐ A. Нельзя записывать вверх в модели Biba
- ☐ B. Нельзя читать снизу в модели Biba
- ☐ C. Нельзя записывать вниз в модели Bell-LaPadula
- ☐ D. Нельзя читать сверху в модели Bell-LaPadula

23. Что лучше всего описывает простое правило безопасности?

- ☐ A. Нельзя записывать вверх в модели Biba
- ☐ B. Нельзя читать снизу в модели Biba
- ☐ C. Нельзя записывать вниз в модели Bell-LaPadula
- ☐ D. Нельзя читать сверху в модели Bell-LaPadula

24. Что из перечисленного ниже было первой математической моделью многоуровневой политики безопасности, использовавшейся для определения концепций состояния безопасности и режимов доступа, описания правил доступа?

- ☐ A. Biba
- ☐ B. Bell-LaPadula
- ☐ C. Clark-Wilson
- ☐ D. Конечные автоматы

25. Что из перечисленного ниже является верным утверждением по отношению к адресации памяти?

- ☐ А. Процессор использует абсолютные адреса. Приложения используют логические адреса. Относительная адресация основана на известном адресе и величине смещения.
- ☐ В. Процессор использует логические адреса. Приложения используют абсолютные адреса. Относительная адресация основана на известном адресе и величине смещения.
- ☐ С. Процессор использует абсолютные адреса. Приложения используют относительные адреса. Логическая адресация основана на известном адресе и величине смещения.
- ☐ D. Процессор использует абсолютные адреса. Приложения используют логические адреса. Абсолютная адресация основана на известном адресе и величине смещения.

Домен 04. Физическая безопасность и безопасность окружения.

Безопасность очень важна для организаций и их инфраструктуры и физическая безопасность – не исключение. Хакинг – не единственный способ компрометации информации и связанных с ней систем. Физическая безопасность учитывает угрозы, уязвимости и риски, отличающиеся от тех, которые мы уже рассматривали ранее. Механизмы физической безопасности включают в себя планирование помещений, компонентов окружения, планирование действий на случай чрезвычайных происшествий, проведение тренингов по выполнению таких планов, контроль доступа, выявление вторжений, защиту систем электроснабжения, противопожарную безопасность. Механизмы физической безопасности защищают людей, данные, оборудование, системы, сооружения и многие другие активы компании.

1. Введение в Физическую безопасность

Физическая безопасность компьютеров и их ресурсов в 60-х и 70-х годах была не так сложна, как сегодня. В те времена использовались в основном мейнфреймы, которые можно было запереть в серверной комнате и только горстка людей знала что с ними делать. Сегодня компьютер есть на каждом столе каждой компании, а доступ к устройствам и ресурсам распространился на всю среду. Компании имеют несколько кроссовых и серверных комнат, удаленных и мобильных пользователей, выносящих мобильные компьютеры и данные за пределы здания компании. Надежная защита этих компьютерных систем, сетей, зданий и сотрудников стала крайне сложной задачей для многих компаний.

Во многих компаниях растет ущерб от воровства, мошенничества, саботажа, вандализма, несчастных случаев. Это связано с тем, что их среда становится более сложной и динамичной. Безопасность и сложность – это труднсовместимые вещи. По мере роста сложности среды и технологий, появляется больше уязвимостей, которые могут привести к компрометации. Многие компании сталкиваются с хищениями модулей памяти и процессоров из рабочих станций, а также хищениями компьютеров и ноутбуков. Многие компании становятся жертвами более опасных преступлений – вооруженные ограбления, вспышки ярости раздраженных сотрудников, бомбы, террористическая деятельность. Многие компании имеют охрану, систему видеонаблюдения, системы выявления вторжений, поддерживают осведомленность сотрудников о рисках безопасности. Это лишь некоторые элементы, относящиеся к физической безопасности. Если какой-то из них не предоставляет необходимого уровня защиты, он может стать слабым звеном и потенциальной брешью в системе безопасности.

Большинство людей, занимающихся информационной безопасностью, недостаточно задумываются о физической безопасности, они сосредоточены на *компьютерной* безопасности, хакерах, портах, вирусах и технологических контрмерах. Но информационная безопасность без надежной физической безопасности – это пустая трата времени.

Даже специалисты по физической безопасности не всегда не всегда имеют целостный взгляд на физическую безопасность. В физической безопасности так много различных аспектов, которые необходимо учитывать, что люди обычно специализируются на отдельных областях, таких как безопасность зданий, выявление и анализ рисков, обеспечение защиты центров обработки данных (ЦОД), противопожарная ,безопасность, внедрение систем обнаружения вторжений и систем видеонаблюдения, обучение персонала по действиям в чрезвычайных ситуациях, аспекты требований законодательства и регуляторов в области физической безопасности и т.п. Каждый фокусируется на своей области и имеет свои навыки, но для обеспечения целостной системы безопасности компании все эти области должны быть поняты и учтены.

Как и большинство программных продуктов, построенных, в первую очередь, с учетом функциональности, а не безопасности, так и многие здания и сооружения построены с

учетом функциональности, эстетизма, а уже затем – безопасности. Многие кражи и несчастные случаи могли бы быть предотвращены, если бы компания реализовала организованную, зрелую и целостную систему физической безопасности. Большинство людей не знают о множестве преступлений, случающихся каждый день. Многие не догадываются о большом количестве гражданских исков к компаниям, не уделяющим должного внимания физической безопасности. Ниже представлено несколько примеров проблем, которые могут возникнуть у компаний, плохо реализовавших или поддерживающих свою физическую безопасность:

- Кусты, растущие очень близко от банкомата, позволяют преступникам скрыться за ними и нападать на людей, снявших деньги со своих банковских карт.
- Неосвещенная часть подземного гаража позволяет злоумышленникам сидеть и ждать сотрудников, которые задержались на работе допоздна.
- Ночной магазин развесил очень много рекламных плакатов и постеров на наружных окнах, что привлекает к нему грабителей, т.к. развешенные плакаты скрывают происходящее внутри магазина от прохожих и проезжающих машин.

Для специалистов по безопасности очень важно производить оценку, основываясь на точке зрения потенциального преступника. Это позволит выявить и устранить наиболее критичные уязвимые места, которые могут быть использованы преступниками. Специалисты по безопасности должны рассматривать безопасность как целостный процесс, оценивать его со всех сторон, использовать различные подходы. Опасность может прийти отовсюду, она может принимать любые формы и иметь различные последствия.

Физическая безопасность работает с набором угроз, уязвимостей и контрмер, отличающимся от компьютерной и информационной безопасности. Этот набор для физической безопасности в большей степени направлен на физическое уничтожение, на нарушителей, на объекты среды, на воровство и вандализм. Когда специалисты по безопасности думают об *информационной* безопасности, они представляют себе, как кто-то несанкционированно проникнет во внутреннюю сеть через открытый на межсетевом экране порт или беспроводную точку доступа. Думая о *физической* безопасности, специалисты по безопасности представляют себе, как злоумышленник входит в здание и наносит ущерб.

Угрозы, перед лицом которых стоит компания, можно разделить на следующие категории:

- **Природные угрозы.** Затопление, землетрясение, шторм, торнадо, пожар, экстремальные температуры и т.д.
- **Угрозы в отношении систем обеспечения.** Перебои электроснабжения, связи, перебои со снабжением водой, газом и т.п.
- **Рукотворные угрозы.** Несанкционированный доступ (внешний и внутренний), ущерб от преднамеренных действий персонала, ошибки сотрудников, халатность, вандализм, мошенничество, воровство и т.д.
- **Политические угрозы.** Нападения, беспорядки, гражданское неповиновение, террористические атаки, бомбы и т.д.

В любой ситуации нужно в первую очередь продумать, что мешает достижению целей безопасности. Когда мы проектируем безопасность, мы в первую очередь должны думать о том, как **защитить человеческую жизнь**. Хорошее планирование поможет нам сбалансировать безопасность жизни людей с другими мерами безопасности. Например, закрытие дверей аварийных выходов с целью предотвращения несанкционированного доступа может помешать эвакуировать людей в случае пожара.

ПРИМЕЧАНИЕ. Цели защиты жизни должны всегда превосходить любые другие цели.

Программа физической безопасности должна объединять в себе механизмы обеспечения

защиты и безопасности. Под защитой (safety) подразумевается защита жизни людей, а также защита активов компании от огня, природных катастроф, разрушительных инцидентов. Под безопасностью (security) подразумевается защита от вандализма, краж, атак злоумышленников. Часто возникают пересечения между различными типами угроз, поэтому их нужно понимать и надлежащим образом планировать. Этот Домен рассматривает как механизмы защиты, так и механизмы безопасности, о которых должен знать каждый специалист по безопасности.

Физическую безопасность следует внедрять на основе *многоуровневой модели защиты* (layered defense model), в которой защитные меры должны работать совместно. Эта концепция предусматривает, что если защита на одном уровне нарушена, другие уровни продолжают защищать ценный актив. Уровни защиты должны быть реализованы на пути от периметра до актива. Например, у вас есть изгородь вокруг здания, затем стены самого здания, система контроля доступа на смарт-картах при входе в здание, система выявления вторжений, заперты компьютерные корпуса и сейфы. Такая последовательность уровней защиты предназначена для защиты критичных активов компании, размещенных в самой глубине контролируемой зоны. Так, если плохие парни преодолеют изгородь и перехитрят охранников на входе, им нужно будет преодолеть еще несколько уровней защиты, прежде чем они получат доступ к защищаемым ресурсам и системам.

Безопасность должна обеспечивать защиту всех активов компании и увеличивать ее продуктивность, обеспечивая безопасную и предсказуемую среду. Хорошая безопасность позволяет сотрудникам сосредоточиться на своих задачах, а злоумышленников вынуждает переключаться на другие, более уязвимые и легкие цели. Это в идеале, конечно. Вспоминая АИС-триаду безопасности, описанную в Домене 01, мы можем взглянуть на физическую безопасность, как на защиту доступности ресурсов компании, целостности активов и среды, конфиденциальности данных и бизнес-процессов.

2. Процесс планирования

Должна быть определена группа разработчиков (или отдельный разработчик) для создания или совершенствования программы физической безопасности компании. Эта группа должна работать с руководством, чтобы определить цели программы, разработать программу, систему показателей ее эффективности, а также процессы оценки этих показателей, позволяющие убедиться, что все цели достигнуты.

Цели программы физической безопасности зависят от требуемого уровня защиты для различных активов и компании в целом. А этот требуемый уровень защиты, в свою очередь, зависит от уровня приемлемых для компании рисков. При определении приемлемого уровня рисков, следует основываться на законодательстве и требованиях регуляторов, которым компания должна соответствовать, а также на основании профиля угроз для компании в целом. Это требует определить, кто (или что) может нанести ущерб бизнес-активам, определить типы атак и преступлений, которые могут произойти, понять степень воздействия на бизнес этих угроз. Тип требуемых физических контрмер, их адекватность (или неадекватность) могут быть оценены на основании профиля угроз компании. Профили угроз финансовых компаний сильно отличаются друг от друга и, соответственно, сильно отличается приемлемый уровень рисков для них. Профиль угроз для продовольственных магазинов довольно прост. А профиль угроз больницы отличается от профиля угроз военной базы. Группа, разрабатывающая программу физической безопасности, должна понимать, какие типы злоумышленников должны быть учтены, каковы их возможности, их ресурсы и тактика. (Посмотрите еще раз Домен 01, где обсуждается концепция приемлемого уровня риска).

Физическая безопасность – это комбинация людей, процессов, процедур и оборудования для защиты ресурсов. Разработка полноценной программы физической безопасности должна быть систематичной и взвешенной с точки зрения целей программы и доступных ресурсов.

Хотя каждая компания имеет свои отличия, подход к построению и поддержке программы физической безопасности у всех компаний одинаков. Сначала нужно определить уязвимости, угрозы, источники угроз и их цели.

ПРИМЕЧАНИЕ. Помните, что уязвимость – это слабость, а угроза – это потенциал того, что кто-то попытается выявить эту слабость и использовать ее против вас. Источник угрозы – это человек или механизм, который сможет воспользоваться выявленной уязвимостью.

Угрозы можно разделить на внутренние и внешние. Внутренние угрозы могут включать неисправность устройств, пожар, действия внутренних сотрудников, направленные на нанесение вреда компании. Внутренние сотрудники имеют более глубокие знания о возможностях и активах компании, которые они используют для выполнения своих повседневных задач. С одной стороны это необходимо им для работы, но в то же время это облегчает им возможные действия, направленные против компании, затрудняя при этом их выявление. К сожалению, серьезной угрозой для компаний могут являться их собственные охранники, которые бездействуют, пока уже не будет слишком поздно. Кроме того, эти люди имеют ключи и коды доступа во все помещения и обычно работают в нерабочее время. Это дает охранникам обширные возможности для совершения преступлений. Компании нужно принять очень важное решение – проводить при приеме на работу охранников собственные проверки кандидатов, либо заплатить за это специализированной компании.

Внешние угрозы могут иметь множество различных форм. Например, государственные учреждения иногда выбирают компании для публичного наказания. Если ваша компания занимается нелегальной или полунелегальной деятельностью, вам стоит опасаться такого развития событий. Ну и, конечно же, банки и бронированные автомобили постоянно привлекают организованную преступность.

Еще сложнее защититься от угроз, связанных с возможным *сговором* двух или более лиц, выполняющих мошенническую деятельность совместно. Многие преступления совершаются скрытыми инсайдерами, работающими совместно с внешними лицами для нанесения вреда компании. Для защиты от этого используются процедурные защитные механизмы, описанные в Домене 01. Они могут включать в себя разделение обязанностей, проверку кандидатов на работу, ротацию обязанностей и надзор.

Также как в любом другом типе безопасности, большой резонанс получают слухи и кричащие заголовки вокруг крупных преступлений. В информационной безопасности люди часто сосредоточены на вирусах и хакерах, а вовсе не на компонентах, реализующих программу безопасности компании. То же самое справедливо и для физической безопасности. Многие люди говорят о грабежах, убийствах и других криминальных действиях, но не уделяют внимания основам, которые должны быть реализованы и поддерживаться для снижения вероятности таких действий. Программа физической безопасности компании должна учитывать следующие цели:

- **Предотвращение преступлений и разрушений посредством сдерживания (устрашения).** Ограждения, посты охраны, сигнализация и т.д.
- **Уменьшение повреждений посредством использования задерживающих механизмов.** Уровни защиты, задерживающие злоумышленника, например, замки, персонал безопасности, барьеры.
- **Выявление вторжений или повреждений.** Дымовые датчики, детекторы движения, видеонаблюдение и т.д.
- **Оценка инцидентов.** Реакция охраны на выявленные инциденты и определение уровня ущерба.
- **Процедуры реагирования.** Механизмы пожаротушения, порядки действий в случае чрезвычайных ситуаций, уведомления о требованиях законодательства, консультации

с внешними специалистами по безопасности.

Таким образом, компании следует пытаться предотвращать возможные преступления и ущерб, однако при этом ей нужно иметь план действий на тот случай, если это все-таки произойдет. Преступник должен быть задержан до получения им доступа к ресурсам, даже если он смог преодолеть несколько уровней защиты. Должна существовать возможность выявления всех типов преступлений и разрушений с помощью компонентов, реализующих программу безопасности. В случае выявления вторжения, должна быть вызвана охрана для оценки ситуации. Охранники должны знать, как нужно правильно реагировать на широкий спектр потенциально опасных действий. Реакция на чрезвычайные ситуации должна осуществляться внутренней группой безопасности или внешними экспертами.

Все это может звучать достаточно просто пока группа, ответственная за разработку программы физической безопасности компании, не приступит к рассмотрению возможных угроз, в условиях ограниченного бюджета, в рамках которого она будет работать, а также сложного выбора правильной комбинации контрмер и обеспечения их согласованной совместной работы, гарантирующей отсутствие брешей в защите. Все эти вещи должны быть абсолютно ясны до начала создания программы безопасности.

Как и для любой программы безопасности, вы можете определить, насколько она выгодна и эффективна, только если вы отслеживаете ее выполнение, используя подход, основанный на показателях и метриках эффективности ваших контрмер. Это даст возможность руководству сопоставлять инвестиции в физическую безопасность компании с ее бизнес-целями. Это направлено на повышение эффективности программы физической безопасности и снижение рисков компании наиболее выгодным способом. Вам следует определить первоначальную эффективность (базис), а затем проводить оценку текущего уровня эффективности на постоянной основе, чтобы быть уверенным, что цели защиты компании достигаются. Ниже приведены несколько примеров возможных показателей эффективности:

- Число успешных преступлений
- Число успешных фактов нанесения ущерба
- Число неудачных преступлений и фактов нанесения ущерба
- Соотношение времени на выполнение этапов выявления, оценки и восстановления
- Влияние инцидентов на бизнес
- Количество ложных срабатываний
- Время, необходимое злоумышленнику, чтобы преодолеть защиту
- Время, необходимое для восстановления функционирования среды

Мониторинг этих показателей эффективности позволяет компании выявить недостатки, оценить улучшение защитных мер, провести анализ затрат/выгод. Группа физической безопасности должна провести анализ рисков, в процессе которого должны быть выявлены уязвимости компании, угрозы и их влияние на бизнес. Группа должна предоставить собранную информацию руководству для определения приемлемого уровня риска, знание величины которого необходимо для разработки программы физической безопасности. Исходя из этого, группа должна разработать описание базиса (минимального уровня безопасности) и метрики для определения влияния и оценки внедряемых контрмер. По мере того, как группа определяет и внедряет контрмеры, их эффективность следует постоянно оценивать посредством заранее определенных метрик. Текущие значения следует сравнивать с базисом. Если базисный уровень безопасности постоянно поддерживается (уровень безопасности не опускается ниже базисного), программа безопасности успешна, так как приемлемый уровень риска не превышает. Это показано на Рисунке 4-1.



Рисунок 4-1. Взаимосвязь рисков, базисов и контрмер

Таким образом, перед внедрением программы безопасности, необходимо выполнить следующие шаги:

- Определить группу внутренних сотрудников и/или внешних консультантов, которые будут создавать программу физической безопасности посредством приведенных ниже шагов.
- Провести анализ рисков для выявления уязвимостей и угроз, рассчитать влияние на бизнес каждой угрозы.
- Работать с руководством для определения приемлемого уровня риска для программы физической безопасности.
- Определить требуемый базисный уровень эффективности защитных мер, исходя из приемлемого уровня риска.
- Создать показатели эффективности контрмер.
- По результатам анализа разработать критерии, описывающие уровень защиты и эффективности, требуемые для следующих категорий программы безопасности:
 - Сдерживание (устрашение)
 - Задерживание
 - Выявление
 - Оценка
 - Реакция
- Определить и внедрить контрмеры для каждой категории программы.
- Постоянно сопоставлять оценку текущего уровня эффективности контрмер с базисным уровнем для уверенности в том, что приемлемый уровень риска не превышает.

Когда эти шаги реализованы (или продолжают выполняться, как в случае последнего из вышеуказанных шагов), группа готова продвинуться дальше в фазе реального проектирования. Проект будет объединять защитные меры, которые требуются для каждой категории программы: сдерживание (устрашение), задерживание, оценка и реакция. Мы глубже рассмотрим эти категории и соответствующие им защитные меры позже в этом Домене, в разделе «Проектирование программы физической безопасности».

Один из подходов, наиболее часто используемый при разработке программы физической безопасности, описан в следующем разделе.

Схожесть в подходах. Шаги анализа рисков очень похожи на шаги разработки программы

безопасности компании, описанные в Домене 01, а также шаги процесса анализа воздействия на бизнес, описанные в Домене 07, так как каждый из этих процессов (разработка программы информационной безопасности, физической безопасности или плана обеспечения непрерывности бизнеса) направлен на цели, похожие на цели двух других процессов, отличаясь только точкой фокуса. Каждый процесс требует создания группы для определения угроз компании, выполнения анализа рисков. Программа информационной безопасности смотрит на внутренние и внешние угрозы ресурсам и данным через бизнес-процессы и технологии. Непрерывность бизнеса анализирует, как могут повредить компании природные катастрофы и разрушения. А физическая безопасность рассматривает внешние и внутренние физические угрозы ресурсам компании. Каждый из этих процессов требует полноценного анализа рисков. Посмотрите еще раз Домен 01, чтобы вспомнить ключевые аспекты любого процесса анализа рисков.

2.1. Предотвращение преступлений посредством проектирования окружения

Предотвращение преступлений посредством проектирования окружения (CPTED – Crime Prevention Through Environmental Design) – это дисциплина, которая описывает, как правильно спроектировать физическое окружение, чтобы снизить вероятность преступлений, напрямую влияя на поведение человека. CPTED представляет собой руководство по снижению и предотвращению преступлений посредством правильной конструкции здания, окружающих компонентов и процедур.

Концепция CPTED была разработана в 1960-х годах. Она основана на влиянии окружения на типы преступлений. CPTED используется не только для разработки корпоративных программ физической безопасности, но также для крупномасштабной деятельности, такой как проектирование районов и целых городов. При этом учитывается рельеф, здания, планировки района, освещение, размещение дорог, дорожное движение и т.п. Учитываются как микроокружение (например, офисы, туалеты), так и макроокружение (например, учебные заведения и города). Концепция CPTED направлена на физическое окружение, которое может создать эффект управления поведением, снижая количество преступлений и опасность совершения преступлений. Учитываются компоненты, влияющие на взаимодействие людей и окружения. Это объединяет физические, социальные и физиологические потребности людей, использующих различные типы окружения и предсказывает поведение обычных людей и преступников.

CPTED говорит о многих вещах, о которых многие из нас не задумываются. Например, живая изгородь вокруг здания не должна быть выше 2,5 футов (0,76 метров), иначе она может позволить злоумышленнику добраться до окон. ЦОД следует размещать в центре здания, чтобы избежать угроз внешнего воздействия на его стены. Уличный инвентарь (скамейки, столы) должен размещаться таким образом, чтобы использующие его люди заодно видели происходящее вокруг, что снизит активность преступников. Ландшафт территории вокруг здания не должен иметь мест, в которых злоумышленник мог бы спрятаться. Убедитесь, что камеры видеонаблюдения хорошо заметны, чтобы преступники понимали, что их действия фиксируются, а обычные люди чувствовали себя в безопасности, видя, что все происходящее вокруг контролируется.

CPTED и нацеленность на укрепленность – это два разных подхода. **Нацеленность на укрепленность** (target hardening) сосредоточена на исключении доступа через естественные и искусственные барьеры (сигнализация, замки, ограждения и т.д.). Традиционная укрепленность может вести к ограничению удобства использования, удовольствия, эстетизма окружения. Мы, конечно же, можем установить целую систему ограждений, замков, поставить устрашающие знаки и барьеры, но что хорошего это нам даст? Если это окружение тюрьмы, возможно, это именно то, что нужно. Но если это офисное здание, его окружение не должно выглядеть как декорации Форт-Нокса, оно просто должно обеспечивать необходимый уровень защиты. Механизмы защиты должны быть едва заметны

и ненавязчивы.

Предположим, что группа решила, что вашей компании требуется защитить боковые двери здания. Традиционный ориентированный на укрепление подход предусматривает установку замков, сигнализации, системы контроля доступа со смарт-картами, камеры над дверью, проведения инструктажа охранников, контролирующих проход через дверь. Подход CPTED предлагает не прокладывать дорогу к этой двери прямо от входа на территорию компании, чтобы посетители и не пытались воспользоваться ей. Подход CPTED предлагает убедиться в отсутствии высоких деревьев или кустов, не позволяющих увидеть того, кто входит в эту дверь. Барьеры, такие как деревья и кусты могут позволить злоумышленнику чувствовать себя более комфортно, пытаясь проникнуть через эту дверь внутрь здания.

Наиболее правильным подходом было бы сначала построить окружение в соответствии с подходом CPTED, а уже потом применять повышающие укрепленность компоненты, если это необходимо.

Если многоярусный подземный гараж компании построен с учетом подхода CPTED, его лестницы и лифты будут иметь стеклянные, а не металлические стены. Это позволит людям чувствовать себя в безопасности, а потенциальным злоумышленникам не позволит совершить преступление в таком окружении. Пешеходные дороги должны быть построены таким образом, чтобы люди видели происходящее за линией машин и выявляли подозрительную активность. Сторона для парковки машин должна быть отделена низкими стенами и столбиками (а не сплошными стенами), позволяющими пешеходам видеть происходящее в гараже.

CPTED предоставляет три основных стратегии, позволяющие увеличить общую защиту за счет физического окружения и социального поведения: естественное управление доступом, естественное наблюдение и естественное укрепление территории.

Естественное управление доступом

Естественное управление доступом (natural access control) – это управление людьми, входящими и выходящими из дверей, из-за ограждений, из освещенных и других мест. Например, офисное здание может иметь внешние столбики ограждения со встроенным освещением, как показано на Рисунке 4-2. Эти столбики ограждения на самом деле выполняют несколько различных функций безопасности. Сами по себе они защищают здание от физических повреждений, предотвращая столкновения со зданием машин. Освещение не позволяет преступникам укрыться в темных местах. Освещенные столбики направляют людей по тротуару к входу без использования каких-либо знаков и указателей. Как показано на Рисунке 4-2, газоны, тротуары, светящиеся столбики и ясные видимые линии используются как естественное управление доступом. Все они работают совместно, чтобы дать людям чувство безопасности окружения и при этом сдерживают преступников.

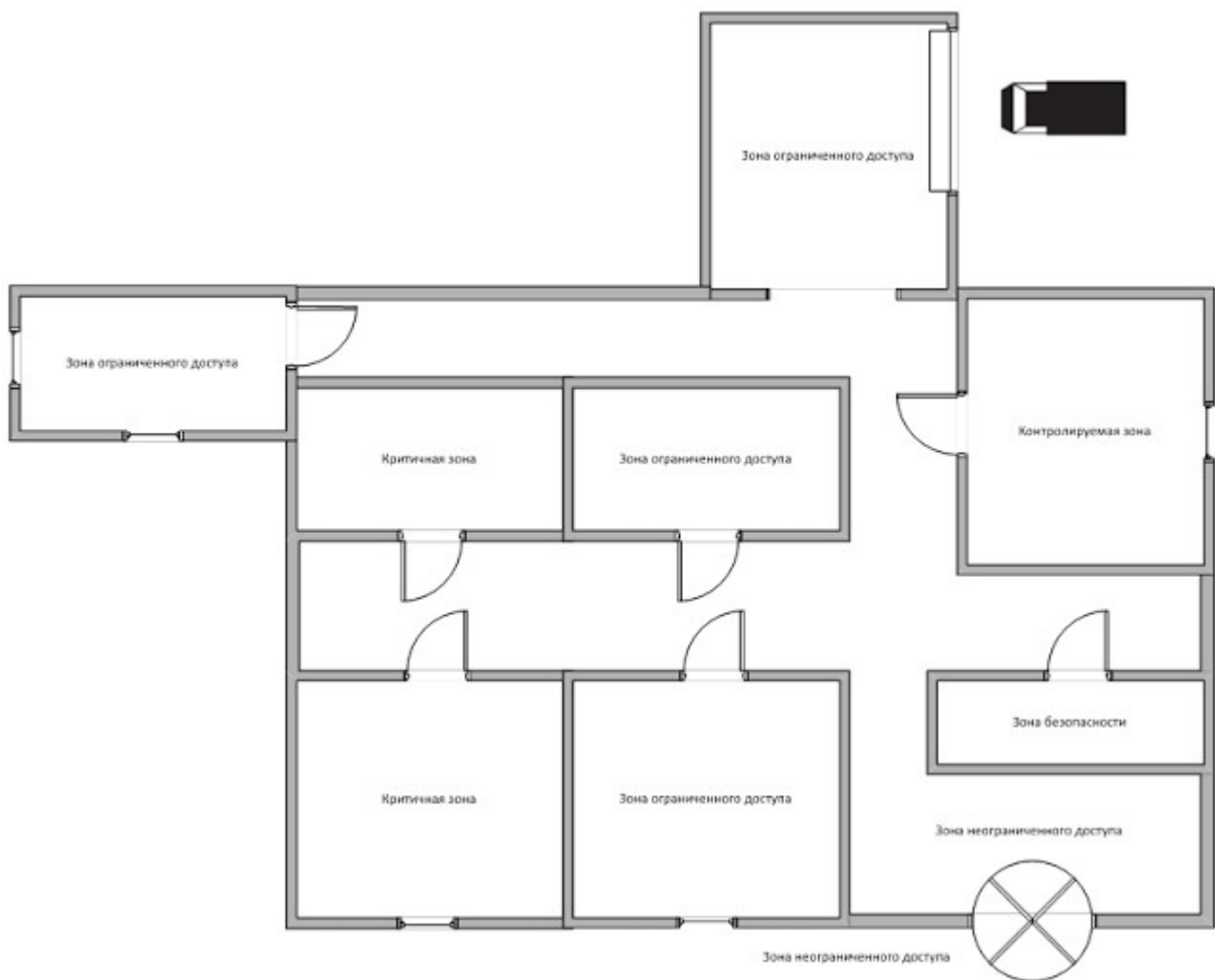


Рисунок 4-2. Тротуары, освещение, газоны могут использоваться для защиты

ПРИМЕЧАНИЕ. Короткие столбики ограждения часто используются для предотвращения въезда транспорта и для защиты пешеходов на тротуарах и здания от машин. Они могут также использоваться для направления движения пешеходов.

Свободные, просматриваемые и прозрачные маршруты могут использоваться, чтобы обескуражить потенциальных преступников, чтобы они не могли скрыть свою преступную деятельность.

Модель CPTED показывает, как можно создавать зоны безопасности. Окружающее пространство может быть разделено на зоны с различными уровнями безопасности в зависимости от того, кто должен находиться в этих зонах и от соответствующих рисков. Зоны могут быть помечены как контролируемые, ограниченного доступа, общие (неограниченного доступа) и критичные. Эта концепция похожа на классификацию данных, описанную в Домене 01. Для классификации данных создаются различные классы, соответствующие им процедуры обработки данных, определяются уровни защиты для каждого класса. То же самое справедливо и для физических зон. Для каждой зоны должен быть определен необходимый уровень защиты, который укажет необходимые типы защитных мер.



Управление доступом необходимо для ограничения и контроля возможностей перемещения людей из одной зоны безопасности в другую. Управление доступом должно быть реализовано также для всех входов в здание и выходов из него. Группа разработки программы безопасности должна учесть все возможные пути, которыми нарушитель может попасть в здание (например, взобравшись на растущее рядом дерево, а с него на крышу, верхний балкон и окно). Различные компании для управления доступом обычно используют следующие защитные меры:

- Ограничение количества точек входа
- Требование для гостей – обязательно заходить на проходную и расписываться в журнале посетителей перед входом на внутреннюю территорию
- Уменьшение количества точек входа в нерабочее время, когда вокруг не так много сотрудников
- Проверка охранниками бейджей с фотографиями перед допуском входящих в здание
- Сопровождение гостей при их перемещении по внутренней территории
- Рекомендации сотрудникам задавать вопросы незнакомцам

Барьеры, ограничивающие доступ, могут быть естественного происхождения (скалы, реки, холмы), уже существующими рукотворными сооружениями (железная дорога, автомобильная трасса) или иметь искусственное происхождение, т.е. быть созданными специально, чтобы препятствовать движению (ограждения, тупики).

Естественное наблюдение

Наблюдение также может быть организационным (охранники), техническим (видеонаблюдение) или естественным (открытые маршруты, низкий ландшафт, приподнятый вход). Цель **естественного наблюдения** (natural surveillance) – создать некомфортные условия для преступника, обеспечивая множество способов обзора, позволяющего увидеть его, а также создать чувство безопасности и комфорта всем остальным людям, предоставляя им открытое и приятное окружение.

Естественное наблюдение – это размещение и использование возможностей физического окружения для обеспечения максимальной видимости пешеходных тротуаров и всех мест, где происходят какие-либо действия. Рисунок 4-3 показывает пример открытой и просматриваемой лестницы в парковочном гараже.



Рисунок 4-3. Открытые области снижают вероятность криминальных действий

Естественное укрепление территории

Третья стратегия CPTED – это **естественное укрепление территории** (natural territorial reinforcement), которое создает физическое окружение, подчеркивающее или расширяющее физическую сферу влияния компании так, чтобы легитимные пользователи этого окружения чувствовали себя его хозяевами. Укрепление территории может быть реализовано с помощью стен, ограждений, газонов, освещения, флагов, четкого написания адресов, декоративных тротуаров. Цель укрепления территории состоит в создании чувства особого сообщества. Компании внедряют эти элементы таким образом, чтобы сотрудники чувствовали гордость за окружение и имели чувство причастности, которое они будут защищать, если понадобится. Эти элементы также внедряют, чтобы создать у потенциальных злоумышленников чувство, что они здесь лишние, что их действия рискованны, что они будут замечены, что их действия никто не будет терпеть и игнорировать.

В большинстве случаев компании используют смесь подхода CPTED и подхода, направленного на укрепленность. CPTED в основном связан со строительством зданий, их внутренним и внешним дизайном, компонентами внешнего окружения, такими как газоны и освещение. Если окружение построено на основе CPTED, дополнительные компоненты укрепленности похожи на крем сверху торта. Направленный на укрепленность подход,

использует более детальные механизмы защиты, такие как замки и детекторы движения.

В оставшейся части этого раздела мы рассмотрим физические защитные меры, которые могут быть использованы в обоих подходах.

2.2. Проектирование программы физической безопасности

Если группа физической безопасности должна оценить уровень защиты существующего здания, ей нужно проанализировать следующее:

- **Внутренние элементы:**

- Системы отопления, вентиляции и кондиционирования воздуха (HVAC)
- Материал, из которого изготовлены стены и потолки
- Системы распределения электроэнергии
- Схемы и виды коммуникаций (медь, телефон, оптоволокно)
- Использование опасных материалов

- **Внешние элементы:**

- Топография
- Близость аэропортов, автомагистралей, железных дорог
- Потенциальные электромагнитные помехи от окружающих устройств
- Климат
- Грунт
- Существующие ограждения, датчики движения, камеры, барьеры
- График рабочего времени сотрудников
- Виды операционной деятельности компании, которые зависят от физических ресурсов
- Транспортная активность
- Соседи

Чтобы надлежащим образом собрать всю эту информацию, группа должна опросить и проинтервьюировать различных сотрудников. Эта информация поможет группе оценить текущие защитные меры, выявить слабые места и убедиться, что внедрение новых защитных мер не окажет негативного влияния на производительность работы компании.

Хотя обычно существуют написанные политики и процедуры, на основе которых должна быть реализована физическая безопасность, они не всегда совпадают с реальностью. Группе важно понаблюдать, как реально используется здание, как оно защищено и какая ежедневная деятельность создает уязвимости. Собранную в процессе наблюдения информацию следует задокументировать и сравнить с тем, что указано в политиках и процедурах. В большинстве случаев вы обнаружите существенные расхождения, которые должны быть учтены и исправлены. Простое написание политики бесполезно, если ее никто не будет соблюдать на практике.

Каждая компания должна соответствовать различным требованиям, например, требованиям по охране здоровья, пожарной безопасности, государственным и местным строительным кодексам, требованиям государственных органов безопасности и охраны правопорядка, энергетическим требованиям, трудовому кодексу, требованиям регуляторов и других государственных учреждений. Группа разработки программы физической безопасности должна не только понимать все требования, которым должна соответствовать компания, а

также знать, как добиться их соблюдения посредством процедур физической безопасности.

Требования законодательства должны быть также поняты и надлежащим образом учтены. Эти требования могут включать обеспечение возможности доступа для инвалидов, вопросы, связанные с обязательствами, нарушения при защите активов и людей, применение излишней силы и т.д. Существует обширный перечень последствий нарушения законодательства компанией, если она не делает то, что должна. Иногда проблемы с законом могут принимать криминальную форму, например, если двери настроены таким образом, что при отключении электроэнергии они автоматически закрываются, то при пожаре сотрудники могут оказаться запертыми и погибнуть, вследствие чего руководству компании могут быть предъявлены обвинения в преступной халатности. Проблемы с законом могут возникнуть даже из-за того, что компания не отчистила ото льда тротуар, а прохожий упал и сломал ногу – он может подать в суд на компанию. Компанию могут признать халатной и обязать компенсировать ущерб.

Каждая компания должна иметь *ответственного за безопасность здания* (facility safety officer), который должен иметь полную информацию о здании, о потребностях компании в обеспечении защиты активов, а также о требованиях, которым должна соответствовать компания. Этот человек должен обеспечивать контроль выполнения задач по защите здания в рабочее и нерабочее время. Он должен быть вовлечен в работу группы, которая проводит оценку программы физической безопасности компании.

Программа физической безопасности – это набор защитных мер, которые внедрены и поддерживаются для обеспечения уровня защиты, необходимого для соответствия политике физической безопасности. Политика должна объединять все требования законодательства и регуляторов, которые должны быть соблюдены, чтобы уровень остаточных рисков находился на приемлемом уровне.

Теперь группа должна провести анализ рисков, который включает в себя выявление уязвимостей, угроз и оценку влияния выявленных угроз на бизнес. Этап проектирования программы следует начинать с общего описания структуры, которое будет взято за основу. Затем эта основа будет расширяться и дополняться необходимыми защитными мерами. Описание должно содержать категории программы и необходимые защитные меры. Ниже приведен упрощенный пример:

1. Сдерживание криминальной деятельности

- Ограждения
- Предупреждающие знаки
- Охранники
- Собаки

2. Задержка нарушителей, чтобы обеспечить их поимку

- Замки
- Внутренние защитные средства
- Контроль доступа

3. Выявление нарушителей

- Внешние детекторы вторжения
- Внутренние детекторы вторжения

4. Оценка ситуации

- Процедуры для охранников
- Структура коммуникаций (calling tree - дерево оповещения)

5. Реакция на вторжения и разрушения

- Команда реагирования
- Процедуры действий в чрезвычайных ситуациях
- Полиция, пожарные, медицинский персонал

Затем группа может начать поочередно рассматривать каждый из этих этапов программы безопасности здания.

Здание

Если компания решает построить здание, она должна учесть многие вещи перед заливкой первой порции бетона. Конечно, к этому моменту уже детально проанализированы цены на землю, количество проживающих в этой местности потенциальных клиентов и стратегия продаж, но нам, как специалистам по безопасности, более интересен уровень доверия и защиты, которую предоставляет данная местность. Некоторые компании, работающие с совершенно секретной или конфиденциальной информацией, делают свои здания незаметными, не привлекающими внимания потенциальных злоумышленников. Такое здание может быть трудно увидеть с близлежащих дорог, знаки компании и логотипы могут быть маленькими, не сообщающими никакой дополнительной информации, кроме простых названий и отметок, не позволяющих понять, что происходит внутри этого здания. Это тип городского камуфляжа, усложняющий поиск здания для злоумышленников.

Компании следует оценить, как далеко от здания находится полицейский участок, пожарная станция или больница. Часто близость таких учреждений заметно поднимает стоимость здания. К примеру, если химической компании, производящей мощную взрывчатку, требуется построить новое здание, лучше будет разместить его недалеко от пожарной станции (хотя персонал пожарной станции вряд ли будет счастлив). Если другая компания, которая производит и продает дорогие электронные устройства, расширилась и нуждается в переносе части операций в другое здание, время прибытия полиции может являться важным фактором, отличающим одно потенциальное место для здания от другого. Кроме того, близость любого из этих учреждений (полиция, пожарные, больница) может также уменьшить страховые ставки и позволить персоналу и руководству чувствовать себя увереннее. Помните, что наиважнейшая цель физической безопасности – это обеспечение безопасности людей. Необходимо постоянно помнить об этом при внедрении любых видов мер физической безопасности.

Некоторые здания построены в местах, окруженных холмами или горами, предотвращающими перехват электрических сигналов, излучаемых установленным в здании оборудованием. В некоторых случаях компания сама создает искусственные холмы или использует другие способы изменения ландшафта для защиты от прослушивания. Некоторые здания строят под землей или на склонах гор для сокрытия и маскировки в естественном окружении и для защиты от радарных средств, шпионской деятельности и бомбардировок с воздуха.

Вопросы при выборе места для здания. Когда выбирается место для здания, для принятия правильного решения следует обратить пристальное внимание на некоторые из следующих вопросов:

- **Видимость**
 - Окружающая местность
 - Знаки и отметки здания
 - Типы соседей

- Население в этой местности
- **Окружающая область и внешние объекты**
 - Уровень преступности, массовые беспорядки, террористические атаки
 - Близость полиции, медицинских учреждений, пожарных станций
 - Возможные угрозы окружающей среды
- **Доступность**
 - Дороги
 - «Пробки»
 - Близость аэропортов, железнодорожных вокзалов, автомагистралей
- **Природные катастрофы**
 - Вероятность наводнений, торнадо, землетрясений или извержения вулканов
 - Опасности местности (оползни, падающие с гор камни, экстремальные дожди или снегопады)

Конструкция

Планируемые конструкционные материалы и структура здания должны быть оценены на предмет того, подходят ли они для этой местности. Также следует оценить их защитные характеристики, практичность, стоимость и преимущества. Различные строительные материалы обеспечивают различный уровень огнеупорности, имеют различные пожарные рейтинги. При принятии решений в отношении конструкции, выбор типов конструкционных материалов (дерево, бетон или сталь) должен учитывать цели использования здания. Если здание предполагается использовать для хранения документов и старого оборудования, требования законодательства к нему будут сильно отличаться от требований к зданию для ежедневной работы людей.

Должна быть заранее определена и учтена в проекте предельная **нагрузка** на стены здания, полы и потолки для уверенности в том, что здание не рухнет в какой-либо ситуации. В большинстве случаев, это может быть продиктовано местными строительными кодексами. Стены, потолки и полы должны состоять из определенных материалов, чтобы соответствовать требуемому пожарному рейтингу и обеспечивать защиту от угрозы затопления. Может потребоваться, чтобы окна обеспечивали защиту от ультрафиолетовых лучей, были небьющимися, полупрозрачными или, наоборот, непрозрачными. Это зависит от размещения окна и целей использования здания. Может потребоваться обеспечить, чтобы двери (внешние и внутренние) открывались в определенном направлении, имели определенный пожарный рейтинг (как у окружающих стен), делали невозможным проход под принуждением, имели знаки аварийного выхода и, в зависимости от размещения, имели установленную и подключенную сигнализацию. В большинстве зданий используется фальшпол для скрытия и защиты проводов и труб, но нужно учитывать, что при этом такие полы должны быть электрически заземлены, поскольку они приподняты.

Все эти требования могут быть предусмотрены в строительных кодексах, но почти всегда остается несколько различных вариантов в рамках каждого требования. Группа разработки программы физической безопасности должна проанализировать эти возможные варианты для обеспечения дополнительной защиты. Выбор правильных вариантов позволит обеспечить потребности компании в части безопасности и функциональности наиболее эффективным с экономической точки зрения способом.

Заземление. Если вы возьмете в руки кабель питания, вы увидите, что вилка на нем имеет два тонких металлических контакта и один толстый. Зачем нужен толстый контакт? Это подключение заземления, которое должно работать как проводник для отвода любых превышений напряжения электрического тока, что необходимо для защиты людей и оборудования от перепадов напряжения. Как вы думаете, куда должно быть подключено заземление? Правильно, к земле. Но,

к сожалению, многие здания имеют неправильную проводку и провод заземления ни к чему не подключен. Это может быть очень опасно, так как повышенному напряжению некуда будет деваться, и он попадет в ваше оборудование или в вас.

Для обеспечения физической безопасности, в процессе проектирования и постройки здания следует учесть следующие основные вопросы:

• **Стены**

- Горючесть материала (дерево, сталь, бетон)
- Пожарный рейтинг
- Укрепления для защищенных областей

• **Двери**

- Горючесть материала (дерево, сталь, бетон)
- Пожарный рейтинг
- Сопротивление силовому проникновению
- Аварийная маркировка
- Размещение
- Запертые или контролируемые входы
- Сигнализация
- Защита петель
- Направленное открывание
- Электрические дверные замки, которые возвращающиеся в открытое состояние для безопасной эвакуации людей в случае отсутствия электропитания
- Тип стекла – требования в отношении небьющегося или пуленепробиваемого стекла

• **Потолки**

- Горючесть материала (дерево, сталь, бетон)
- Пожарный рейтинг
- Нагрузочный рейтинг
- Решения в отношении подвесного потолка

• **Окна**

- Требования полупрозрачности или затемнению
- Защита от разбивания
- Сигнализация
- Размещение
- Доступность для злоумышленников

• **Пол**

- Нагрузочный рейтинг
- Горючесть материала (дерево, сталь, бетон)
- Пожарный рейтинг

- Фальшпол (электрическое заземление)
- Изоляционная поверхность и материал
- **Отопление, вентиляция и кондиционирование воздуха**
 - Положительное (выше атмосферного) давление воздуха
 - Защищенные вентиляционные отверстия
 - Выделенные линии электропередачи
 - Аварийные запорные вентили и переключатели
 - Размещение
- **Источники электроэнергии**
 - Резервные и альтернативные источники электроэнергии
 - Чистые и постоянные источники энергии
 - Выделенные передающие линии (фидеры) в областях, где это необходимо
 - Размещение и доступ к распределительным панелям и автоматическим выключателям
- **Водопровод и газопровод**
 - Запорные вентили – помеченные и ярко окрашенные для заметности
 - Положительные потоки (вещество потока должно выходить наружу, а не внутрь здания)
 - Размещение – правильное расположение и маркировка
- **Обнаружение и тушение пожара**
 - Размещение датчиков и сенсоров
 - Размещение системы пожаротушения
 - Тип датчиков и огнетушащее вещество

Результаты анализа рисков помогут группе определить тип конструкционного материала, который следует использовать при строительстве нового здания. Есть несколько типов материалов для строительства здания. Например, **легкие конструкционные материалы** имеют меньшую огнеупорность и сопротивляемость взлому. Эти материалы изготовлены из опилок, которые горят во время пожара. Легкие конструкционные материалы обычно используются при строительстве домов, что связано, в первую очередь, с низкой стоимостью таких материалов, а также с тем, что в домах обычно не возникает таких видов пожаров и угроз вторжения, как в офисных зданиях.

Для строительства офисных зданий обычно используются **тяжелые деревянные строительные материалы**. Горючие легкие материалы из опилок также используются в конструкциях такого типа, но для них устанавливаются требования по толщине и составу материала с целью обеспечения большей огнеупорности. Конструкционные материалы должны иметь толщину не менее 4 дюймов (10,2 см). При использовании, прочное дерево плотно соединяется металлическими болтами и пластинами. В то время как конструкции из легких материалов сопротивляются огню около 30 минут, конструкции из тяжелых деревянных материалов – около часа.

Здание может быть построено из **негорючих материалов**, например, стали, которая обеспечивает более высокий уровень огнеупорности, чем упомянутые ранее материалы, но она теряет прочность при высоких температурах, что может привести к обрушению здания.

Если здание строят из *огнеупорного материала*, в материал бетонных стен и поддерживающих балок вмонтируют стальные прутья. Это обеспечивает еще большую огнеупорность и повышает защиту от силового проникновения.

Группе следует выбрать конструкционный материал на основе выявленных угроз для компании и противопожарных требований, которые компания должна соблюдать. Если компания собирается разместить в здании только нескольких офисных работников и при этом она не имеет реальных врагов, которые могут попытаться уничтожить ее здание, она может использовать легкие материалы или тяжелые деревянные материалы. Здание для правительственной организации, которой угрожают террористические акты, должно быть построено из огнеупорного материала. Финансовые компании также должны использовать огнеупорные и усиленные материалы в своих зданиях. Это особенно важно для внешних стен, в которые злоумышленники могут направить машины, чтобы пробить их и получить доступ в хранилище.

Расчеты приблизительного времени проникновения при использовании различных типов взрывчатки основываются на толщине бетонных стен и калибре использованной арматуры (под *арматурой* подразумеваются стальные прутья, вмонтированные в бетон). Даже если разрушить бетон, потребуется еще сломать или разрезать арматуру. Используя толстую арматуру и правильно размещая ее внутри бетона можно обеспечить еще большую защиту. Укрепленные стены, арматура, использование двойных стен позволяет создать задерживающие механизмы. Идея заключается в создании для злоумышленников условий, которые заставят их потратить больше времени на преодоление укрепленных стен, что даст необходимое время службам реагирования для прибытия на место и ареста злоумышленников.

Точки входа

Очень важно понимать, какие и какого типа точки входа в конкретное здание необходимы компании. Среди различных типов точек входа могут быть такие, как двери, окна, доступ через крышу, пожарные выходы, трубы и служебные точки доступа. Также, типами точек входа являются внутренние двери, разделяющие различные части здания, внешние двери, эскалаторы и лестничные колодцы. Окна на уровне земли должны быть защищены, поскольку их легко разбить. Про пожарные выходы, лестничные колодцы, выходящие на крышу, и трубы часто забывают, но они также являются потенциальными точками входа.

ПРИМЕЧАНИЕ. Вентиляционные каналы и служебные туннели могут также быть использованы злоумышленниками и поэтому должны быть соответствующим образом защищены сигнализацией и механизмами контроля доступа.

Наислабейшая часть здания, которой обычно являются окна и двери, наверняка будет атакована первой. Если взглянуть на дверь, ее слабыми местами обычно являются дверная коробка, петли и материал двери. Болты, дверная коробка, петли и материал – это именно то, что обеспечивает определенный уровень стойкости двери и ее защищенности. Например, если компания использует мощные цельнометаллические двери, но при этом они устанавливаются на слабые петли, то эти двери могут быть просто сняты злоумышленником. Таким образом, деньги на укрепленные двери компанией были потрачены зря.

Дверь, окружающие стены и потолок должны обеспечивать одинаковый уровень защиты. Если компания имеет отлично усиленную и защищенную дверь, но материалом окружающих стен является легкое дерево, это также означает, что деньги на дверь потрачены зря. Нет смысла тратить много денег только на одну контрмеру, ведь злоумышленнику проще преодолеть более слабую контрмеру, находящуюся в непосредственной близости.

Двери

Существуют различные типы дверей, предназначенные для разных целей:

- Двери хранилища

- Двери для прохода персонала
- Промышленные двери
- Двери для проезда автомобилей
- Пуленепробиваемые двери

Двери могут быть пустотелыми или полнотелыми. Группа разработки программы физической безопасности должна понимать различные типы входов и угрозы потенциального силового проникновения, что поможет ей правильно выбрать тип дверей, который следует использовать. Через пустотелые двери достаточно легко проникнуть, сломав или разрезав их, поэтому они обычно используются как внутренние. Вместо них, группа может выбрать полнотелые двери, которые изготавливаются из различных материалов, обеспечивающих различный пожарный рейтинг и различный уровень защиты от силового проникновения. Как уже было сказано ранее, пожарный рейтинг и уровень защиты двери должен соответствовать пожарному рейтингу и уровню защиты окружающих стен.

Возможно использование пуленепробиваемых дверей, если существует соответствующая угроза. Такие двери изготавливаются в виде «бутерброда» из пуленепробиваемого материала между деревянными или стальными слоями. Эти слои нужны для придания двери эстетичного вида, а пуленепробиваемый слой обеспечит необходимую защиту.

Петли и пластины замков должны быть защищены, в особенности для внешних дверей и дверей, защищающих вход в критичные помещения. Петли следует крепить заклепками, которые не могут быть удалены. Дверная коробка должна предоставлять такой же уровень защиты, как и дверь в целом.

Пожарный кодекс содержит требования по числу и размещению дверей с приспособлениями для аварийного выхода. Это перекрестные планки, освобождающие внутренний замок для открытия запертой двери. Приспособления аварийного выхода могут размещаться как на обычных входных дверях, так и на дверях аварийного выхода. Их наличие обычно отмечается специальными знаками, на которых дополнительно указывается, что эти двери не предназначены для выхода в обычных условиях. Эти двери снабжают сигнализацией, которая сработает, если кто-то откроет их.

Шлюзы и турникеты могут использоваться для того, чтобы заблокировать в них посетителей, несанкционированно проникших в здание, чтобы они не могли самостоятельно выйти.

Шлюз (mantrap) – это маленькая комната с двумя дверями. Проход через шлюз осуществляется следующим образом. В начале первая дверь заперта. Охранник, либо специализированная система (биометрическая или использующая смарт-карты) проводит идентификацию и аутентификацию входящего. После того, как личность человека проверена и доступ ему разрешен, первая дверь открывается и человек входит в шлюз. Первая дверь закрывается за ним, и человек в ловушке. Человек должен быть повторно аутентифицирован перед тем, как вторая дверь откроется и позволит ему пройти в здание. Некоторые шлюзы используют биометрические системы, которые проводят взвешивание вошедшего человека, чтобы гарантировать проход только одного человека.

Существует два варианта настройки дверей с автоматическими замками: на защиту активов, либо на безопасность персонала. Настройка двери на безопасность персонала (нормально открытая, fail-safe) означает, что при отключении электропитания дверь автоматически открывается, и работавший в помещении персонал может беспрепятственно и быстро покинуть его (например, до срабатывания газовой системы пожаротушения в случае пожара). Настройка двери на защиту активов (нормально закрытая, fail-secure) означает, что при отключении электропитания дверь остается закрытой.

Окна

Окна должны быть правильно размещены (нужно учесть, что здесь часто сталкиваются безопасность и эстетизм), иметь прочный каркас, необходимый материал стекол и, возможно, защитное покрытие. Обычно применяются следующие варианты материала стекол: стандартные, закаленные, акриловые, армированные и многослойные.

Стандартные оконные стекла (standard glass) легко разбить. Обычно такие стекла используются в жилых домах. **Закаленные стекла** (tempered glass) изготавливаются путем их нагревания и последующего быстрого охлаждения. Разбить такое стекло значительно труднее, обычно оно в 5-7 раз прочнее, чем стандартное стекло.

Акриловые стекла (acrylic glass) могут быть изготовлены из поликарбонатного акрила. Они прочнее стандартного стекла, но при нагревании они выделяют токсичные вещества. Поликарбонатный акрил прочнее обычного акрила, хотя оба они изготавливаются из прозрачного пластика. В связи с горючестью таких стекол, их использование может быть запрещено пожарным кодексом. Наипрочнейшим материалом для окон является прессованное поликарбонатное стекло. Оно противостоит широкому кругу угроз (огонь, химическое воздействие, попытки разбития), однако оно очень дорогое. Этот тип стекла следует использовать в помещениях, подверженных максимальным угрозам.

Некоторые окна изготавливают из **армированного стекла** (wired glass). В действительности это два стекла со специальной проволокой между ними. Проволока уменьшает вероятность разбития или разрезания стекла.

Многослойное стекло (laminated glass) – это два слоя стекла с пластиковой пленкой между ними. Эта пластиковая пленка затрудняет разбитие стекла. В зависимости от материала стекла и типа пластика меняется устойчивость к разбитию этого многослойного стекла.

Часто стекла имеют пленки, обеспечивающие защиту от жары и холода. Они не пропускают ультрафиолетовые лучи и обычно являются тонированными, что затрудняет злоумышленникам подглядывание и наблюдение за происходящей внутри деятельностью. Существуют пленки, обеспечивающие повышенную защиту от разбития даже на случай взрыва бомбы, шторма, попыток взлома злоумышленниками.

Типы окон. Специалисты по безопасности могут быть привлечены на этапе планирования постройки здания, чтобы учесть все аспекты, необходимые при постройке безопасного здания и его окружения. Ниже приведена обобщающая информация по всем типам окон, которые могут быть использованы.

- **Стандартное.** Без дополнительной защиты. Дешевое, обеспечивает минимальный уровень защиты.
- **Закаленное.** Стекло нагрето и затем быстро охлаждено для увеличения прочности.
- **Акриловое.** Разновидность пластика вместо стекла. Поликарбонатный акрил прочнее, чем обычный акрил.
- **Армированное.** Проволочная сетка между двумя слоями стекла. Проволока помогает предотвратить разбитие или разрезание стекла.
- **Многослойное.** Слой пластика между двумя слоями стекла. Пластик помогает защитить стекло от разбития.
- **Пленка, защищающая от солнечного света.** Обеспечивает дополнительную безопасность за счет тонирования. Кроме того, пленка защищает стекло от разбития.
- **Защитная пленка.** Прозрачная пленка, усиливающая стекло.

Внутренние помещения

Многие элементы здания должны рассматриваться с точки зрения безопасности. Внутренние перегородки используются для того, чтобы создать барьеры между различными помещениями. Эти перегородки могут использоваться для разделения отдельных помещений, но их не следует использовать для отделения защищенных помещений, в которых находятся критичные системы и устройства. Многие здания имеют фальшпотолки.

Между фальшпотолком и настоящим потолком остается некоторое пространство, в которое потенциально может проникнуть злоумышленник, подняв панель фальшпотолка. Такой пример показан на Рисунке 4-4. Во многих случаях, для этого не требуется специальных средств и серьезных усилий, а в некоторых офисных зданиях это можно сделать даже из публичных холлов). Такой тип внутренних перегородок не следует использовать для защиты критичных помещений.

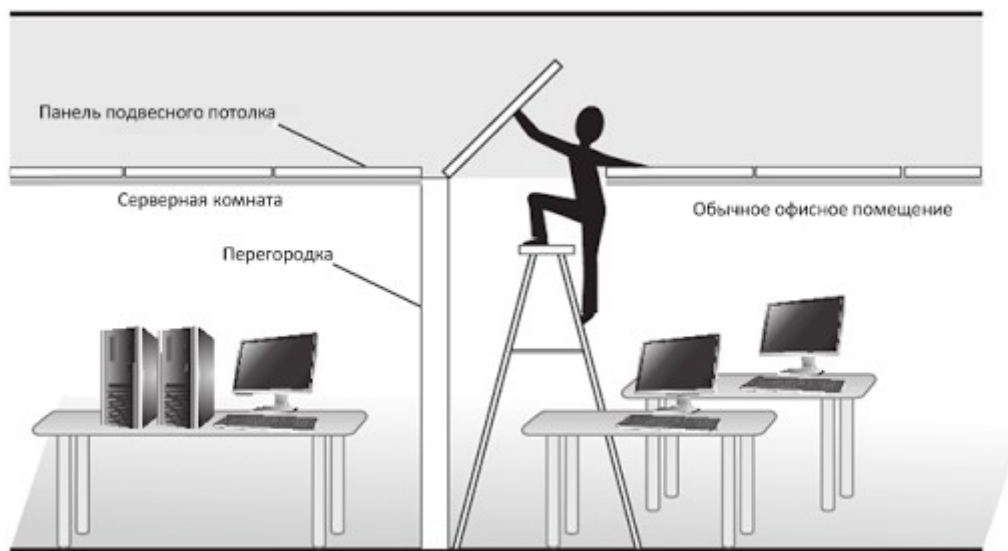


Рисунок 4-4. Злоумышленник может поднять панель подвесного потолка и проникнуть в защищенное помещение

Серверные и кроссовые помещения

Необходимо иметь персонал, отвечающий за надлежащую поддержку и эксплуатацию серверных комнат. На сегодняшний день большинство серверов, маршрутизаторов, мостов, мейнфреймов и другого оборудования размещено в серверных комнатах и всем этим оборудованием можно управлять удаленно. Это позволяет обеспечить поддержку функционирования этого оборудования небольшим количеством людей. Поскольку в серверных комнатах больше нет рабочих мест для постоянно работающего персонала, эти комнаты могут быть построены более эффективно для оборудования, а не для людей.

Небольшие устройства можно устанавливать вертикально для экономии места. Их следует монтировать в специальные серверные стойки или шкафы. Проводку следует размещать вблизи от оборудования для снижения стоимости кабелей и рисков их повреждения.

Центры обработки данных (ЦОД), серверные комнаты и кроссовые помещения следует размещать в центральных областях здания, вблизи от кабельных распределительных центров. Для ограничения доступа в такие помещения должны применяться строгие механизмы и процедуры контроля доступа, которые могут использовать смарт-карты, биометрические считыватели или комбинированные замки, как было описано в Домене 02. Такие помещения должны иметь только одну входную дверь, однако требования пожарного кодекса обычно указывают на необходимость установки двух дверей для большинства ЦОД'ов и серверных комнат. В таком случае, только одна дверь должна использоваться постоянно для входа и выхода, а другая дверь должна использоваться только в аварийных ситуациях. Эта вторая дверь не должна позволять войти в помещение – она должна обеспечивать возможность только для аварийного выхода. Она должна быть постоянно закрыта, но на ней должно быть установлено приспособление для аварийного открывания, позволяющее открыть замок при необходимости.

Желательно, чтобы входы в эти помещения ограниченного доступа не были расположены в публичных местах, таких как лестницы, коридоры, лифты, комнаты отдыха и т.д. Люди, которые подходят к дверям защищенных помещений, скорее всего, имеют соответствующие

цели, а не просто идут мимо них по пути в комнату отдыха и не стоят вокруг, разговаривая друг с другом.

Поскольку в ЦОД'ах обычно находится дорогое оборудование и критичные данные компании, их защита должна быть тщательно продумана перед реализацией. ЦОД'ы не следует размещать на верхних этажах здания, так как в них сложнее (и дольше) будет попасть аварийной команде в случае пожара. По этой же причине не следует размещать ЦОД'ы в подвале, чтобы возможное затопление не могло повредить серверы. Если здание размещено в холмистой местности, размещать ЦОД лучше выше уровня земли. ЦОД следует размещать в самом центре здания для обеспечения защиты от природных катастроф, бомб, а также для предоставления более простого доступа аварийной команде в случае необходимости.

Какие средства контроля доступа и обеспечения безопасности следует применять для защиты ЦОД'а, зависит от критичности обрабатываемых в нем данных и от требуемого уровня защиты. Сигнализация на дверях ЦОД'а должна быть включена в нерабочее время, должна существовать политика, определяющая порядок предоставления доступа в ЦОД в рабочее и нерабочее время, а также в случае аварий. Если для входа в ЦОД используется кодовая комбинация, она должна изменяться не реже, чем раз в полгода, а также сразу же после увольнения любого сотрудника, знавшего эту комбинацию.

Различные защитные меры, которые будут описаны далее, показаны на Рисунке 4-5. Группа, ответственная за проектирование нового ЦОД'а (или оценку имеющегося ЦОД'а), должна понимать все защитные меры, показанные на Рисунке 4-5, и быть способна правильно их выбирать.

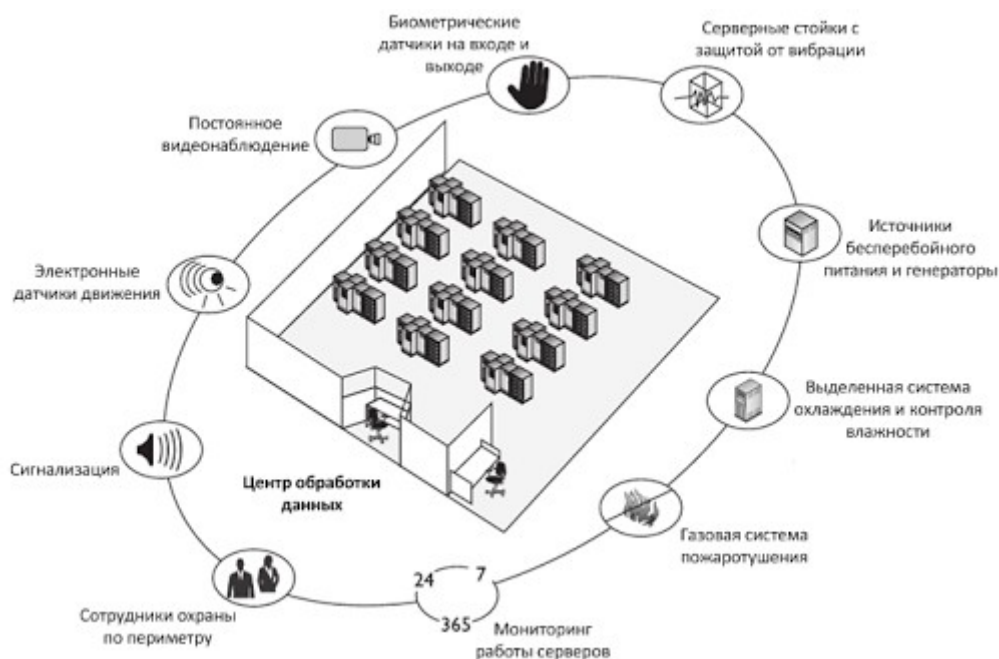


Рисунок 4-5. Для защиты ЦОД'а должно применяться множество физических защитных мер

ЦОД должен быть построен как отдельное помещение. Это помещение должно быть расположено вдали от проходящих по зданию труб водоснабжения, т.к. в случае их прорыва может произойти затопление помещений в непосредственной близости от них. Входные отверстия и трубы систем HVAC следует защищать какими-либо барьерами (например, металлическими прутьями), сами трубы должны быть сделаны тонкими, чтобы исключить возможность проникновения в ЦОД через них. В помещении ЦОД'а должно быть положительное (выше атмосферного) давление воздуха, чтобы грязь и пыль не попадали в него и в компьютерные вентиляторы.

Во многих ЦОД'ах рядом с дверью есть аварийный выключатель электропитания, чтобы

можно было отключить электропитание в случае необходимости. В случае пожара сотрудник может быстро обесточить помещение ЦОД'а и покинуть его до включения системы пожаротушения. Это особенно важно, если используемое системой пожаротушения вещество основано на воде, так как вода проводит электричество, что не очень хорошо, особенно во время пожара. Компания может установить систему пожаротушения, связанную с этим выключателем и автоматически обесточивающую ЦОД перед своим включением. (Вещество, используемое системой пожаротушения, может быть газом или водой. Газ обычно является наилучшим выбором для помещения, заполненного компьютерами. Мы рассмотрим различные вещества, используемые в системах пожаротушения, в разделе «Предотвращение, выявление и тушение пожара» далее в этом Домене).

Портативные огнетушители следует размещать поблизости от оборудования, они должны быть хорошо заметны и легкодоступны. На потолке следует устанавливать дымовые или пожарные датчики, под фальшполом – датчики воды. Важно, чтобы вода не попала под фальшпол, где размещена вся проводка, поэтому следует установить звуковую сигнализацию, реагирующую на наличие воды.

ПРИМЕЧАНИЕ. Влага в ЦОД'е или здании может привести к появлению плесени и ржавчины, что легко может превратиться в проблему. Желательно использовать промышленные осушители, влагопоглотители и т.п., чтобы избежать этой проблемы.

Вода может вызвать серьезные повреждения оборудования, пола, стен, компьютеров, фундамента здания. Очень важно обеспечить возможность своевременного выявления утечек воды. Датчики воды следует устанавливать под фальшполом и над фальшпотолком (для выявления протечек с верхних этажей). Размещение этих датчиков должно быть документировано или помечено, чтобы их можно было легко найти. Как дымовые и пожарные датчики, датчики воды должны быть подключены к системе сигнализации. Сигнализация обычно посылает сигнал тревоги только специальному персоналу, а не всем в здании. Этот персонал отвечает за анализ причин срабатывания сигнализации, и специально обучен, в т.ч. и по вопросу минимизации ущерба от потенциального воздействия воды. Перед выяснением, какие помещения могли быть затоплены, следует временно обесточить ту часть здания, где сработали датчики воды.

Датчики воды могут предотвратить повреждение:

- Оборудования
- Пола
- Стен
- Компьютеров
- Фундамента здания

Места, где следует устанавливать датчики воды:

- Под фальшполом
- Над фальшпотолком

Очень важно поддерживать надлежащий уровень температуры и влажности в ЦОД'ах, для этого в таких помещениях должны быть установлены специализированные HVAC-системы. Слишком высокая температура может стать причиной перегрева и отключения оборудования, слишком низкая температура может привести к снижению его производительности. При высокой влажности воздуха вероятна коррозия отдельных частей компьютеров, при, наоборот, низкой влажности вероятно появление статического электричества. В связи с этим, ЦОД должен иметь собственное управление температурой и влажностью воздуха, независимое от остальной части здания.

Лучше всего подключать ЦОД к отдельной от остальной части здания системе электроснабжения, если это возможно. Если возникнут проблемы с электроснабжением основной части здания, это не затронет ЦОД. Также, ЦОД может требовать наличия дублирующего электропитания, что можно обеспечить за счет двух или более вводов электричества от нескольких независимых электрических подстанций. Идея заключается в том, что если один из вводов потеряет мощность, компания будет по-прежнему получать электроэнергию с другой подстанции.

Однако наличие двух или более вводов электричества само по себе не означает, что обеспечено дублирование системы электроснабжения. Очень часто оказывается, что оба ввода идут с одной и той же подстанции, хотя компания платит за оба этих ввода! Это сводит на «нет» весь смысл нескольких вводов электричества.

ЦОД'ы должны иметь собственные системы резервного питания, либо источники бесперебойного питания (ИБП) и генераторы. Различные типы систем резервного электропитания обсуждаются далее в этом Доме. На данном этапе важно знать, что резервные системы должны обладать достаточной мощностью для поддержки работы ЦОД'а.

Многие компании делают стены ЦОД'а из больших стеклянных панелей, чтобы персонал в ЦОД'е был постоянно на виду. Это стекло должно быть небьющимся, поскольку оно выполняет функцию наружных стен. Двери ЦОД'а не должны быть полыми. Двери должны открываться наружу, чтобы не повредить оборудование при открытии. Лучшие практики говорят о том, что дверной проем должен быть закреплен в стене специальными шпильками, и что на двери должны быть установлены как минимум три петли. Эти характеристики сделают дверь более устойчивой к взлому.

3. Защита активов

Основными угрозами, с которыми борется физическая безопасность, являются кражи, перебои в предоставлении услуг, физические повреждения, нарушение целостности систем и окружения, несанкционированный доступ в помещения.

Реальные потери компании определяются стоимостью замены украденных вещей, негативным воздействием на производительность работы, снижением репутации и доверия клиентов, затратами на консультантов, а также расходами по восстановлению утраченных данных и уровня производительности. Часто компании просто проводят инвентаризацию оборудования и оформляют ведомости с указанием его стоимости, которые затем используются при анализе рисков с целью определения ущерба в случае хищения или уничтожения этого оборудования. Однако информация, хранящаяся на этом оборудовании, может быть гораздо ценнее, чем само оборудование. Поэтому для более правильной оценки ущерба, нужно учитывать и стоимость восстановления информации.

Кражи ноутбуков возросли до невообразимых масштабов за последние годы. Ноутбуки воровали годами, но раньше их воровали исключительно с целью продажи «как железа», теперь их начали воровать и с целью получения критичной информации для использования с преступными намерениями. Было украдено огромное количество ноутбуков, содержащих информацию, составляющую коммерческую тайну компаний, персональные данные граждан и даже государственную тайну. Важно понимать, что потеря ноутбука потенциально может являться очень опасной. Многие люди говорят «вся моя жизнь в моем ноутбуке (КПК)». Сотрудники используют ноутбуки во время поездок, при этом их ноутбуки содержат очень критичную информацию компании или ее клиентов, а ведь вероятность попадания таких ноутбуков в чужие руки крайне высока. Следующий список содержит ряд защитных механизмов, которые могут быть использованы для защиты ноутбуков и данных, хранящихся на них:

- Проводите инвентаризацию всех ноутбуков, учитывая их серийные номера для

правильной идентификации.

- Защищайте операционную систему.
- Защищайте BIOS паролем.
- Регистрируйте все ноутбуки с указанием производителя, уведомляйте его, если ноутбук был украден (если ноутбук принесут производителю в ремонт, он уведомит вас).
- Не сдавайте ноутбук в багаж при авиаперелетах.
- Никогда не оставляйте ноутбук без присмотра, подписывайте сумку, в которой вы его носите.
- Выгравировайте на ноутбуке символ или номер для точной идентификации.
- Пристегивайте ноутбук к массивным стационарным объектам с использованием специального приспособления (Kensington lock).
- Делайте резервные копии информации с ноутбука и сохраняйте их на стационарном компьютере или резервном носителе.
- Используйте специальные сейфы для хранения ноутбука в машине.
- Шифруйте все критичные данные.

На ноутбук может быть установлено специальное программное обеспечение, которое будет «звонить домой», если ноутбук украдут. Есть несколько продуктов, предоставляющих такую функциональность. После установки и настройки такое программное обеспечение будет периодически отправлять сигнал в контрольный центр. Если вы сообщите о краже ноутбука, производитель этого программного обеспечения будет взаимодействовать с провайдерами услуг и правоохранительными органами для поиска и возврата вашего ноутбука.

Компании следует использовать сейфы для хранения, например, лент с резервными копиями, оригиналов договоров и других ценностей. Сейф должен быть устойчивым к взлому и/или несгораемым – в зависимости от того, что в нем будет храниться. Ваша компания может выбрать сейфы одного из следующих видов:

- Сейф, встроенный в стену или пол, и легко скрывающийся
- Отдельно стоящий сейф
- Депозитарий – сейф с ячейками, в которые удобно класть ценности
- Хранилище – большой сейф, в котором можно даже ходить

Если сейф имеет кодовый замок, следует периодически менять код. Лишь небольшой круг людей может иметь доступ к коду или ключу от сейфа. Сейф следует размещать на видном месте, чтобы тот, кто использует его, был на виду. Это нужно для того, чтобы все попытки несанкционированного доступа к сейфу были заметны. Некоторые сейфы имеют возможности пассивной или термической блокировки. Если сейф имеет функцию *пассивной блокировки*, он может выявить попытку взлома, после чего сработает внутренний механизм, дополнительно запирающий дверь с помощью мощных болтов, чтобы еще больше затруднить несанкционированный доступ. Если сейф имеет функцию *термической блокировки*, то при достижении определенной температуры (например, при сверлении), сработает аналогичный механизм блокировки.

4. Внутренние системы поддержки и снабжения

Иметь укрепленное здание с отдельными защищенными помещениями это прекрасно, но иметь в этом здании свет, кондиционированный воздух и воду – еще лучше. Требования

физической безопасности должны учитывать эти поддерживающие функции, потому что их отсутствие или проблемы в их работе могут оказать негативное влияние на компанию множеством различных способов.

4.1. Электроэнергия

Компьютеры и коммуникации стали крайне важны в корпоративном мире, перебои с электроснабжением стали гораздо опаснее и разрушительнее, чем 10-15 лет назад. Поэтому компаниям необходимо иметь эффективные планы восстановления после чрезвычайных ситуаций, что позволит им минимизировать ущерб от ураганов, сильных ветров, аппаратных сбоев, ударов молнии и других событий, которые могут вызвать проблемы с подачей электроэнергии. Непрерывность электроснабжения обеспечивает доступность ресурсов компании, поэтому специалисты по безопасности должны быть знакомы с угрозами, связанными с электроэнергией, и знать о соответствующих контрмерах.

Существует несколько различных вариантов организации резервного электропитания. Для выбора оптимального варианта нужно рассчитать общий ущерб от возможного простоя и оценить его последствия. Эта информация может быть собрана из подготовленных ранее отчетов, либо получена от других организаций, находящихся в том же районе и подключенных к тому же поставщику электроэнергии. Общая стоимость часа резервного питания определяется путем деления годовых расходов (на систему резервного питания) на ежегодное среднее количество часов его использования.

К перебоям электроснабжения могут привести как крупные, так и мелкие проблемы. Это может выражаться в виде отклонения напряжения от нормы, которое может длиться от нескольких миллисекунд до нескольких дней. Компания может заплатить за получение двух различных источников электроэнергии, однако такой подход может быть достаточно дорогим. Другими, менее дорогостоящими механизмами, являются источники бесперебойного питания и генераторы. Некоторые генераторы имеют датчики, которые при выявлении проблем с электропитанием и автоматически запускают генератор. В зависимости от типа и размера генератора, он может обеспечивать электроснабжение от нескольких часов до нескольких суток. Источники бесперебойного питания – это, как правило, кратковременное решение, по сравнению с генераторами.

Защита электроснабжения

Существует три основных метода защиты от проблем электроснабжения: источники бесперебойного питания (ИБП), устройства защиты от электрических помех и резервные источники электроэнергии. ИБП использует батареи, которые отличаются размерами и емкостью. ИБП могут быть линейными или линейно-интерактивными. **Линейные ИБП** (online UPS system) используют напряжение сети переменного тока для зарядки своих батарей. В процессе работы ИБП преобразует постоянный ток со своих батарей в требующийся для оборудования переменный ток и регулирует напряжение. Процесс такого преобразования показан на Рисунке 4-6. Линейные ИБП при работе в нормальном режиме постоянно пропускают первичную электроэнергию через свои защитные механизмы. Они предоставляют потребителям электричество со своего собственного преобразователя даже при нормальном внешнем электропитании. Такой ИБП способен быстро выявить проблему электропитания, поскольку электричество проходит через него все время. Линейный ИБП может восстанавливать электропитание гораздо быстрее, чем линейно-интерактивный.

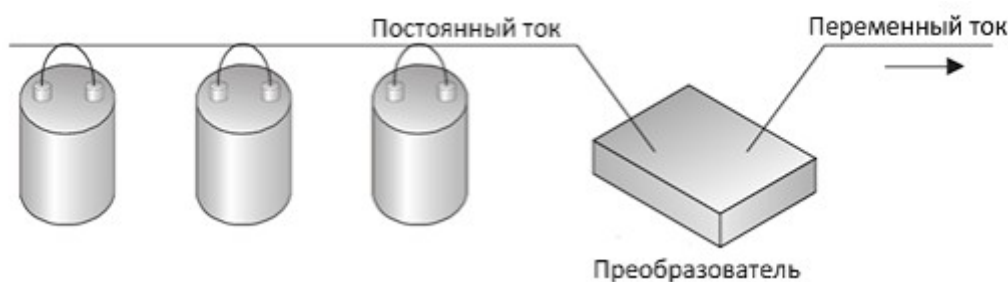


Рисунок 4-6. ИБП с помощью преобразователя преобразует постоянный ток с внутренних или внешних батарей в переменный ток

Линейно-интерактивный ИБП (standby UPS) неактивен до тех пор, пока не возникнет проблема с электропитанием. Для выявления такой проблемы ИБП имеет соответствующие датчики, в случае выявления проблемы потребитель переключается на работу от батарей. Переключение на батареи происходит с небольшой задержкой, из-за которой линейно-интерактивный ИБП восстанавливает электропитание немного медленнее, чем линейный, однако линейный ИБП стоит гораздо дороже, чем линейно-интерактивный.

Резервные источники питания требуются, когда перебои с электроснабжением продолжаются дольше, чем может проработать ИБП. Резервным источником может быть как отдельная линия, подключенная к другой электрической подстанции, так и электрогенератор, который может использоваться для питания оборудования или зарядки батарей ИБП.

Компании следует определить, какие из ее критичных систем нуждаются в защите источниками бесперебойного питания, а затем рассчитать, на какое время может потребоваться резервное питание и какая мощность нужна каждому устройству. Некоторые ИБП предоставляют электропитание только на время, достаточное для корректного выключения системы, тогда как другие могут поддерживать работу системы длительное время. Соответственно, компания должна решить, достаточно ли ей только корректного отключения систем или ей требуется продолжать выполнение критичных операций.

Для того чтобы чувствовать себя в безопасности, компании недостаточно просто иметь генератор в шкафу. Работоспособность альтернативного источника энергии должна периодически проверяться. Будет не очень приятно, если при аварии окажется, что генератор неисправен или кто-то забыл его заправить.

Проблемы электропитания

Электроэнергия обеспечивает возможности для нашей работы. Отсутствие электроэнергии даже на небольшое время может нанести нам значительный ущерб.

Чистое электропитание – это такое электропитание, в котором отсутствуют помехи и перепады напряжения. Возможными видами помех в электросети являются **электромагнитные помехи** (EMI – electromagnetic interference) и **радиочастотные помехи** (RFI – radio frequency interference). Они нарушают нормальное движение потока электроэнергии, как показано на Рисунке 4-7. EMI могут создаваться за счет разницы потенциалов между тремя проводами: напряжением, нейтралью и землей, либо за счет воздействия магнитных полей. Молния или электрические моторы могут также вызывать EMI, которые, в свою очередь, могут нарушить нормальное течение электрического тока в электросети внутри здания, а также в линии электропередачи до или после здания. RFI могут быть вызваны любым устройством, создающим радиоволны. В настоящее время одной из основных причин, вызывающих RFI в здании, являются люминесцентные лампы. Для решения этой проблемы вы можете отказаться от использования люминесцентного освещения, либо использовать экранированные провода. Электрические провода и провода компьютерной сети не следует прокладывать в непосредственной близости от люминесцентных ламп.

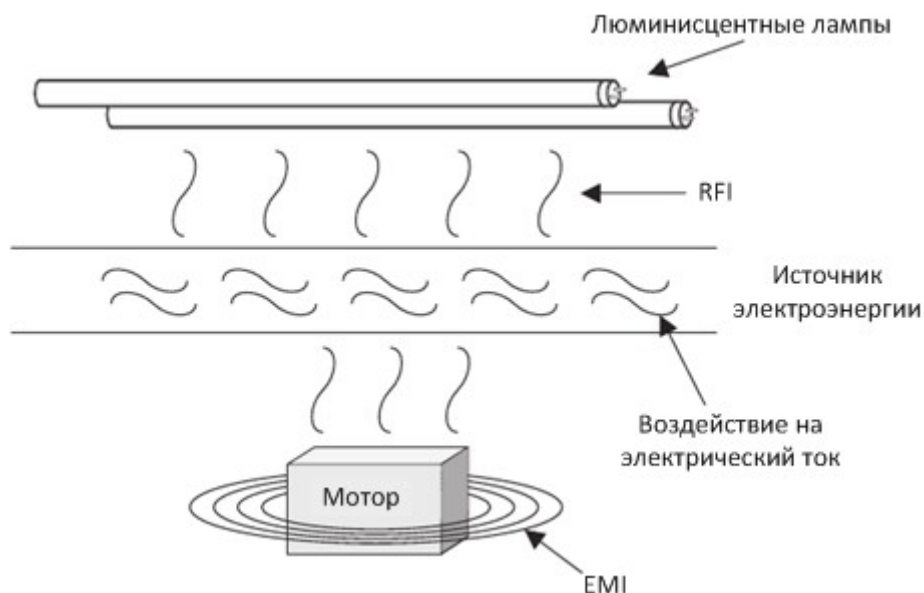


Рисунок 4-7. RFI и EMI могут вызвать помехи в электрических сетях и линиях электропередач

Определения, связанные с электроэнергией. Приведенный ниже перечень обобщает концепции, связанные с электроэнергией:

- **Земля.** Провод, связанный с землей, через который выводятся все превышения напряжения
- **Помехи.** Электромагнитные и радиочастотные помехи, которые нарушают нормальное движение электрического тока и могут вызвать перепады напряжения
- **Переходные помехи.** Кратковременные пробой линии электропередачи
- **Чистое электропитание.** Электрический ток без перепадов напряжения
- **EMI.** Электромагнитные помехи
- **RFI.** Радиочастотные помехи

Помехи нарушают нормальное движение электрического тока и могут вызвать перепады напряжения (т.е. величина напряжения будет отличаться от ожидаемой). Любой перепад напряжения может привести к повреждению оборудования и травмам людей. Далее рассматриваются различные виды перепадов напряжения электрического тока:

• **Превышение напряжения**

- **Пик (Spike).** Мгновенный (кратковременный) скачок напряжения
- **Перенапряжение (Surge).** Длительное повышение напряжения

• **Потеря напряжения**

- **Перебой (Fault).** Мгновенное (кратковременное) отсутствие напряжения
- **Отключение (Blackout).** Длительное полное отсутствие напряжения

• **Снижение напряжения**

- **Проседание/спад (Sag/dip).** Мгновенное (кратковременное) снижение напряжения от одного цикла до нескольких секунд
- **Провал (Brownout).** Длительное снижение напряжения ниже нормального уровня

• **Пусковой ток (In-rush current).** Первоначальный скачок тока, необходимый для запуска механизма

При включении электрического устройства ему может временно потребоваться повышенный ток. Это называют **пусковым током** (in-rush current). Если устройство потребляет достаточно много тока, оно может вызвать проседание напряжения для окружающих устройств, что может негативно отразиться на их работе. Поэтому, как было сказано ранее, целесообразно подключать ЦОД к отдельному (от остального здания) электрическому сегменту, поскольку в таком случае оборудование ЦОД'а не будет подвержено этому эффекту. Например, если в вашем доме проводка сделана не совсем правильно, вы можете заметить, что при включении пылесоса или микроволновой печи освещение временно становится менее ярким, что как раз и вызвано пусковым током включенного устройства. Поэтому, если в каком-то сегменте вашей электрической сети подключены устройства, потребляющие большой пусковой ток, к этому сегменту нельзя подключать системы обработки данных.

Перенапряжение (surge) – это длительное повышение напряжения от источника электроэнергии. Перенапряжение может очень быстро привести ко множеству повреждений. Перенапряжение – это одна из наиболее частых проблем электропитания, для защиты от которой используются устройства защиты от перенапряжений (surge protector - сетевой фильтр). Эти устройства используют варисторы, которые позволяют вывести излишки напряжения в землю в случае возникновения перенапряжения. Источником перенапряжения может быть сильный удар молнии, включение/выключение электрического оборудования. Большинство компьютеров имеет встроенную защиту от перенапряжения, однако возможности встроенной защиты весьма ограничены. Поэтому вы должны убедиться, что все оборудование подключено к устройствам защиты от перенапряжений, которые не позволяют повышенному напряжению попасть в защищаемое электрическое устройство.

Отключение (blackout) – это когда напряжение падает до нуля. Это может быть вызвано молнией, штормом или неоплаченным счетом за электричество. Это может продолжаться от нескольких секунд до нескольких дней. Для защиты от таких случаев, с целью обеспечения непрерывности бизнеса требуются резервные источники питания.

Когда энергетические компании сталкиваются с повышенными запросами своих потребителей, они часто снижают напряжение в электрической сети – это называют **провалом** (brownout). Для регулирования колебаний напряжения могут использоваться трансформаторы постоянного тока. Они могут работать в широком диапазоне входных напряжений и выдавать на выходе строго определенное напряжение (например, 220 вольт).

Помехи в линиях электропередач могут вызываться молниями, использованием люминесцентных ламп, работой двигателей, другими окружающими устройствами или действиями людей. Помехи могут оказать воздействие на функционирование электрического оборудования. Молнии иногда приводят к пикам напряжения в коммуникационных и электрических сетях, что может привести к повреждению оборудования или изменению передаваемых данных. Когда из-за увеличения электрической нагрузки на электростанции включаются дополнительные генераторы, это также может привести к опасным пикам напряжения.

Поскольку такие проблемы все-таки периодически случаются, должны быть внедрены механизмы, выявляющие нежелательные колебания напряжения и защищающие целостность среды обработки данных. Для обеспечения чистого и равномерного электропитания оборудования следует использовать стабилизаторы и регуляторы напряжения. Сначала электроэнергия проходит через **регулятор напряжения** (voltage regulator) или **стабилизатор** (line conditioner). Они обеспечивают отвод повышенного напряжения на случай возникновения пиков и накапливают энергию, чтобы компенсировать недостаток напряжения в случае его проседания.

Во многих ЦОД'ах используется критичное к электропитанию оборудование. Поскольку перенапряжения, проседания, провалы, отключения и пики напряжения часто становятся причиной повреждения данных, такие ЦОД'ы должны иметь высокий уровень защиты от таких проблем. Однако в ряде других случаев это не обеспечивается. В офисах часто подключают устройства различного типа к одним и тем же розеткам. Это вызывает множество помех в электросети и приводит к снижению напряжения для каждого устройства.

Превентивные меры и Хорошие практики

Следующие рекомендации могут помочь защитить электрооборудование и окружение:

- Подключайте каждое устройство к устройству защиты от перенапряжений, чтобы защитить его от повышенного напряжения
- Корректно выключайте оборудование, чтобы предотвратить потерю данных, повреждение оборудования
- Применяйте средства мониторинга электрических линий для выявления изменений частоты и напряжения электрического тока
- Используйте регуляторы напряжения для обеспечения равномерного и чистого питания
- Защитите распределительные панели, основные рубильники и трансформаторные кабели от несанкционированного доступа
- Обеспечьте защиту от магнитной индукции, используя экранированные линии
- Используйте экранированные кабели на длинных участках электропроводки
- Не прокладывайте сетевые и коммуникационные кабели поверх люминисцентных ламп
- Используйте кабели с трехконтактными вилками (с заземлением), вместо двухконтактных
- Не включайте тройники и удлинители один в другой

4.2. Проблемы окружения

Ненадлежащий контроль окружения может привести к нарушению работы сервисов, повреждению оборудования и травмам. Прерывание некоторых сервисов может привести к непредсказуемым и неблагоприятным последствиям. Такие сервисы, как электроснабжение, отопление, вентиляция и кондиционирование, а также контроль качества воздуха, могут быть сложными и состоящими из множества переменных. Они должны правильно функционировать и регулярно контролироваться.

В процессе постройки здания, группа физической безопасности должна удостовериться, что трубы подачи воды, отопления, газа имеют запорные вентили (shutoff valve) и дренажные системы (positive drain), обеспечивающие, что поток направлен наружу, а не внутрь. При повреждении трубы водопровода, перекрывающий ее вентиль должен быть легкодоступен. Сотрудники и охранники должны знать, как пользоваться этим вентилем и где он находится, им должны быть предоставлены четкие инструкции по действиям в случае чрезвычайной ситуации. Это поможет снизить потенциальный ущерб.

Большинство видов электронного оборудования должно функционировать в атмосфере с контролируемым климатом. Хотя нужно понимать, что оборудование может выйти из строя от перегрева и в контролируемой атмосфере – например, из-за поломки вентилятора в системном блоке. При нагревании устройства его компоненты расширяются, изменяются электрические характеристики контактов, что может привести к частичной потере ими своей

эффективности и вызвать повреждение всей системы.

ПРИМЕЧАНИЕ. Для сред обработки данных должны быть предусмотрены отдельные климатические системы. Информация о функционировании таких систем должна записываться и просматриваться на ежегодной основе.

Поддержка надлежащей температуры и влажности важна для любого здания, в особенности для здания, в котором установлены компьютерные системы. Неправильный уровень этих параметров может привести к неисправности компьютеров и других электрических устройств. Высокая влажность может вызвать коррозию, а низкая привести к статическому электричеству. Статическое электричество может привести к замыканию в устройствах и их повреждению.

ПРИМЕЧАНИЕ. Следует поддерживать влажность в диапазоне от 40% до 60%, а температуру – в диапазоне от 21°C до 23°C.

Низкая температура может понизить производительность систем, высокая – вызвать периодические перезагрузки системы из-за перегрева. В Таблице 4-1 приведен список различных компонентов и уровень температуры, который может привести к их повреждению.

Материал или Компонент	Температура, вызывающая повреждение
Компьютерные системы и периферийное оборудование	80°C
Магнитные носители информации	40°C
Бумажные документы	175°C

Таблица 4-1. Компоненты, для которых важна температура

В сухом климате или зимой воздух менее влажен, что может вызвать статическое электричество (напряжение разряда может составлять до нескольких тысяч вольт). Это может быть более опасно, чем вы думаете. Если вы коснетесь не заземленного корпуса, а внутренних компонентов компьютерной системы, это может привести к выходу ее из строя. Люди, которые работают с внутренним монтажом компьютера, обычно одевают антистатические повязки на руки.

В более влажном климате или летом, повышенная влажность воздуха также может негативно воздействовать на компоненты, приводя к снижению электрической эффективности контактов. Для контроля влажности обычно используют *гигрометр*. Автоматические гигрометры могут отправлять сигнал тревоги в случае достижения уровнем влажности некоего порогового значения.

Превентивные шаги против статического электричества. Ниже представлен простой список мер для предотвращения образования статического электричества:

- Используйте антистатические покрытия пола в помещениях обработки данных
- Обеспечьте необходимый уровень влажности
- Обеспечьте наличие заземления для проводки и розеток
- Не используйте ковровые покрытия в ЦОД'ах, в случае необходимости используйте антистатические ковры
- Одевайте антистатические повязки при работе с внутренним монтажом компьютерных систем

4.3. Вентиляция

К вентиляции воздуха применяется ряд требований, позволяющих обеспечить безопасную и комфортную среду. Для поддержки качественного воздуха следует использовать рециркуляционные кондиционеры (т.е. кондиционеры, использующие только воздух внутри здания и проводящие его очистку, а не забирающие воздух извне). Также должны использоваться средства, поддерживающие положительное (выше атмосферного) давление, вентиляцию и контроль загрязнения воздуха. *Положительное давление* (positive

pressurization) означает, что при открытии двери воздух выходит из помещения наружу, а не наоборот. Это особенно важно при пожаре, чтобы дым выходил наружу.

Группа физической безопасности должна понимать, какие виды загрязнений могут попасть в воздух внутри помещений, а также последствия, к которым это может привести, и шаги, которые нужно предпринять для защиты от опасных веществ или высокого уровня загрязнений. Должен контролироваться приемлемый уровень загрязняющих частиц, находящихся в воздухе. Пыль также может привести к повреждению оборудования. Высокая концентрация в воздухе некоторых веществ может вызвать повышенную коррозию, оказать влияние на производительность систем или вывести из строя электронное оборудование. Для противодействия этим угрозам применяются системы вентиляции и очистки воздуха.

4.4. Предотвращение, выявление и тушение пожара

Предмет физической безопасности был бы раскрыт не полностью без обсуждения вопросов противопожарной безопасности. Компании должны соблюдать государственные и местные стандарты предотвращения, выявления и тушения пожара. ***Предотвращение пожара*** включает в себя обучение сотрудников правильной реакции в случае пожара, поддержку соответствующего противопожарного оборудования в исправном состоянии, надлежащее хранение горючих материалов. Предотвращение пожара также может включать в себя использование негорючих конструкционных материалов, планировку здания, обеспечивающую противодействие распространению огня и дыма.

Существуют различные виды систем реагирования на выявление пожара. Есть ручные средства – красные кнопки на стенах. Автоматические системы реагирования на выявление пожара используют сенсоры, реагирующие на появление огня или дыма. Мы рассмотрим различные виды систем обнаружения пожара в следующем разделе.

Системы пожаротушения используют специальные тушащие вещества (suppression agent). Средства пожаротушения могут быть ручными (портативные огнетушители) или автоматическими (спинклерные системы, либо газовые системы на основе хладона или углекислого газа). Различные типы тушащих веществ будут рассмотрены позднее в разделе «Тушение пожара». Наиболее широко используются автоматические спинклерные системы, они весьма эффективно защищают здание и его содержимое. При принятии решения о выборе системы пожаротушения, компании нужно оценить множество факторов, включая оценку предполагаемого рейтинга пожара, наиболее вероятного типа пожара, величины предполагаемых повреждений в результате пожара, а также доступные для выбора типы систем пожаротушения.

Процессы обеспечения противопожарной защиты должны включать в себя внедрение средств раннего оповещения при выявлении огня или дыма, а также системы отключения оборудования при повышении температуры. При обнаружении огня или дыма звуковое оповещение о пожаре должно срабатывать до включения системы пожаротушения, чтобы персонал мог успеть отключить ее в случае ложной тревоги или мелкого пожара.

Рейтинги огнеупорности. Рейтинги огнеупорности устанавливаются в результате проведения лабораторных тестов в определенной среде. American Society for Testing and Materials (ASTM) является организацией, которая создает стандарты для проведения этих тестов и интерпретации их результатов. ASTM проводит аккредитацию тестовых центров, выполняющих оценку в соответствии с этими стандартами и присваивающих рейтинг огнеупорности, который затем используется в федеральных и государственных пожарных кодексах. В ходе тестов оценивается огнеупорность различных материалов в различных условиях. Система рейтингов используется для классификации различных компонентов зданий.

Типы пожарных датчиков

Огонь представляет собой серьезную угрозу безопасности, т.к. может повредить данные или оборудование, а также представляет опасность для жизни людей. Дым, высокие температуры, коррозионные газы могут привести к ужасным последствиям. Поэтому очень

важно оценить пожарную безопасность здания и его отдельных помещений.

Источником пожара может быть неисправность электрического оборудования, ненадлежащее хранение горючих материалов, неисправность обогревателя, неосторожное обращение с огнем (например, при курении), поджог. Огню нужно топливо (бумага, дерево, горючая жидкость и т.п.) и кислород. Чем больше топлива на квадратный метр, тем более интенсивен огонь. Здание должно быть построено и содержаться таким образом, чтобы минимизировать концентрацию топлива для огня.

Существует четыре класса пожара (А, В, С и D), которые обсуждаются далее в разделе «Тушение пожара». Необходимо знать разницу между этими классами, чтобы понимать, как их правильно тушить. Портативные огнетушители имеют маркировку, обозначающую классы пожара, для которых они предназначены. Портативные огнетушители должны находиться не далее 50 футов (15 метров) от любого электрического оборудования, а также рядом с выходами. Огнетушители должны быть промаркированы, маркировка должна быть хорошо заметна. Они должны быть легкодоступны, работоспособны, их следует проверять ежеквартально.

Многие компьютерные системы делают из негорючих материалов, но при перегреве они могут плавиться или обугливаться. Большинство компьютерных схем использует постоянное напряжение от 2 до 5 вольт, что не может вызвать пожар. Если пожар происходит в серверной комнате, скорее всего это будет пожар, связанный с электричеством, вызванный перегревом изоляции проводов или окружающих пластиковых компонентов. Перед возгоранием обычно происходит длительное задымление.

Существует несколько видов датчиков, работающих различными способами. Датчик может срабатывать на дым или повышение температуры.

Датчики, срабатывающие на дым (дымовые извещатели), являются хорошим средством раннего оповещения. Они могут использоваться для активации системы голосового оповещения перед включением системы пожаротушения. Фотоэлектрические устройства, также называемые оптическими датчиками, выявляют отклонения в интенсивности света. Детектор излучает луч света через защищаемую область и если он искажен, включается тревога. Рисунок 4-8 иллюстрирует работу фотоэлектрического устройства.

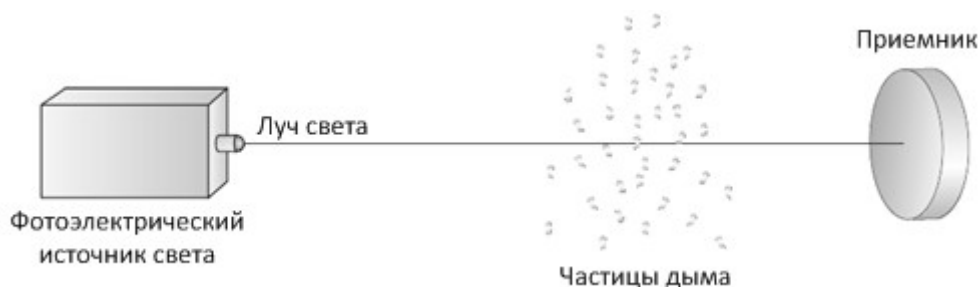


Рисунок 4-8. В фотоэлектрическом устройстве находится источник света и приемник

Другой тип фотоэлектрического устройства берет образцы окружающего воздуха, пропуская их через трубу. Если воздух непрозрачный, включается тревога.

Датчики, срабатывающие на температуру (тепловые извещатели), могут быть настроены на включение сигнала тревоги при достижении заранее определенной температуры или при повышении температуры в течение определенного временного интервала (такие датчики обычно срабатывают быстрее, т.к. они более чувствительны, однако они чаще допускают ложные срабатывания). Датчики могут быть бессистемно распределены по зданию или установлены линейно, работая как чувствительный к температуре кабель.

Недостаточно просто иметь в здании пожарные датчики – они должны быть правильно размещены. Датчики следует устанавливать над и под фальшпотолком и под фальшполом,

поскольку компании прокладывают множество видов проводки именно в этих местах и именно в них может начаться пожар, вызванный электричеством. Никто не узнает о пожаре, пока он не проявится из под фальшпотолка или над фальшполом, если в этих местах не установлены датчики. Датчики следует также располагать в отопительных и воздушных коробах, поскольку именно в них вероятнее всего будет скапливаться дым, прежде чем попасть в другие места. Очень важно оповестить людей о пожаре как можно быстрее, а также в кратчайшее время активировать системы пожаротушения. Это поможет сохранить жизнь людей и минимизирует ущерб. На Рисунке 4-9 показано правильное размещение дымовых датчиков.

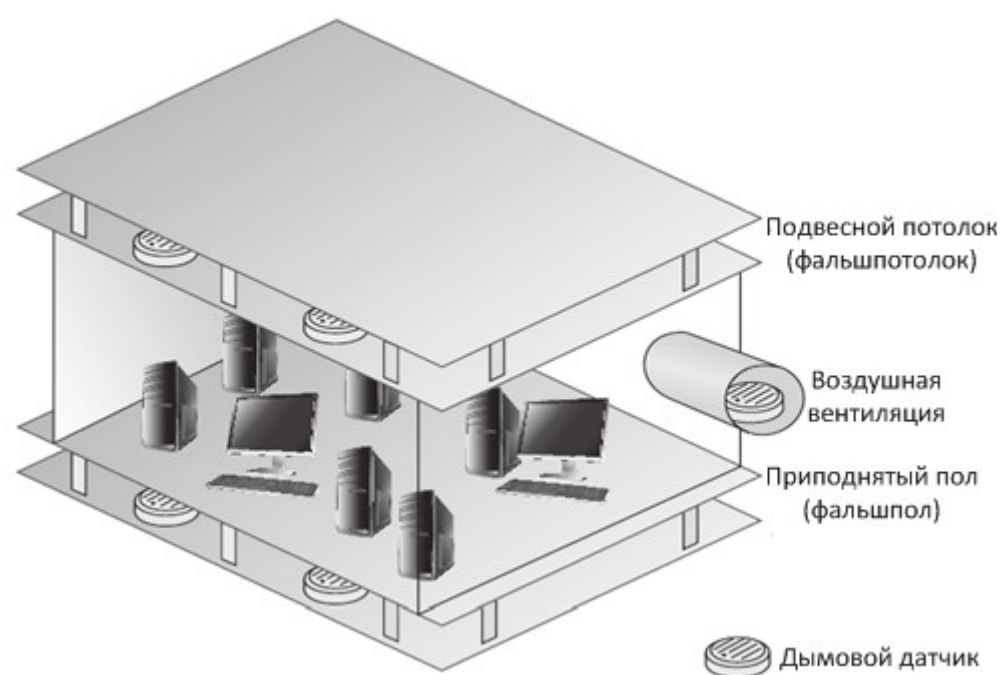


Рисунок 4-9. Дымовые датчики следует размещать под фальшполом, над фальшпотолком и в коробах воздушной вентиляции

Замкнутые пространства. Проводку и кабели протягивают в замкнутом пространстве, таком как пространство над фальшпотолком, полости в стенах, пространство под фальшполом. В этих замкнутых пространствах (Plenum areas) должны быть установлены пожарные датчики. Кроме того, в замкнутых пространствах могут использоваться только пригодные для этого провода, сделанные из материалов, не выделяющих опасных газов при горении.

Автоматическая передача сигнала тревоги по коммутируемому каналу. Системы выявления пожара могут быть настроены для автоматического звонка на местную пожарную станцию и, возможно, в ближайший полицейский участок, для сообщения об обнаруженном пожаре. Системы воспроизводят предварительно записанное сообщение, которое передает необходимую информацию, позволяющую официальным лицам должным образом подготовиться к работе в чрезвычайной ситуации и прибыть в нужное место.

Тушение пожара

Важно знать не только различные типы пожаров, но и как правильно тушить их. Каждый тип пожара имеет рейтинг, указывающий на то, какие горят материалы. Таблица 4-2 показывает четыре типа пожаров и методы их тушения, которые следует знать всем сотрудникам.

Класс пожара	Тип пожара	Горящие материалы	Метод тушения
A	Обычные горючие вещества	Деревянные материалы, бумага, ламинат	Вода, пена
B	Жидкости	Нефтепродукты, охлаждающие вещества	Газ, CO ₂ , пена, порошок
C	Электрический пожар	Электрическое оборудование и провода	Газ, CO ₂ , порошок
D	Горючие металлы	Натрий, магний, калий	Порошок

Таблица 4-2. Четыре типа пожара и методы их тушения

Существует много различных способов тушения пожара, но все они требуют определенных мер предосторожности. Во многих зданиях системы пожаротушения расположены в различных местах и настроены на автоматическое включение в случае срабатывания определенного датчика (или триггера). Каждая система имеет свою зону действия. В случае возгорания в некоторой зоне, включается соответствующая этой зоне система и тушит пожар. Существуют различные типы тушащего вещества – CO₂ (сжиженный углекислый газ), вода, хладон, пена и другие. CO₂ хорошо гасит огонь, но он не совместим с жизнью людей. Если компания использует CO₂, система пожаротушения должна иметь систему звукового оповещения и задержки включения, дающую время персоналу покинуть опасную зону перед включением системы пожаротушения. CO₂ не имеет цвета и запаха, он удаляет кислород из воздуха, что приведет к гибели людей, если они своевременно не покинут помещение. Поэтому такой тип системы пожаротушения лучше всего использовать в безлюдных зданиях или помещениях.

Для пожаров класса В и С могут использоваться некоторые виды сухого порошка, который содержит бикарбонат натрия или калия, карбонат кальция или фосфат моноаммония. Первые три вида порошка останавливают химическую реакцию горения. Фосфат моноаммония снижает температуру и удаляет кислород, являющийся топливом для пожара.

Пена в основном изготавливается на водной основе, она содержит пенообразующее вещество, создающее пену на поверхности горящего вещества и удаляет кислород.

Хладон (halon) – это газ, который широко используется для тушения пожаров, т.к. он препятствует химической реакции горения. Он быстро смешивается с воздухом и не наносит вреда компьютерным системам и другим устройствам обработки данных. Он используется в основном в ЦОД'ах и серверных комнатах. Однако хладон поглощает озон, а его концентрация в воздухе свыше 10% опасна для человека. Хладон используется при исключительно высокой температуре пожара, в котором горят токсичные химические вещества, еще более опасные для человека. Поскольку хладон является опасным веществом, он постепенно заменяется нетоксичными системами пожаротушения. Ниже приведен список одобренных ЕРА заменителей хладона:

- FM-200
- NAF-S-III
- СЕА-410
- FE-13
- Вода
- Инерген
- Аргон
- Аргонит

ПРИМЕЧАНИЕ. В действительности существует класс пожара К для коммерческих кухонь. Эти пожары следует тушить жидкими химическими веществами – в большинстве случаев солями калия. Химические вещества работают лучше, выводя кислород из зоны пожара.

Огню нужно топливо, кислород и высокая температура. В таблице 4-3 показано как различные вещества, используемые для тушения пожара, противодействуют его компонентам.

Компоненты пожара	Метод тушения	Как происходит тушение
Топливо	Углекислый натрий (soda acid)	Удаляет топливо
Кислород	CO ₂ (углекислый газ)	Удаляет кислород
Температура	Вода	Снижает температуру
Химическая реакция горения	Газ хладон (или его заменители)	Препятствует химической реакции между элементами

Таблица 4-3. Как различные вещества препятствуют компонентам пожара

ПРИМЕЧАНИЕ. Хладон не производится с 1 января 1992 года в соответствии с международным

соглашением. Монреальский Протокол запретил хладон в 1987 году и 5 лет понадобилось различным странам, чтобы реализовать эту директиву. Самый эффективный заменитель хладона – FM-200, который очень похож на хладон по всем характеристикам, однако не наносит вреда озону.

Для правильного и своевременного отключения систем HVAC в случае пожара, следует соединять их с пожарной сигнализацией и системами пожаротушения. Огню нужен кислород, а работающие системы HVAC будут доставлять его в зону пожара. Кроме того, системы HVAC могут распространять смертельно опасный дым в другие помещения здания. Многие противопожарные системы настраивают на автоматическое отключение системы HVAC при срабатывании сигнала пожарной тревоги.

Водяные спринклеры

Водяные спринклеры (water sprinkler) обычно проще и дешевле, чем системы, использующие хладон или FM-200, но в них используется вода, которая может нанести ущерб оборудованию. В случае пожара, вызванного электричеством, вода может только увеличить интенсивность огня и ухудшить ситуацию, поскольку вода проводит электрический ток. При использовании воды в любом окружении с электрическим оборудованием, электричество обязательно должно отключаться перед включением воды. Для отключения электричества следует использовать датчики, работающие в автоматическом режиме. Каждый спринклер следует включать отдельно, чтобы избежать широкомасштабных повреждений, кроме того, следует предусмотреть специальные вентили, которые позволят прекратить (предотвратить) подачу воды при необходимости.

Компании следует очень внимательно отнестись к выбору наиболее подходящего именно ей варианта системы пожаротушения и используемого в ней тушащего вещества. Доступны четыре основных вида водяных спринклерных систем: с «мокрыми» трубами, с «сухими» трубами, упрещающие и поточные.

- **Системы с «мокрыми» трубами** (wet pipe) всегда держат воду в трубах и, как правило, включаются по команде датчика, контролирующего температуру. Единственным недостатком таких систем является то, что вода может замерзнуть в холодном климате. Кроме того, при повреждении сопла или прорыве трубы это может привести к затоплению и причинить большой ущерб. Системы такого типа также называют системами с замыкающей головкой (closed head system).
- В **системах с «сухими» трубами** (dry pipe), вода в трубах не находится постоянно. Вода содержится в специальном резервуаре, пока в ней не возникнет необходимость. В трубах находится воздух под давлением, который вытесняется давлением воды при срабатывании сигнализации в случае пожара или задымления. Сначала срабатывает температурный датчик или датчик задымления, затем вода заполняет трубы, устремляясь к спринклерам, звучит сигнал тревоги, отключается электрическая энергия, включаются спринклеры для тушения пожара. Такие системы лучше подходят для холодного климата, так как вода в трубах не замерзает. На Рисунке 4-10 показана система с «сухими» трубами.
- **Упрещающие** (preaction) системы похожи на системы с сухими трубами в том плане, что в них также вода не находится постоянно в трубах, а попадает в трубы только при снижении давления воздуха в них. Когда это происходит, трубы заполняются водой, но она не разбрызгивается в тот же момент. В головку спринклера встроена тепловая пробка, расплавляющаяся при повышении температуры. Целью такой системы является предоставление людям большего времени для реакции на ложную тревогу или, в случае небольшого пожара, тушения его вручную. Тушение небольшого пожара ручными огнетушителями более предпочтительно, поскольку позволяет избежать повреждения водой электрического оборудования. Эти системы обычно применяются только в помещениях обработки данных, так как они имеют самую высокую стоимость среди систем подобного типа.

- **Поточные** (deluge) системы имеют более широкие спринклерные головки, что позволяет им выпускать больший объем воды в единицу времени (в связи с этим они не используются в помещениях обработки данных).

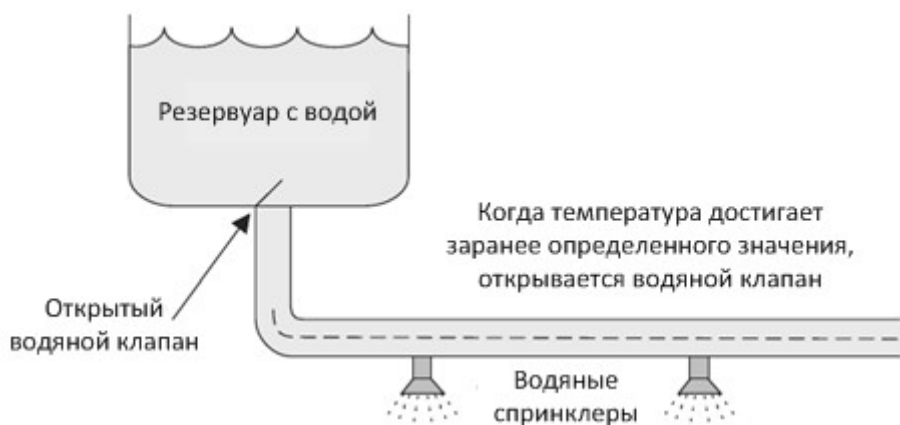


Рисунок 4-10. В системах с «сухими» трубами вода не содержится в трубах постоянно

5. Безопасность периметра

Первая линия защиты – это контроль периметра здания для предотвращения несанкционированного доступа в него. Как говорилось ранее в этом Домене, физическая безопасность должна использовать многоуровневый подход к защите. Например, перед тем, как нарушитель сможет получить записи о секретном рецепте приготовления соуса барбекю, принадлежащем вашей компании, он должен перелезть через ограждение, скрыться от охранников, взломать дверной замок, обмануть биометрическую систему контроля доступа во внутреннее помещение, а затем взломать сейф, в котором хранится рецепт. Идея заключается в том, что если атакующий преодолевает один уровень защиты, на его пути к ценностям компании, должен существовать еще один (как минимум) уровень защиты.

ПРИМЕЧАНИЕ. Важно иметь отдельные и независимые защитные меры. Например, если один ключ подходит к замкам четырех дверей, злоумышленнику может хватить только одного ключа. Каждый замок должен иметь свой ключ или аутентификационную комбинацию.

Эта модель защиты должна работать в двух основных режимах: в рабочее время компании и в нерабочее. Когда здание в нерабочее время закрыто, все двери должны быть заперты, а средства мониторинга включены для выявления подозрительной активности. В рабочее время обеспечение безопасности усложняется, так как уполномоченный персонал нужно отличать от неуполномоченных посетителей. Безопасность периметра связана со зданием, контролем доступа персонала, механизмами защиты внешних границ, выявлением вторжений, корректирующими действиями. Следующие разделы описывают элементы, которые входят в состав этих категорий.



5.1. Контроль доступа в здание и помещения

Контроль доступа в рамках физической безопасности должен обеспечиваться физическими и техническими компонентами. Физический контроль доступа использует механизмы идентификации людей, пытающихся войти в здание или в помещение. Он разрешает проход уполномоченным посетителям и блокирует доступ неуполномоченным, а также предоставляет журнал регистрации этих действий. Наличие персонала в наиболее критичных помещениях – это одна из лучших мер безопасности, так как люди в таком помещении могут самостоятельно выявить подозрительное поведение. Однако их нужно предварительно обучить, какую деятельность считать подозрительной и как сообщать о ней.

Перед тем, как компания внедрит необходимые механизмы защиты, необходимо провести детальный анализ и определить какие люди в какие помещения должны иметь доступ. Должны быть определены точки контроля доступа, затем эти точки должны быть классифицированы как внешние, основные и запасные входы. Персонал должен входить и выходить только через определенные входы, доставка должна осуществляться через другие входы, доступ в критичные помещения должен быть ограничен. На Рисунке 4-11 показаны различные типы точек контроля доступа в здании. После определения и классификации точек контроля доступа, следующим шагом является определение порядка их защиты.

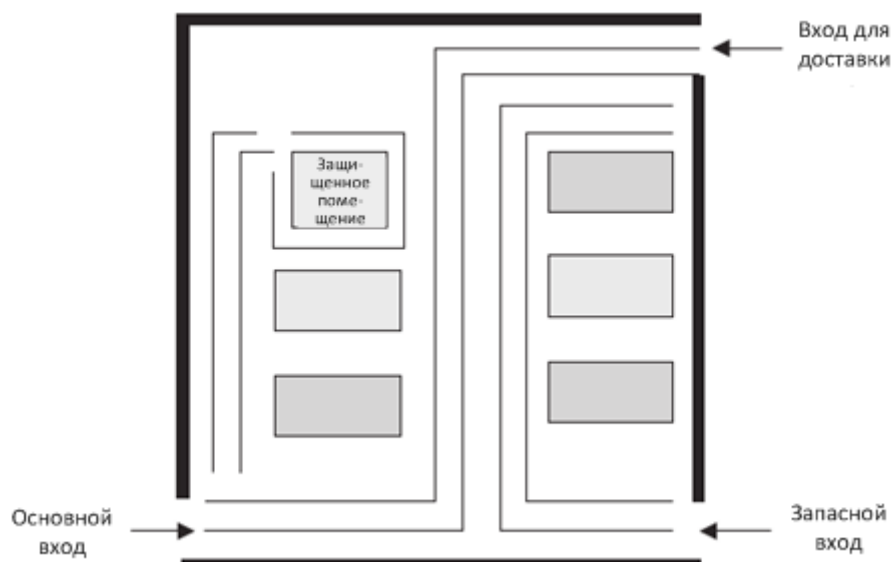


Рисунок 4-11. Точки контроля доступа должны быть идентифицированы, промаркированы и надлежащим образом контролироваться

Замки

Замок – это недорогой, общепризнанный и широко используемый механизм контроля доступа. Замок является задерживающим устройством по отношению к нарушителям. Чем больше времени займет взлом замка, тем больше времени будет у охраны и полиции, чтобы приехать на место в случае выявления нарушителя. Почти любая дверь оснащена замком, но ключи легко потерять или скопировать, а сам замок может быть сломан или взломан. Если компания применяет для защиты только механизмы с замками и ключами, любой, кто имеет ключ, может беспрепятственно войти и выйти, избежав какого-либо контроля. Замки следует использовать только как часть схемы защиты, но не как единственный компонент этой схемы.

Замки имеют различный функционал. Навесные замки (padlock) могут использоваться на воротах в изгороди, врезные замки (preset lock) обычно используются в дверях, кодовые замки используются в дверях и хранилищах. Замки бывают различных видов и имеют различные размеры. Важно использовать правильный тип замков, это позволит обеспечить нужный уровень защиты.

Для взломщика замок – это маленькая головоломка, а не средство устрашения. Вы должны обеспечить сложность этой головоломки, ее стойкость и высокое качество запирающего механизма.

ПРИМЕЧАНИЕ. Задержка времени, обеспечиваемая замком, должна соответствовать сопротивляемости проникновению окружающих элементов (дверей, дверных рам, петель). Умный вор пойдет по пути наименьшего сопротивления – не обязательно взламывать замок, если можно удалить штифты из петель или просто ударить по двери.

Механические замки

Существует два основных типа механических замков – с нарезкой и цилиндровые. **Замок с нарезкой** (warded lock) – это обычный висячий замок, показанный на Рисунке 4-12. Он имеет пружинную задвижку с вырезанным в ней пазом. Ключ заполняет этот паз, и задвижка смещается из заблокированного в разблокированное положение. Внутри замка имеются металлические выступы (ward) вокруг замочной скважины, как показано на Рисунке 4-13. Правильный ключ для такого замка имеет вырезы, которые совпадают с этими выступами и пазом, что позволяет задвижке двигаться вперед и назад. Это самые дешевые замки, которые легче всего взломать.

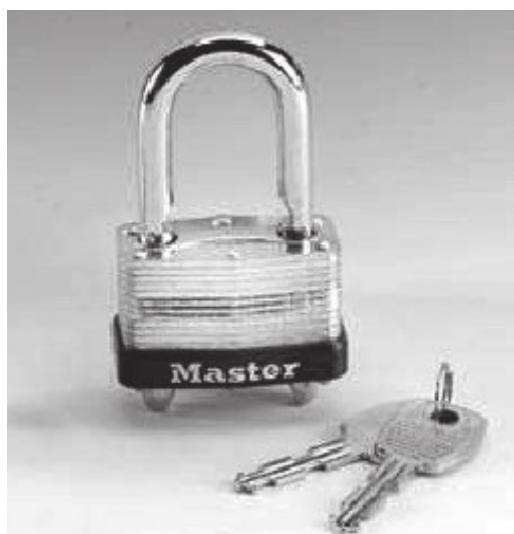


Рисунок 4-12. Замок с нарезкой

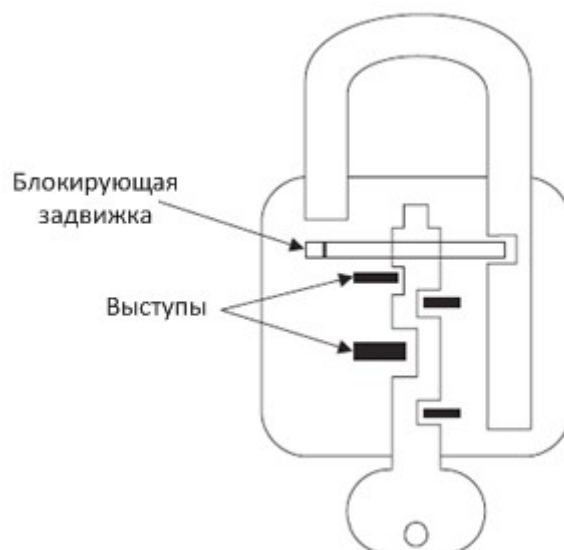


Рисунок 4-13. Ключ заполняет паз для перемещения задвижки в заблокированное или разблокированное положение

Цилиндровый замок (tumbler lock) имеет больше частей и элементов, чем замок с нарезкой. Как показано на Рисунке 4-14, ключ входит в цилиндр, в котором поднимаются запорные металлические штифты (пины) на нужную высоту, чтобы задвижка могла переместиться в заблокированное или разблокированное положение. После того, как все металлические штифты встали на правильный уровень, внутренний цилиндрический механизм может быть повернут. Правильный ключ имеет правильную длину и последовательность выступов, обеспечивающих перемещение металлических штифтов в правильное положение.

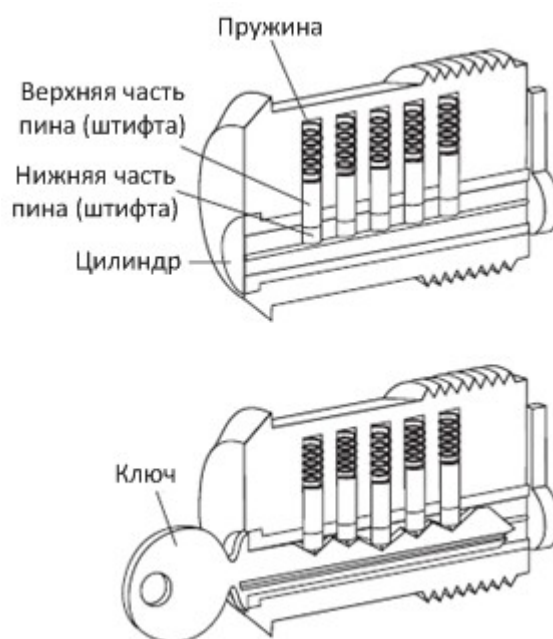


Рисунок 4-14. Цилиндровый замок

Существует три вида цилиндровых замков – пиновые (штифтовые), пластинчатые и сувальдные (lever tumbler lock). **Пиновый цилиндровый замок** (pin tumbler lock) показан на Рисунке 4-14, это наиболее часто используемый тип цилиндровых замков. Ключ должен иметь соответствующие канавки, которые поставят все пружинные штифты в правильное положение, необходимое для блокировки или разблокировки замка.

Пластинчатые цилиндрические замки (wafer tumbler lock) (также называемые дисковыми цилиндрическими замками) представляют собой небольшие круглые замки, которые вы обычно

видите на картотечных шкафах. Они используют плоские диски (пластины) вместо штифтов внутри замка. Они часто используются в качестве автомобильных замков или замков в письменном столе. Такой тип замков не предоставляет существенной защиты, поскольку его можно легко взломать.

ПРИМЕЧАНИЕ. Некоторые замки имеют сменные сердечники, т.е. они предусматривают возможность извлечения сердечника замка. Вы можете использовать такой тип замков, если вам нужно открывать одним ключом несколько замков. Для этого вы можете просто установить во все замки одинаковый сердечник.

Кодовые замки (combination lock) требуют ввода правильного сочетания цифр для их открытия. Внутри этих замков есть колесики, которые должны выстроиться в правильную линию, что необходимо для разблокировки замка. Пользователь использует внешний интерфейс замка для правильного выстраивания внутренних колесиков. После того, как все элементы установлены на свои места, все колесики находятся в правильном положении, замок может быть разблокирован. Чем больше колесиков в замке, тем больший уровень защиты он обеспечивает. Современные кодовые замки не используют внутренние колесики, вместо них используется клавиатура, которая более удобна.

Шифро-замки (cipher lock), также называемые программируемыми замками (programmable lock), не имеют ключа и используют клавиатуру для контроля доступа в помещение или здание. Такой замок требует ввода определенной комбинации на клавиатуре или с помощью карточки. Они дороже, чем обычные замки, но их комбинации могут быть изменены, отдельные комбинации могут быть заблокированы, кроме того, можно ввести специальные комбинации, которые позволят персоналу под принуждением ввести соответствующий код, который не только откроет дверь, но и включит сигнал тревоги. Таким образом, по сравнению с традиционными замками, шифро-замки могут обеспечить гораздо более высокий уровень безопасности и контроля доступа в здание.

Ниже перечислены некоторые функциональные возможности, доступные на большинстве шифро-замков, повышающие эффективность контроля доступа и обеспечивающие повышенный уровень безопасности:

- **Дверной таймер (Door delay).** Если дверь была открыта дольше определенного времени, будет подан сигнал для оповещения персонала безопасности о подозрительной деятельности.
- **Замещение ключа (Key override).** Может быть запрограммирована специальная комбинация для использования в чрезвычайных ситуациях с целью обхода обычных процедур или контроля.
- **Мастер-ключ (Master keying).** Позволяет контролирующему персоналу изменить коды доступа и другие функции шифро-замка.
- **Открытие под принуждением (Hostage alarm).** Если человека принуждают открыть дверь, он может ввести специальную комбинацию, что вызовет передачу охране (и, возможно, в полицейский участок) специального сигнала.

Если дверь оборудована шифро-замком, его клавиатура должна быть закрыта от просмотра посторонними набираемых на ней комбинаций. Автоматизированные шифро-замки должны иметь резервную батарею электропитания и должны быть настроены на автоматическое разблокирование в случае отсутствия питания, чтобы персонал не оказался в ловушке внутри помещения в случае чрезвычайной ситуации.

ПРИМЕЧАНИЕ. Важно периодически менять кодовую комбинацию замка и использовать при этом случайные последовательности цифр. Зачастую люди не меняют кодовые комбинации и не чистят клавиатуру, что позволяет злоумышленнику узнать, какие цифры используются в комбинации, т.к. соответствующие им клавиши грязные и изношенные. Злоумышленнику останется только подобрать правильное сочетание этих цифр.

Некоторые шифро-замки требуют, чтобы все пользователи знали и использовали одну и ту же комбинацию, что не обеспечивает какого-либо персонифицированного учета и контроля. Некоторые из наиболее сложных шифро-замков позволяют присвоить персональные коды, уникальные для каждого пользователя. Это дает больше возможностей, так как каждый человек несет ответственность за сохранение в тайне своего личного кода доступа, а также позволяет фиксировать факты входа и выхода каждого пользователя персонально. Эти замки часто называют умными замками (smart lock), поскольку они позволяют разграничивать доступ уполномоченных лиц по определенным дверям и в определенное время.

ПРИМЕЧАНИЕ. В гостиницах часто используются смарт-карты. Они программируются персоналом гостиницы перед их передачей клиенту. С помощью такой карты клиент может войти в свой номер в гостинице, в спорт-зал или мини-бар.

Блокирующие устройства

К сожалению, часто происходят случаи хищения оборудования из помещений компаний. Чтобы помешать этому, необходимы блокирующие устройства (device lock). Например, существуют специальные замки с металлическими кабелями (kensington lock), которые могут обеспечить защиту компьютера, периферийного устройства или другого оборудования.

Ниже приведены некоторые примеры существующих блокирующих устройств:

- **Защита кнопок включения** (switch control). Закрывает кнопки, позволяющие включать и отключать устройство.
- **Защитные слоты** (slot lock). В них закрепляется стальной кабель, другой конец которого прикрепляется к стационарному объекту, что позволяет защитить устройство от хищения.
- **Защита портов** (port control). Блокирует доступ к дисковым устройствам, неиспользуемым последовательным и параллельным портам.
- **Защита включения периферийных устройств** (peripheral switch control). Защищает клавиатуру, путем установки переключателя (включено/выключено) между системным блоком и клавиатурным разъемом.
- **Кабели-ловушки** (cable trap). Предотвращают извлечение устройств ввода/вывода, пропуская их кабели через блокирующее устройство.

Административные обязанности

Важно не только выбрать правильный тип блокирующего устройства, но и обеспечить его правильное использование и сопровождение. Между ответственными за здание (помещения) руководителями следует распределить ключи, оформив это документально. Должны быть разработаны процедуры, подробно описывающие порядок распределения ключей, их инвентаризации, уничтожения (при необходимости), а также порядок действий в случае потери ключей. Среди ответственных за здание (помещения) руководителей компании должна быть выделена роль, на которую возложены обязанности по контролю и сопровождению ключей и комбинаций для шифро-замков.

Большинство компаний имеют мастер-ключи и субмастер-ключи для использования ответственными за здание (помещения) руководителями. Мастер-ключ открывает все замки внутри здания, а субмастер-ключ – один или несколько замков отдельных помещений. Желательно, чтобы каждый замок имел свой собственный уникальный ключ. Поэтому, если в здании 100 офисов, работник каждого офиса может иметь собственный ключ. А мастер-ключ позволяет получить доступ к любому офису для персонала безопасности, а также на случай чрезвычайных ситуаций. Если один охранник отвечает за половину здания, ему может быть выдан субмастер-ключ только от соответствующих офисов.

Мастер- и субмастер-ключи должны надежно охраняться и не использоваться излишне

широким кругом лиц. Политика безопасности должна описывать, какие части здания и какое оборудование должны быть заперты. Как специалист по безопасности, вы должны понимать какие типы замков предпочтительно использовать в каждой конкретной ситуации, какой уровень защиты обеспечивается различными типами замков и как эти замки можно взломать.

Стойкость замков. Существует три основных класса замков:

- **Класс 1.** Коммерческое и промышленное использование
- **Класс 2.** Усиленный для частного использования / облегченный для коммерческого использования
- **Класс 3.** Одноразовый для частного использования / потребительского использования

Цилиндры в замках делятся на три основных категории:

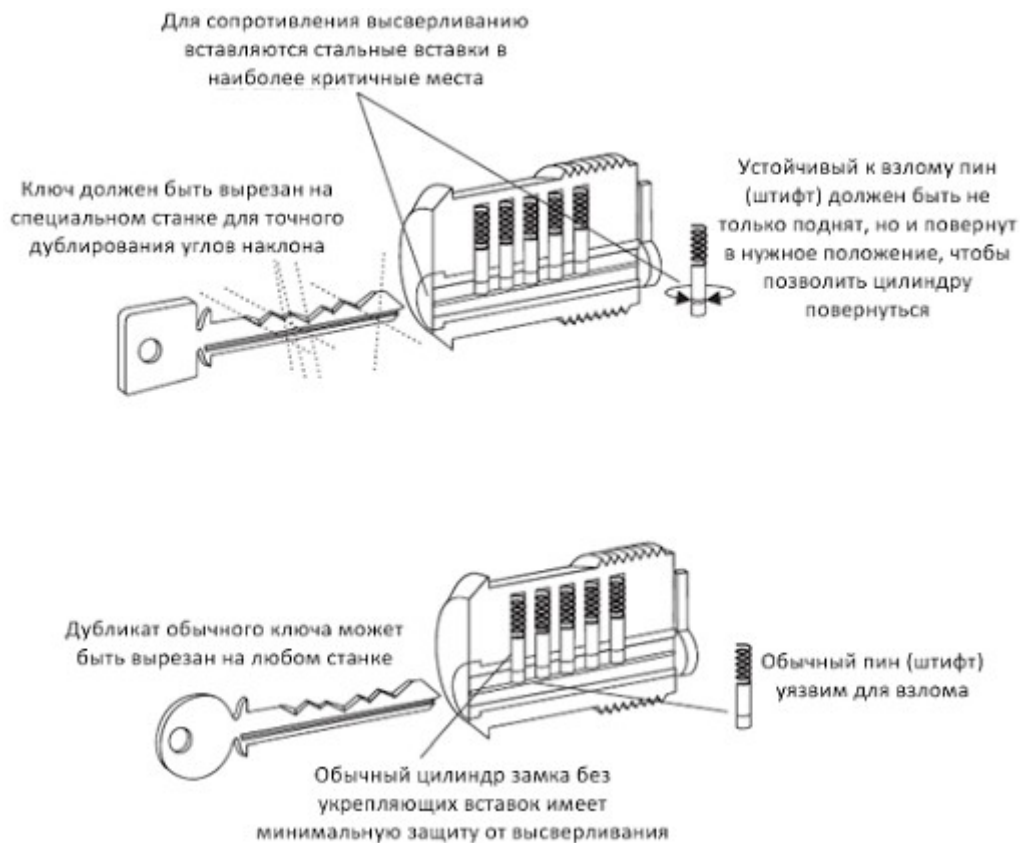
- **Низкая безопасность.** Не обеспечивается защита от взлома или высверливания (может относиться к любому из трех классов замков)
- **Средняя безопасность.** Обеспечивается определенная степень сопротивления взлому (используется более плотная и сложная замочная скважина (комбинация пазов и выступов); может относиться к любому из трех классов замков)
- **Высокая безопасность.** Сопротивляемость взлому обеспечивается несколькими различными механизмами (используется только в замках классов 1 и 2)

Взлом замков

Для каждого типа замков существуют специальные средства, которые могут быть использованы для взлома (открывания без ключа). Для вращения движущейся части цилиндрического замка используется средство для поворота (tension wrench), похожее на букву «L». Взломщик использует отмычки для установки отдельных штифтов (пинов) замка в нужное положение. Когда некоторые штифты уже поставлены на правильное место, поворот цилиндра придерживает их, пока взломщик пытается поставить на место остальные штифты. После того, как все штифты поставлены на правильное место, средство поворота используется для открывания замка.

Взломщик может использовать другую технику, оказывающую воздействие на несколько штифтов (raking). Для этого используется отмычка с закругленным концом. Когда отмычка проходит вдоль штифтов, она поднимает и опускает их и в момент, когда поверхность одного из наиболее плотно сидящих штифтов совмещается с линией внутренней поверхности корпуса, цилиндр слегка поворачивается и т. д.

Также, взломщик может воспользоваться техникой бампинга (lock bumping). Для этого используется специальный, изготовленный из твердого материала (тверже материала замка) бамп-ключ, в котором все вырезы сделаны на максимальную глубину. Этот ключ вставляется на полную глубину замочной скважины, затем вытягивается на один штифт, после чего нужно ударить по тыльной стороне ключа. Импульс от ключа передается штифтам, которые придут в движение, и уже через долю секунды нужно попытаться повернуть ключ и открыть замок.



А если все это злоумышленнику не удалось, он может просто высверлить замок, используя специальное сверло, попытаться взломать дверь или удалить петли. У плохих парней есть большой выбор.

5.2. Контроль доступа персонала

Чтобы проверить, имеет ли право доступа в здание или помещение пытающийся войти в него человек, необходима надежная идентификация. Идентификация и аутентификация могут производиться на основании анатомических атрибутов (биометрические системы), смарт-карт, магнитных (swipe) карт, с помощью предоставления сотруднику охраны документа, содержащего фотографию, используя ключи, либо карты, защищенные паролем или пин-кодом.

Наиболее частой проблемой при контроле прав доступа в здание или помещение является проход неуполномоченного человека по реквизитам уполномоченного. Обычно это происходит, когда человек просто следует за другим и проходит через дверь не предъявляя своих реквизитов или прав доступа (piggybacking). Самой лучшей защитной мерой против этого является установка охранника в точках доступа и обучение сотрудников требованиям безопасности.

Если компания хочет использовать считыватели для карт-бейджей, она может выбирать из нескольких различных типов таких систем. Часто используются магнитные карты, которые содержат информацию для доступа. Считыватель должен просто «увидеть» наличие этой информации, либо он может быть подключен к более сложной системе, которая после считывания информации принимает более сложные решения о доступе, а также ведет учет использования бейджей по их идентификаторам и фиксирует время прохода.

Если используются карты памяти (магнитные карты), то считыватель просто загружает информацию с карты и принимает решение о доступе. Если используются смарт-карты, то человеку необходимо ввести пин-код или пароль, которые считыватель должен сравнить с информацией, хранящейся на самой карте или на сервере аутентификации. (Карты памяти и смарт-карты рассматривались в Домене 02).

Эти карты могут быть использованы со считывателями, активируемыми пользователем (т.е. пользователь должен самостоятельно выполнить какое-либо действие – провести карту по считывателю или ввести пин-код). Считыватели систем контроля доступа, которые называют proximity-устройствами или ретрансляторами (transponder), обнаруживают приближение объекта к определенным областям. Такая система не требует от пользователя проводить карту по считывателю. Считыватель посылает сигнал запроса и получает код доступа с карты пользователя без необходимости пользователю делать что-либо.

ПРИМЕЧАНИЕ. Токены электронного контроля доступа (ЕАС – Electronic access control) – это общий термин, который используется для описания аутентификационных proximity-устройств, которые могут быть бесконтактными считывателями, шифро-замками или биометрическими системами, проводящими идентификацию и аутентификацию пользователей, прежде чем позволить им войти в контролируемые помещения.

5.3. Механизмы защиты внешних границ

Компоненты защиты периметра применяются, как правило, для предоставления одного или нескольких из следующих сервисов:

- Управление потоком пешеходов и автомобилей
- Различные уровни защиты для разных зон безопасности
- Буферы и механизмы задержки для защиты от попыток входа под принуждением
- Ограничения и контроль точек входа

Эти сервисы могут реализовываться с помощью следующих механизмов:

- **Механизмы контроля доступа.** Замки и ключи, электронная карточная система доступа, повышение осведомленности персонала
- **Физические барьеры.** Ограждения, ворота, стены, двери, окна, защита вентиляции, барьеры для транспорта
- **Выявление вторжений.** Датчики периметра, внутренние датчики, механизмы оповещения
- **Оценка.** Охранники, камеры видеонаблюдения
- **Реакция.** Охранники, местные правоохранительные органы
- **Сдерживание (устрашение).** Знаки, освещение, дизайн окружения

Компании могут использовать различные типы механизмов защиты и контроля периметра для защиты помещений, активов и персонала. Они могут сдерживать потенциальных нарушителей, выявлять вторжения и необычную деятельность, предоставлять возможности решения возникающих проблем. Периметр контроля безопасности может быть естественным (горы, реки) и искусственным (ограждения, освещение, ворота). Проектирование ландшафта (landscaping) – это использование сочетания обоих этих элементов. В начале этого Домена мы изучили подход CPTED и его использование для снижения вероятности преступлений. Проектирование ландшафта – это инструмент, используемый в подходе CPTED. Например, тротуары и пешеходные дорожки могут указывать людям правильные точки входа, а деревья и остроконечный кустарник можно использовать в качестве естественных препятствий. Выбирать эти кусты и деревья, а также размещать их, следует таким образом, чтобы они не могли быть использованы в качестве лестницы или средства для получения несанкционированного доступа к точкам входа. Кроме того, количество деревьев и кустарника не должно быть чрезмерным – они не должны позволять нарушителям скрыться. В следующих разделах мы рассмотрим искусственные компоненты, которые могут использоваться при проектировании ландшафта.

Ограждения

Ограждение может быть достаточно эффективным физическим барьером. Несмотря на то, что наличие ограждения может быть лишь задержкой для злоумышленника, пытающегося получить доступ, оно может работать в качестве сдерживающего психологического фактора, говорящего о том, что ваша компания серьезно относится к собственной безопасности.

Установка ограждений может управлять толпой или доступом к входам в здание. Однако ограждение может быть дорогостоящим и смотрящимся некрасиво. Для эстетики и чтобы сделать здание менее заметным, многие компании сажают кусты или деревья перед ограждениями, окружающими их здание. Но такая растительность может со временем повредить ограждение или нарушить его целостность. Ограждение должно надлежащим образом содержаться, поскольку провисший, проржавевший и покосившийся забор будет говорить всем о том, что компания не является дисциплинированной и вряд ли серьезно относится к безопасности. Но приятное, блестящее, надежное ограждение говорит об обратном. Особенно, если сверху оно украшено тремя рядами колючей проволоки.

При выборе типа ограждения, нужно учесть несколько факторов. Тип металла должен соответствовать видам физических угроз, которым компания, вероятно, будет подвержена. После проведения анализа рисков (описанного в начале этого Домена), группа физической безопасности должна понимать, какие вероятные враги будут пытаться сломать ограждение, перелезть через него или пролезть под ним. Понимание этих угроз поможет группе определить необходимый материал и размер ячейки прутьев ограждения.

Данные анализа рисков также помогут определить необходимую компании высоту ограждения, поскольку различная высота соответствует разным уровням защищенности:

- Ограждение от трех до четырех футов (0,9 – 1,2 метра) высотой может сдерживать только случайных нарушителей.
- Ограждение от шести до семи футов (1,8 – 2,2 метра) высотой достаточно высоко, чтобы нельзя было легко перелезть через него.
- Ограждение восьми футов (2,5 метра) высотой (возможно, с колючей проволокой наверху) говорит, что вы серьезно относитесь к охране своего имущества. Такое ограждение может сдерживать даже решительного злоумышленника.

Колючая проволока на верхней части ограждения может быть наклонена вперед или назад, что также обеспечивает дополнительную защиту. Например, наклоненная внутрь колючая проволока наверху ограждения тюрьмы затруднит возможность побега для заключенных. Наклоненная наружу колючая проволока военной базы затруднит возможность перелезть через ограждение снаружи и получить доступ к зданию.

Наиболее критичные зоны должны иметь ограждение, высотой не менее восьми футов (2,5 метра), чтобы обеспечить надлежащий уровень защиты. Ограждения не должны нигде прогибаться, должны быть в хорошем состоянии и надежно закреплены на столбах. Не должно быть возможностей легко преодолеть ограждение, вытянув из земли столб, к которому оно крепится. Эти столбы должны быть достаточно глубоко вкопаны в землю, надежно забетонированы, чтобы их нельзя было выкопать или вытащить, привязав к автомобилю. Если грунт мягкий или неравномерный, это может позволить злоумышленнику сделать подкоп под ограждением. В таком случае, ограждение само должно быть вкопано в землю, чтобы помешать такому виду атак.

Ограждения являются механизмами «первой линией обороны». Наряду с ними могут использоваться и другие защитные меры. Ворота должны быть прочными и надежными. Не имеет смысла устанавливать надежное и дорогое ограждение, но при этом не иметь надежных ворот или не закрывать их.

Калибры и размеры ячеек. Калибр прутьев ограждения – это толщина прутьев, из которых состоит сетка ограждения. Чем ниже номер калибра, тем больше диаметр прутьев:

- 11 калибр = диаметр 0,0907 дюймов (2,3 мм)
- 9 калибр = диаметр 0,1144 дюймов (2,9 мм)
- 6 калибр = диаметр 0,162 дюймов (4,1 мм)

Размер ячейки в сетке – это минимальное пустое расстояние между прутьями. Обычно размер ячейки составляет 2 дюйма (50,8 мм), 1 дюйм (25,4 мм) или 3/8 дюйма (9,5 мм). Сложнее перелезть или перерезать ограждение с маленьким размером ячейки и толстыми прутьями. Следующий список показывает уровень защищенности наиболее часто используемых калибров и размеров ячеек в современных ограждениях:

- Экстремально высокая безопасность. 3/8 дюйма, 11 калибр
- Очень высокая безопасность. 1 дюйм, 9 калибр
- Высокая безопасность. 1 дюйм, 11 калибр
- Повышенная безопасность. 2 дюйма, 6 калибр
- Нормальная промышленная безопасность. 2 дюйма, 9 калибр

Ограждения PIDAS. Система обнаружения и оценки вторжений в периметр (PIDAS – Perimeter Intrusion Detection and Assessment System) – это тип ограждений, оборудованных датчиками на прутьях сетки и в основании ограждения. Эти датчики используются для выявления попыток распиливать ограждение или перелезть через него. В качестве датчиков применяются пассивные кабельные датчики вибрации, которые подают сигнал в случае выявления вторжения. PIDAS очень чувствительны и могут вызвать много ложных срабатываний.

Существует четыре основных класса ворот:

- **Класс I** – для частного использования
- **Класс II** – для коммерческого использования, если в основном предполагается публичный доступ; пример – ворота общедоступной парковки.
- **Класс III** – промышленное использование, если предполагается ограниченный доступ; пример – вход на товарный склад.
- **Класс IV** – ограниченный доступ; например, вход в тюрьму, где отслеживается каждый человек.

Каждый класс ворот имеет свой обширный набор руководств по применению и эксплуатации для обеспечения необходимого уровня защиты. Эти классы и описания разработаны Лабораторией по технике безопасности США (UL – Underwriters Laboratory) – некоммерческой организацией, которая тестирует, обследует и классифицирует электронные устройства, противопожарное оборудование, специальные конструкционные материалы. Она проводит сертификацию этих различных элементов, обеспечивая их соответствие национальным строительным кодексам. Например, их стандарт UL-325 относится к механизмам и системам гаражных дверей, занавесов, ворот, жалюзей и окон. Так, если в информационной безопасности за лучшими практиками обращаются к NIST, то в физической безопасности – к UL.

Столбики ограждения

Столбики ограждения (bollards) обычно выглядят как маленькие бетонные столбики (pillars) вокруг здания. Иногда компании стараются их украсить с помощью цветов или подсветки. Они размещаются вокруг стен здания и предотвращают угрозы столкновения со зданием автомобилей. Обычно их располагают между зданием и парковкой, а также между зданием и дорогой, что позволяет прикрыть внешние стены. В некоторых случаях вместо столбиков для защиты важных зданий используются массивные камни, которые могут защитить от тех же видов угроз.

Освещение

Многие из вещей, упомянутых ранее в этом Домене, считаются сами собой разумеющимися в повседневной жизни. Освещение – это, конечно же, одна из таких вещей. Неосвещенная (или плохо освещенная) парковка является приманкой для злоумышленников в отличие от хорошо освещенных мест. Плохое освещение на парковке может стать причиной повреждения автомобилей, их угона, нападения на сотрудников, задержавшихся на работе, а также других угроз. Специалисты по безопасности должны понимать, что необходимо обеспечить хорошее освещение, не имеющее мертвых зон (неосвещенных участков) между источниками света. Места, где могут прогуливаться люди должны быть хорошо освещены. Специалисты по безопасности должны также понимать возможности и различия между доступными типами освещения.

Даже если используется осветительный блок (array of lights), каждый источник света освещает свою зону или область. Зона покрытия каждого источника света зависит от его мощности, которая обычно напрямую связана с мощностью используемых ламп. В большинстве случаев, чем больше мощность ламп, тем больше освещенность. Также важно обеспечить перекрытие зон покрытия. Например, компания имеет открытую парковку, которую она хочет осветить. Она должна расположить осветительные столбы на нужном расстоянии друг от друга, чтобы обеспечить отсутствие мертвых зон. Если используемая лампа дает 30-футовый (9,3 метра) радиус освещенного пространства при расположении на высоте 30 футов над землей, значит именно на расстоянии 30 футов и нужно располагать столбы.

Если компания выбрала неправильный тип освещения или не обеспечила надлежащее покрытие, это увеличивает риск криминальных действий, инцидентов и нарушений закона.

Внешнее освещение обычно не требует большой интенсивности света, в отличие от освещения внутренней территории. Исключением являются области, необходимые персоналу безопасности для проверки пропусков на входе. Также важно обеспечить хорошее освещение при использовании различных типов систем наблюдения. Должен быть обеспечен высокий контраст между потенциальным нарушителем и элементом фона, что можно обеспечить только хорошим освещением местности. Если фонарь находится на границе освещенной зоны, освещает грязь или поверхность, окрашенную в темный цвет, требуется существенно большая мощность освещения, чтобы обеспечить необходимый контраст между людьми и окружением. Если наоборот, поверхность является светлой (например, чистый бетон), такой большой мощности освещения уже не требуется. Это связано с тем, что при падении одинакового количества света на объект и окружающие фоновые предметы, наблюдатель необходим контраст между ними, чтобы их увидеть.

После установки освещения, оно должно быть направлено на те области, из которых наиболее вероятны действия потенциальных нарушителей. Часто это области, удаленные от постов охраны. Например, освещение должно быть установлено на воротах или внешних точках доступа, а охранник, наоборот, в большей степени должен находиться в тени, чем под освещением. Это называется **защитой от ослепления** (glare protection) сил безопасности. В военных организациях на точках входа размещаются укрепленные здания охраны с прожекторами, направленными на подъезжающие автомобили. Большой знак инструктирует вас выключить фары, чтобы охранники не были ослеплены ими и имели хороший обзор всего происходящего на дороге.

Освещение, установленное по периметру безопасности компании, должно быть направлено наружу, что обеспечит безопасность персонала в условиях относительной темноты, и позволит легко наблюдать за злоумышленниками за пределами периметра компании.

Прожектора, обеспечивающие равномерное освещение всей площади, как правило, называют **непрерывным освещением** (continuous lighting). Примерами являются равномерно освещенные автостоянки, фонарные столбы, установленные по всему наружному периметру здания, или массивы флуоресцентных ламп на парковках. Если здание компании находится в

непосредственной близости от собственности другой компании, железной дороги, аэропорта или шоссе, может потребоваться дополнительно обеспечить, чтобы освещение компании не выходило за ее территорию. Поэтому освещение должно быть **контролируемым**, компания должна установить освещение таким образом, чтобы не слепить своих соседей, проезжающие автомобили, поезда или самолеты.

Вы, возможно, уже знакомы со специальными домашними устройствами освещения, которые при отсутствии хозяев по заранее заданной программе включают и отключают освещение в доме, создавая иллюзию для потенциальных взломщиков, что в доме кто-то есть. Компании могут использовать аналогичные технологии, которые называют **дежурным освещением** (standby lighting). То же самое можно делать в помещениях, в которых работает персонал – сотрудники безопасности могут настроить время, в которое освещение включается и выключается, чтобы потенциальные нарушители думали, что там кто-то есть.

ПРИМЕЧАНИЕ. Дополнительное или дежурное освещение должно быть доступно в случае проблем с электроснабжением или чрезвычайных ситуаций. Нужно внимательно подобрать правильные типы освещения для различных частей здания и для различных ситуаций. Освещение может работать от генераторов или аккумуляторных батарей.

При выявлении подозрительной активности системой IDS может включаться **активная система освещения** (responsive area illumination) соответствующей области. Эта система должна повернуть прожекторы, направив их на место, где была выявлена эта подозрительная активность. Когда такие технологии подключены к автоматизированным системам IDS, возникает высокая вероятность ложной тревоги. Вместо того чтобы постоянно посылать охранников разбираться с причинами тревоги на месте, может быть внедрена система видеонаблюдения для просмотра соответствующей области с целью выявления злоумышленников.

Если злоумышленник хочет напасть на сотрудников охраны или избежать своего обнаружения при попытке войти в здание компании, он может попытаться отключить свет или перерезать электрические провода. Поэтому выключатели освещения и системы управления освещением должны быть защищены, заперты и расположены централизованно.

Устройства наблюдения

Обычно установка одних только ограждений и освещения не обеспечивает необходимого уровня защиты здания, оборудования и персонала компании. Помещения должны находиться под наблюдением для выявления нежелательной деятельности и заблаговременного принятия превентивных мер, не позволяющих проблеме реально произойти. Наблюдение может осуществляться визуальным путем или с помощью устройств, использующих сложные методы выявления необычного поведения и нежелательных условий. Для любой компании важно применять правильный набор освещения, персонала безопасности, систем IDS, а также технологий и методов наблюдения.

Устройства видеозаписи

Поскольку наблюдение основано на чувственном восприятии, устройства наблюдения обычно работают совместно с персоналом охраны и другими механизмами мониторинга, чтобы расширить их возможности и диапазон восприятия. Наиболее часто используемым компаниями устройством мониторинга является видеонаблюдение (CCTV – closed-circuit TV), однако перед его покупкой и внедрением нужно учесть следующие аспекты:

- **Цель системы видеонаблюдения:** выявление, оценка и/или идентификация нарушителей
- **Тип окружения, в котором будут работать камеры видеонаблюдения:** внутренние помещения или внешние области
- **Требуемая область обзора:** широкую или узкую область необходимо контролировать

- **Величина освещенности окружения:** светлые области, темные области, области, находящиеся под солнечными лучами
- **Интеграция с другими механизмами контроля безопасности:** охрана, IDS, системы сигнализации

Существует множество различных типов камер, объективов, мониторов, из которых состоят различные продукты видеонаблюдения, поэтому продумать указанные выше вопросы нужно заранее, перед покупкой системы видеонаблюдения. Вы должны понимать, что нужно именно вашей системе контроля физической безопасности, чтобы купить и внедрить систему правильного типа.

Система видеонаблюдения состоит из камер, передатчиков, приемников, записывающей системы и монитора. Камера снимает изображение и передает его приемнику, который обеспечивает отображение полученного изображения на экране монитора. Это изображение может записываться, что позволит просмотреть его спустя некоторое время, если это необходимо. На рисунке 4-15 показано, как несколько камер могут быть подключены к одному мультиплексору, который позволяет одновременно контролировать несколько различных областей. Мультиплексор принимает видеосигнал со всех камер и выстраивает изображения с них в линию на центральном мониторе. Это более эффективно, чем в старых системах, которые требовали от охранников постоянно переключаться с одной камеры на другую. В этих старых системах охранник мог наблюдать только за изображением с одной камеры, что существенно повышало вероятность того, что подозрительные действия не будут замечены.

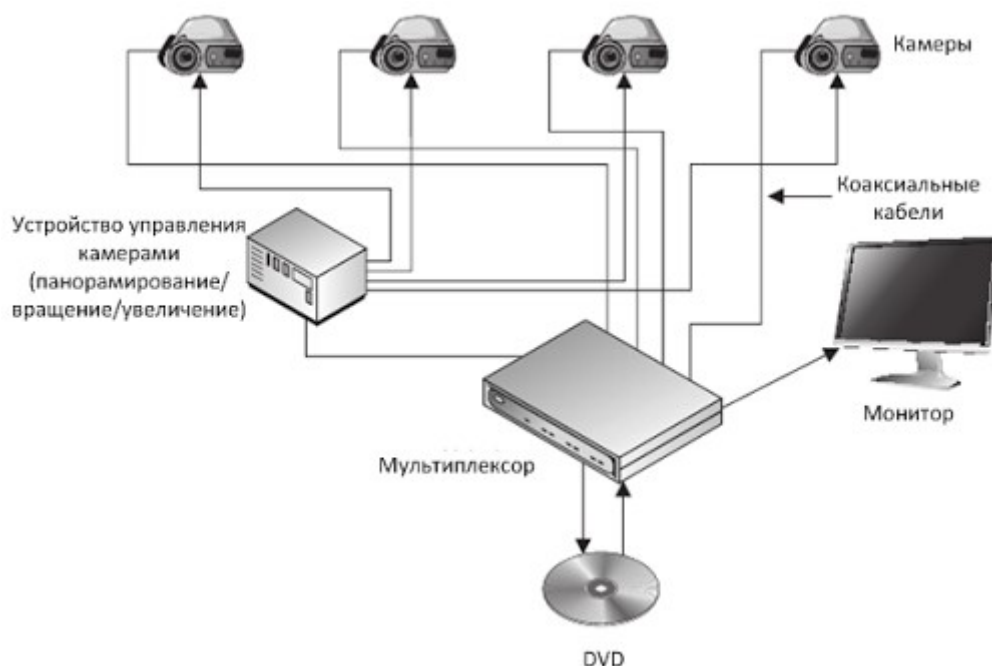


Рисунок 4-15. Несколько камер могут быть подключены к мультиплексору

Система видеонаблюдения обычно отправляет изображение с передатчиков камер в приемники мониторов посредством коаксиального кабеля, а не по общедоступным сетям. Поэтому такие системы называют замкнутыми. Эти кабели должны быть устойчивы к внешним воздействиям, чтобы злоумышленники не могли управлять видеоизображением на мониторах охраны. Чаще всего используется вид атаки, при котором незаметно реализуется повтор ранее записанного изображения на мониторах охраны. Например, если атакующий сможет взломать систему видеонаблюдения компании и воспроизвести запись, записанную днем ранее, охранники могут не узнать о присутствии злоумышленника в здании и выполнении им преступной деятельности. Это еще одна причина, по которой системы

видеонаблюдения следует использовать совместно с системой IDS, которая описывается в следующем разделе.

ПРИМЕЧАНИЕ. Для системы видеонаблюдения следует организовать видеозапись. Цифровые системы видеозаписи сохраняют изображение на жесткие диски и позволяют использовать технологии поиска по сложным критериям, недоступным в случае записи изображения на видеокассеты. Цифровые системы видеозаписи используют эффективные технологии сжатия, которые значительно уменьшают требования к объему носителей информации.

В большинстве камер видеонаблюдения, использующихся в настоящее время, применяются светочувствительные микросхемы, называемые устройствами с зарядовой связью (CCD – charged-coupled device). CCD – это электрическая схема, которая получает через объектив внешний свет и преобразует его в электрические сигналы, формирующие изображение на мониторе. Картинка с помощью линз фокусируется на поверхности CCD-микросхемы, которая формирует электрическое представление оптического изображения. Эта технология обеспечивает очень высокий уровень детализации объектов и точности отображения, т.к. сенсоры работают в инфракрасном диапазоне, расширяющем возможности человеческого восприятия. Сенсор CCD улавливает дополнительные «данные» и добавляет их в изображение на экране монитора, обеспечивая повышенную детальность и качество видеоизображения.

CCD также используются в факсах, копировальных машинах, считывателях штрих-кодов и даже в телескопах. Системы видеонаблюдения, использующие CCD, обеспечивают отображение на мониторе более детализированной информации по сравнению с устаревшими технологиями видеонаблюдения.

Существует два основных типа объективов, используемых в системах видеонаблюдения: с фиксированным фокусным расстоянием и переменным (zoom-объективы). **Фокусное расстояние** (focal length) объектива определяет его эффективность при просмотре объектов по горизонтали и вертикали. Возможные углы обзора связаны со значением фокусного расстояния. Короткое фокусное расстояние объектива обеспечивает широкий угол обзора, а длинное – более узкий. Размер изображения, которое будет показано на мониторе, вместе с площадью, охватываемой одной камерой, определяются фокусным расстоянием. Например, если компания устанавливает камеры видеонаблюдения на складе, то значение фокусного расстояния объектива должно быть от 2,8 до 4,3 мм, чтобы камера смогла охватить всю площадь. Если компания устанавливает камеру видеонаблюдения, которая будет контролировать только один вход, значение фокусного расстояния объектива должно быть около 8 мм, что позволит контролировать меньшую площадь.

ПРИМЕЧАНИЕ. Существуют объективы с фиксированным фокусным расстоянием для различных углов обзора: широкого, среднего и узкого. Объектив, который обеспечивает «нормальное» фокусное расстояние, создает картину, примерно соответствующую полю зрения человеческого глаза. Широкоугольные объективы имеют короткое фокусное расстояние, а телеобъективы – длинное. Если компания выбирает объективы с фиксированным фокусным расстоянием для обзора части окружения, она должна понимать, что в случае необходимости изменить угол обзора (с широкого на узкий), ей придется менять объективы.

Таким образом, если нам нужно наблюдать за большой территорией, следует использовать объектив с меньшим значением фокусного расстояния. Прекрасно, но что делать, если охранник слышит шум или видит нечто подозрительное? Объектив с фиксированным фокусным расстоянием неподвижен, соответственно охранник не может направлять камеру из одной точки в другую и автоматически фокусировать объектив. **Zoom-объективы** обеспечивают дополнительную гибкость, позволяя наблюдателю изменять поле зрения на нужный угол и расстояние. Персонал безопасности, как правило, имеет средства дистанционного управления, интегрированные в централизованную систему видеонаблюдения, позволяющие перемещать камеры, увеличивать и уменьшать изображение объектов в случае необходимости. Когда нужен как широкий угол, так и съемки крупным

планом, лучше использовать зум-объективы. Этот тип объективов позволяет изменять фокусное расстояние, переходя от широкоугольного режима до режима телефото, сохраняя правильную фокусировку изображения.

Чтобы понять следующую характеристику, глубину резкости (depth of field), представьте себе, что вы делаете фотографии во время отпуска с семьей. Например, вы хотите сделать фотографию вашей супруги на фоне с Гранд-Каньоном, главный объект изображения – ваша супруга. Ваша камера приближает изображение и использует *небольшую глубину резкости*. Это обеспечит мягкий фон, который позволит зрителю сконцентрироваться при просмотре фотографии на переднем плане, на котором будет ваша супруга. Теперь, допустим, что вы устали от съемки вашей супруги и хотите получить живописные картины самого Гранд-Каньона. При этом камера будет использовать *большую глубину резкости*, чтобы не было таких различий между объектами на переднем и заднем плане.

Необходимо понимать, что такое глубина резкости, чтобы выбрать правильные объективы и правильно их настроить в системе видеонаблюдения вашей компании. **Глубина резкости** определяет ту часть изображения, которая находится в фокусе (т.е. изображена резко) при отображении на экране монитора. Глубина резкости зависит от степени открытия диафрагмы, расстояния до объекта и фокусного расстояния объектива. Глубина резкости возрастает при уменьшении диафрагмы, увеличении расстояния до объекта или уменьшении фокусного расстояния объектива. Итак, если вы хотите охватить большую площадь, а не сосредоточивать внимание на конкретных деталях, лучше использовать широкоугольный объектив и небольшую диафрагму.

Объективы камер систем видеонаблюдения имеют **диафрагму** (iris), которая контролирует количество света, попадающего в объектив. Для ручной настройки диафрагмы на объективе камеры имеется специальное кольцо. Объективы, имеющие ручное управление диафрагмой, следует использовать в тех местах, в которых обеспечен постоянный уровень освещенности, поскольку диафрагма не может автоматически адаптироваться к изменениям количества света. **Объективы с автоматической регулировкой диафрагмы** (auto iris lens) могут применяться в любых условиях, т.к. они автоматически подстраиваются под изменения в уровне освещенности. При увеличении количества света диафрагма автоматически подстраивается. Персонал безопасности настраивает камеры видеонаблюдения на конкретные фиксированные значения экспозиции, которую обеспечивает диафрагма. В солнечный день диафрагма объектива закрывается для уменьшения количества света, проходящего в камеру, а в ночное время диафрагма открывается, чтобы захватить больше света – точно так же, как и наши глаза.

При выборе правильной системы видеонаблюдения для имеющегося окружения, вы должны определить, количество света в этом окружении. Различные камеры и объективы имеют различные требования к освещению для обеспечения наилучшего качества изображения. Требования по уровню освещенности, как правило, указываются в «люксах», являющихся мерой освещенности (также освещенность иногда измеряют в «фут-свечах» (foot-candle) – 1 фут-свеча соответствует 10,76 люксов). Величину освещенности нельзя точно указать на лампочке, поскольку существует множество факторов, непосредственно влияющих на освещенность. Именно поэтому величину освещенности целесообразно измерять в конкретном месте, где работает источник света.

Далее, нужно учесть требования по монтажу камер видеонаблюдения. Может использоваться как вариант **жесткого монтажа** камеры, так и вариант, при котором камера может перемещаться при необходимости (PTZ). Жестко закрепленная камера не может двигаться по командам персонала безопасности, тогда как камеру PTZ можно при необходимости поворачивать, наклонять или изменять масштаб.

Таким образом, покупка и внедрение системы видеонаблюдения может быть не так проста, как кажется. Как специалисту по безопасности, вам необходимо понимать назначение систем

видеонаблюдения, окружение, которое будет контролироваться, а также какие задачи потребуются выполнять сотрудникам службы безопасности, ежедневно использующим систему видеонаблюдения. Различные компоненты, составляющие систему видеонаблюдения, показаны на Рисунке 4-16.

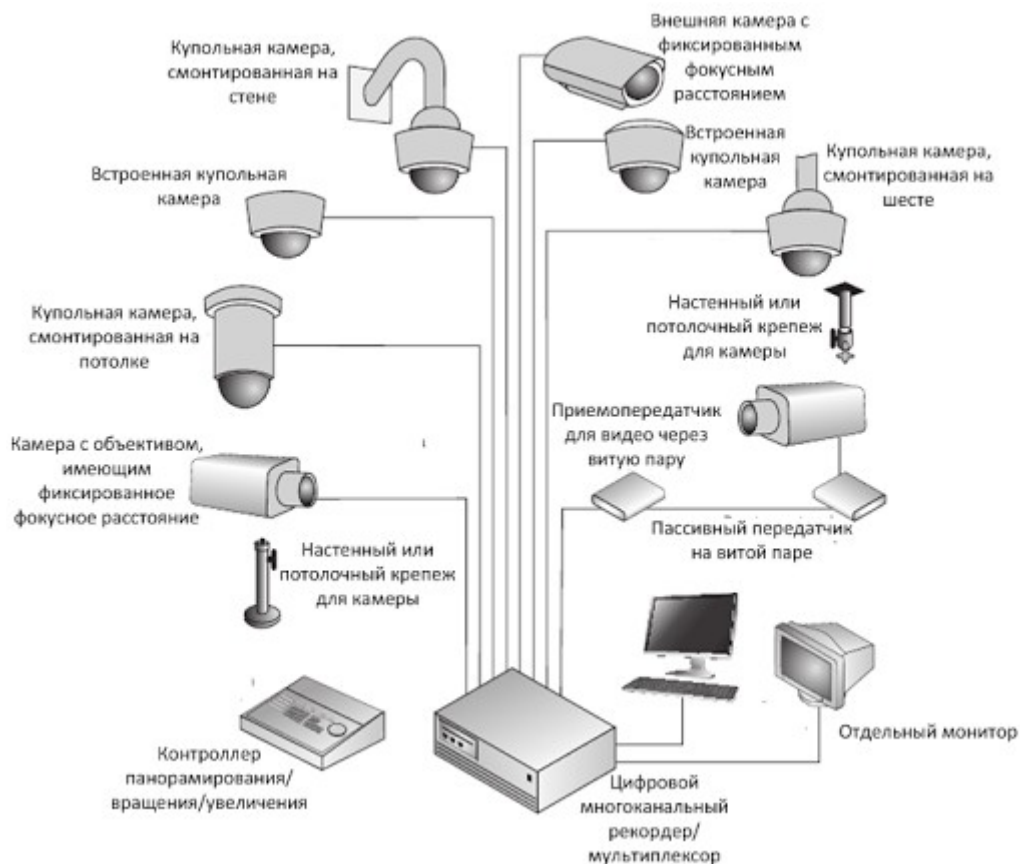


Рисунок 4-16. Система видеонаблюдения может состоять из множества компонентов

Прекрасно, ваша группа оценки провела все свои научные исследования, закупила и внедрила правильную систему видеонаблюдения; теперь было бы неплохо, чтобы кто-то реально следил за мониторами с целью выявления подозрительной активности. Нужно понимать, что непрерывное наблюдение за мониторами психически убивает человека, поэтому имеет смысл внедрить какую-либо систему оповещения. Доступны различные виды систем оповещения, которые могут «слушать» шум или выявлять движение и активировать при необходимости электрические устройства, такие как освещение, сирена, камеры видеонаблюдения. Вместо того чтобы смотреть на монитор видеонаблюдения в течение восьми часов, охранник может осуществлять другие полезные виды деятельности и получать предупреждение, например, при обнаружении движения на экране.

5.4. Системы выявления вторжений

Методы наблюдения используются для выявления нежелательного поведения, тогда как устройства выявления вторжений контролируют изменения, происходящие в окружении. Оба метода являются методами мониторинга, но они используют различное оборудование и подходы. В этом разделе рассматриваются технологии, использующиеся для выявления присутствия нарушителя. Пример устройства сканирования периметра показан на рисунке 4-17.

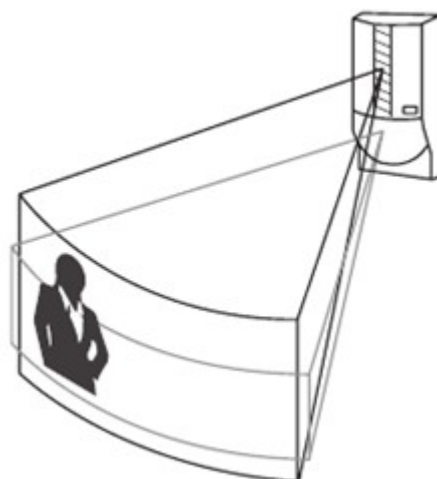


Рисунок 4-17. Различные устройства сканирования периметра покрывают определенные области

IDS используются для выявления несанкционированных действий и уведомления ответственных лиц о необходимости принятия мер. Эти системы могут контролировать входы, двери, окна или съемные чехлы оборудования. Многие из них используют магнитные контакты или чувствительные к вибрации датчики, которые позволяют выявить множество видов происходящих в окружении изменений. При выявлении изменения устройство IDS включает звуковую тревогу в помещении охраны (либо, дополнительно, отправляет сигнал в полицейский участок).

IDS может использоваться для выявления следующих изменений:

- Лучи света
- Звук или вибрация
- Движение
- Различные виды полей (микроволновые, ультразвуковые, электростатические)
- Электричество

IDS могут использоваться для обнаружения вторжений с помощью электро-механических систем (магнитные переключатели, металлическая фольга на окнах) или объемных систем. Объемные системы более чувствительны, поскольку обнаруживают изменения в тончайших характеристиках окружения, таких как вибрация, микроволны, ультразвуковые частоты, инфракрасные лучи, а также фотоэлектрические изменения.

Электронно-механические системы выявляют изменения или разрывы в цепи.

Электрические цепи могут быть встроены в полоски фольги или подсоединены к окнам. Если окно разбивается, фольга разрушается и звучит сигнал тревоги. Детекторы вибрации могут обнаруживать движение на стенах, экранах, потолках и полах, а также разрывы проводов, проложенных в этих структурах. Выключатели с магнитными контактами могут быть установлены на окна и двери. Если контакты разделяются в результате открытия двери или окна, звучит сигнал тревоги.

Другим типом электро-механических датчиков является панель давления. Ее размещают под ковром или полом и автоматически активируют, например, через час после того, как персонал покидает помещение. Если кто-то после этого наступит на эту панель, зазвучит сигнал тревоги, потому что никто не должен находиться в этом помещении в это время. Разновидностями объемных IDS являются фотоэлектрические, акустико-сейсмические, ультразвуковые и микроволновые.

Фотоэлектрические (или **фотометрические**) системы выявляют изменение светового

потока и, следовательно, могут быть использованы только в комнатах без окон. Эти системы работают, как фотоэлектрические датчики дыма, которые испускают луч, попадающий в приемник. Если этот луч света прерывается, звучит сигнал тревоги. Испускаемые фотоэлементами лучи могут быть кросс-секционными, а также видимыми или невидимыми. Кросс-секционность означает, что каждая область может иметь вокруг себя несколько различных световых лучей, которые отражаются от скрытых зеркал, пока не попадают в приемник. Такие системы часто показывают в кино. Вы, наверное, помните Джеймса Бонда, который с помощью прибора ночного видения просматривает невидимые лучи, а затем проходит через них.

Пассивные инфракрасные системы выявляют изменения тепловых волн в контролируемом помещении. Если в некоторой части температура воздуха повышается, это может быть свидетельством присутствия нарушителя, поэтому звучит сигнал тревоги.

Системы акустического контроля используют микрофоны, установленные на стенах, полу, потолке. Их целью является выявление любых звуков, которые могут быть следствием силового проникновения. Эти системы легко установить, однако они очень чувствительны и не могут использоваться в помещениях, в которых слышен ветер или шум потока машин.

Датчики вибрации очень похожи на акустические, они также применяются для выявления силовых проникновений. Финансовые компании могут выбрать этот тип датчиков для установки на внешние стены, на которые банковские грабители могут попробовать направить автомобили, чтобы пробить их. Они также часто используются на полу и потолке хранилищ для выявления попыток проникновения через них.

Волновые детекторы движения различают частоты волн, которые они контролируют. Эти различные частоты могут быть микроволнами, ультразвуком, низкими частотами. Все эти устройства генерируют специальную сетку волн (wave pattern), которые пропускаются через контролируемую зону и отражаются от приемника. Если вернувшаяся сетка не нарушена – все нормально. Если сетка вернулась искаженной – это может быть связано с чьим-то перемещением в помещении, поэтому включается сигнал тревоги.

Датчики приближения, или **емкостные датчики**, являются источником магнитного поля. Они контролируют это поле и включают сигнал тревоги, если оно нарушено. Такие устройства обычно используются при защите специфичных объектов (произведений искусства, кабинетов, сейфов), а не для защиты всего помещения или большой области. Чтобы поймать нарушителя, используется контроль изменение емкости электростатического поля. Но давайте сначала разберемся, что означает изменение емкости. Электростатические IDS создают электростатическое магнитное поле, которое является просто электрическим полем, взаимодействующим со статическими электрическими зарядами. Все объекты имеют заряды статического электричества. Все объекты состоят из множества субатомных частиц, и когда все стабильно и статично, эти частицы создают один целостный электрический заряд. Т.е. существует баланс между электрической емкостью и индуктивностью. Теперь, если нарушитель войдет в помещение, его субатомные частицы нарушат установившийся баланс электростатического поля, в связи с изменением емкости. При этом включится сигнал тревоги.

Следует учитывать, что данный тип датчиков является энергоемким, а самих датчиков требуется значительное количество, чтобы покрыть ими всю контролируемую область. Кроме того, размер, форма помещения и его содержимое могут оказаться препятствиями, что потребует существенного большего количества датчиков для полноценного контроля этого помещения.

Характеристики систем выявления вторжений. Системы IDS – это очень ценные защитные механизмы, применяемые во всех программах физической безопасности, но нужно понимать некоторые вещи перед тем, как внедрить их.

- Они дорогие и требуют вмешательства человека для реакции на сигналы тревоги

- Требуется дополнительное электропитание и резервные источники электроэнергии
- Могут быть соединены с централизованной системой безопасности
- Должны иметь отказоустойчивую конфигурацию, в которой по умолчанию датчики находятся в активном состоянии
- Должны быть устойчивы к взлому и выявлять попытки взлома

IDS – это поддерживающий механизм, который позволяет выявлять и уведомлять о вторжениях. Он не предотвращает вторжения и не задерживает нарушителей, он может только помочь узнать об этих фактах персоналу безопасности компании.

5.5. Патрульные и охранники

Одним из лучших механизмов безопасности являются охранники и/или патрульные, контролирующие территорию компании. Этот тип защитных мер более гибок, чем другие механизмы безопасности, обеспечивает хорошую реакцию на нежелательную деятельность и хорошо работает в качестве сдерживания (устрашения). Однако это может быть слишком дорого, так как требуется платить охранникам зарплату, премии, оплачивать работу в нерабочее время. К тому же люди иногда бывают ненадежны. При отборе охранников должен быть очень жесткий контроль, только это обеспечит некоторый уровень уверенности. Одна из проблем может быть связана с тем, что охранники решат сделать исключение для человека, который не следует принятым политикам компании. Это связано с человеческой природой, которая говорит нам, что нужно доверять и помогать людям. Однако такой вроде бы невинный поступок может подвергнуть компанию существенному риску.

IDS и меры физической защиты требуют крайне высокого уровня участия человека. Охранники могут иметь фиксированный пост или патрулировать определенную область. Различные компании имеют различную потребность в охранниках. Охранники могут потребоваться для проверки пропусков у посетителей и обеспечения их использования при входе. Они могут отвечать за мониторинг системы IDS и реагировать на инциденты. Они могут проверять бейджи посетителей, реагировать на пожарную тревогу, обеспечивать соблюдение внутренних правил в здании, контролировать материальные объекты, вносимые и выносимые из здания. Охранник может потребоваться для проверки защищенности дверей, окон, сейфов, хранилищ; уведомления о выявленных проблемах безопасности; обеспечения ограничения доступа в критичные помещения; сопровождения посетителей по зданию.

Охранник должен иметь ясные и четкие задачи, он должен понимать, как их нужно выполнять. Охранник должен быть обучен действиям, которые он должен предпринимать в тех или иных ситуациях. Охранник должен иметь центральный контрольный пункт, два типа радиосвязи, чтобы обеспечить гарантированное наличие связи, а также необходимые права доступа в те помещения, которые он должен защищать.

Лучшая безопасность – это комбинация различных механизмов безопасности, она не зависит от одного компонента защиты. Поэтому использование охранников следует применять вместе с другими механизмами наблюдения и выявления.

5.6. Собаки

Собаки могут быть очень полезны при выявлении нарушителей и других нежелательных ситуаций. Их слух и обоняние значительно превышают человеческие возможности, их ум и преданность могут быть использованы для защиты.

Лучшие охранные собаки проходят интенсивные тренировки для реакции на широкий круг команд и для выполнения множества задач. Собаки могут быть обучены на удержание нарушителя до прихода сотрудников безопасности или преследование и нападение на нарушителя.

Конечно, собаки могут не всегда отличить уполномоченное лицо от неуполномоченного, так,

если кто-то зайдет на работу в нерабочее время, он может получить большие неприятности. Собаки могут обеспечить хорошее дополнение к механизмам безопасности.

5.7. Контроль физического доступа

Системы контроля физического доступа могут использовать программное обеспечение для выполнения журналирования событий в отношении попыток доступа. Следующая информация должна журналироваться и анализироваться:

- Дата и время попытки доступа
- Точка входа, на которой осуществлялась попытка доступа
- Идентификатор пользователя, пытавшегося войти
- Любые неудачные попытки доступа в неразрешенное время

Также как и с компьютерными журналами регистрации событий, журналы физического доступа бесполезны без их постоянного анализа. Охранники должны быть обязаны постоянно просматривать эти журналы, однако персонал безопасности, ответственный за здание, должен также периодически анализировать эту информацию. Руководство должно знать расположение точек входа в здание и кто пытается воспользоваться ими.

Журналы регистрации попыток доступа – это детективная мера, а не превентивная. Они используются для исправления уже свершившейся ситуации, а не пытаются предотвратить попытку несанкционированного доступа.

5.8. Тестирование и тренировки

Наличие пожарных датчиков, переносных огнетушителей и систем пожаротушения – это прекрасно, но кроме этого нужно надлежащим образом обучить людей, чтобы они знали, что нужно делать в случае пожара (или другой чрезвычайной ситуации). Должен быть разработан план эвакуации и план действий в случае чрезвычайной ситуации, эти планы должны быть реально введены в действие. Эти планы должны быть документированы и легко доступны в момент кризиса. Люди, связанные с выполнением конкретных задач, должны быть информированы и обучены, как следует выполнять эти задачи, должны быть проведены упражнения, в которых люди будут выполнять те действия, которые они должны делать в чрезвычайных ситуациях. Эти тренировки должны проводиться не реже одного раза в год, а вся программа (планы) должны постоянно обновляться и совершенствоваться.

Тесты и тренировки готовят персонал к тому, с чем они могут столкнуться; в ходе них персоналу предоставляется управляемая среда для изучения стоящих перед ними задач. Эти тесты и тренировки также помогают выявить проблемы, которые не были обнаружены ранее, чтобы учесть их в процессе планирования.

Упражнения должны проходить по заранее определенному сценарию, с которым компания может однажды столкнуться в реальности. Конкретные параметры и границы этих упражнений должны быть разработаны до того, как зазвучит сигнал тревоги. Группа тестируемых должна явно определить, что предполагается протестировать и как правильно определить успех или неудачу. Группа должна согласовать время и продолжительность упражнений, кто будет принимать в них участие, кто и какие получит задачи, какие шаги должны быть выполнены. В процессе эвакуации определенные люди должны предоставить список сотрудников, в отношении которых они обязаны убедиться, что эти сотрудники покинули здание. Это единственный способ для компании узнать, остался ли кто-то внутри, и кто именно остался.

- Тестирование и тренировки:
 - Подготовка персонала

- Предоставление управляемой среды
- План эвакуации и план действий в случае чрезвычайной ситуации:
 - Должны быть разработаны
 - Должны быть введены в действие
 - Должны быть документально оформлены
 - Должны быть помещены в легко доступное место
 - Людям должны быть назначены конкретные задачи
 - Люди должны быть обучены и информированы, как выполнять эти задачи
- Тренировки должны происходить не реже одного раза в год
- Вся программа должна постоянно обновляться и совершенствоваться
- Следует определить и принять следующие параметры для проведения тренировок и тестов:
 - Время и продолжительность упражнения
 - Кто будет принимать участие в упражнении
 - Кто какие получит задания
 - Какие шаги следует предпринять

6. Резюме

Наше распределенное окружение возлагает гораздо больше обязанностей на отдельных пользователей, руководство зданий (объектов), а также на административные процедуры и защитные меры, чем в прежние времена. Физическая безопасность – это не только ночной сторож, который ходит вокруг с фонариком. В настоящее время безопасность может быть исключительно технической, она реализуется во многих формах, а также вызывает множество обязательств и правовых вопросов. Стихийные бедствия, пожары, наводнения, нарушители, вандализм, проблемы экологии, строительные материалы, а также источники энергии – все должно быть запланировано и учтено.

Каждая компания должна разработать, внедрить и поддерживать программу физической безопасности, которая содержит следующие категории контроля: сдерживание (устрашение), задержка, выявление, оценка, реагирование на инциденты. Компании необходимо определить приемлемый уровень риска и конкретные защитные меры, которые необходимы для выполнения работ по каждой категории.

Физическая безопасность не часто принимается во внимание, когда люди думают об организационной безопасности и защите активов компании, но это реальные угрозы и риски, которые необходимо учесть и подготовиться к ним. Кому будут интересны риски получения хакером доступа к сети компании, если сгорит здание компании?

Тест

Вопросы экзамена CISSP являются концептуальными, поэтому они сформулированы соответствующим образом. Задачей кандидата является выбор наилучшего из всех представленных вариантов ответа. Среди вариантов ответа может не быть идеального ответа на поставленный вопрос - кандидат должен выбрать лучший ответ из имеющихся вариантов.

1. Что является первым шагом, который нужно предпринять при выявлении пожара?

- ☐ A. Отключить систему HVAC и активировать открытие пожарных выходов
- ☐ B. Определить тип пожара
- ☐ C. Посоветовать людям покинуть здание
- ☐ D. Активировать систему пожаротушения

2. Компании нужно внедрить систему видеонаблюдения для мониторинга большой территории снаружи здания. Какая из приведенных

ниже комбинаций будет правильным выбором для компании?

- ☐ A. Широкоугольные объективы с небольшим открытием диафрагмы
- ☐ B. Широкоугольные объективы с большим открытием диафрагмы
- ☐ C. Широкоугольные объективы с большим открытием диафрагмы и небольшим фокусным расстоянием
- ☐ D. Широкоугольные объективы с большим открытием диафрагмы и большим фокусным расстоянием

3. Когда следует использовать огнетушитель Класса С вместо огнетушителя Класса А?

- ☐ A. Когда горит электрическое оборудование
- ☐ B. Когда горит дерево или бумага
- ☐ C. Когда горят горючие жидкости
- ☐ D. Когда пожар происходит в открытом пространстве

4. Какое из следующих утверждений является неправильным по отношению к объективам системы видеонаблюдения?

- ☐ A. Объективы, которые имеют ручное управление диафрагмой следует использовать для наружного наблюдения
- ☐ B. Zoom-объективы автоматически фокусируются
- ☐ C. Глубина резкости повышается по мере закрытия диафрагмы объектива
- ☐ D. Глубина резкости повышается по мере уменьшения фокусного расстояния

5. Как хладон борется с пожаром?

- ☐ A. Он сокращает способность поглощения топлива огнем
- ☐ B. Он понижает температуру горящей области, уменьшая интенсивность огня
- ☐ C. Он нарушает химическую реакцию горения
- ☐ D. Он уменьшает содержание кислорода в горящей области

6. Что такое шлюз (mantrap)?

- ☐ A. Доверенный домен безопасности
- ☐ B. Механизм логического управления доступом
- ☐ C. Комната с двумя дверями, используемая для физического контроля доступа
- ☐ D. Устройство пожаротушения

7. Что является правильным утверждением в отношении ретранслятора (transponder)?

- ☐ A. Информация с карты считывается без необходимости ее «прокатывания» через считыватель
- ☐ B. Это пассивное proximity-устройство
- ☐ C. Пользователь «прокатывает» карту через считыватель для входа в здание (или помещение)
- ☐ D. Он позволяет обмениваться токенами с сервером аутентификации

8. Когда охранники являются лучшим выбором в качестве механизма физического контроля доступа?

- ☐ A. Когда необходимо принимать обдуманные решения в различных ситуациях
- ☐ B. Когда требуется выявлять вторжения
- ☐ C. Когда бюджет безопасности мал
- ☐ D. Когда внедрены меры контроля доступа

9. Что из приведенного ниже не является правильным утверждением в отношении электростатической системы выявления вторжений?

- ☐ A. Она создает электростатическое поле и контролирует изменение его емкости
- ☐ B. Она может использоваться в качестве системы выявления вторжений для больших областей (помещений)
- ☐ C. Она контролирует баланс между электрической емкостью и индуктивностью объекта
- ☐ D. Она может выявить проникновение нарушителя в определенные границы вокруг объекта

10. Что является наиболее частой проблемой с датчиками, выявляющими вибрацию, при их использовании для безопасности периметра?

- ☐ A. Они могут быть обезврежены путем направления определенных электрических сигналов в защищаемую область
- ☐ B. Источники энергии для них могут быть легко отключены
- ☐ C. Они вызывают ложные срабатывания
- ☐ D. Они создают помехи для компьютерного оборудования

11. Что из приведенного ниже является примером защиты от ослепления?

- ☐ A. Использование объективов с автоматической регулировкой диафрагмы с коротким фокусным расстоянием
- ☐ B. Использование дежурного освещения, которое исходит от камер видеонаблюдения
- ☐ C. Направление света в сторону точек входа и от постов охраны
- ☐ D. Обеспечение использование системой освещения положительного давления

12. Что из перечисленного ниже не является основным компонентом CPTED?

- ☐ A. Естественное управление доступом
- ☐ B. Естественное наблюдение
- ☐ C. Укрепление территории

☐ D. Нацеленность на укрепленность

13. Какие проблемы могут быть вызваны влажностью в помещениях с электрическим оборудованием?

- ☐ A. Высокая влажность может привести к повышению напряжения, а низкая влажность – вызвать коррозию
- ☐ B. Высокая влажность может вызвать коррозию, а низкая – привести к статическому электричеству
- ☐ C. Высокая влажность может вызвать перепады напряжения, а низкая – привести к статическому электричеству
- ☐ D. Высокая влажность может вызвать коррозию, а низкая – привести к перепадам напряжения

14. Что означает «положительное давление» по отношению к вентиляции?

- ☐ A. Когда дверь открывается, воздух поступает внутрь помещения
- ☐ B. При пожаре источники электропитания отключаются
- ☐ C. При пожаре дым отводится в одно помещение
- ☐ D. Когда дверь открывается, воздух выходит из помещения

15. Какой из приведенных ниже ответов содержит категорию защитных мер, не относящихся к программе физической безопасности?

- ☐ A. Сдерживание и задерживание
- ☐ B. Реакция и выявление
- ☐ C. Оценка и выявление
- ☐ D. Задерживание и освещение

16. Что не является административными мерами, относящимися к процедурам действий в аварийных ситуациях?

- ☐ A. Системы выявления вторжений
- ☐ B. Обучение и повышение осведомленности
- ☐ C. Тренировки и проверки
- ☐ D. Делегирование обязанностей

17. Если является механизм контроля доступа «нормально открытым» (fail-safe), а не «нормально закрытым» (fail-secure), что это означает?

- ☐ A. По умолчанию «нет доступа»
- ☐ B. По умолчанию открыт
- ☐ C. По умолчанию закрыт
- ☐ D. По умолчанию передает сигнал тревоги, а не включает его в месте своего нахождения

18. Что из перечисленного ниже не является задерживающим механизмом?

- ☐ A. Замки
- ☐ B. Внутренние защитные средства
- ☐ C. Предупреждающие знаки
- ☐ D. Контроль доступа

19. Что является двумя основными видами идентификационных proximity-устройств?

- ☐ A. Биометрические устройства и устройства контроля доступа
- ☐ B. Устройства с картами памяти и пассивные устройства
- ☐ C. Устройства с предустановленными кодами и беспроводные устройства
- ☐ D. Активируемые пользователем устройства и считывающие устройства, обнаруживающие приближение объекта к определенным областям

20. Какой из перечисленных ниже ответов лучше всего описывает отношения между анализом рисков, приемлемым уровнем рисков, базисами, контрмерами и метриками?

- ☐ A. Результаты анализа рисков используются для определения необходимых контрмер. Базисы используются для оценки этих контрмер. Метрики используются для отслеживания эффективности контрмер, чтобы убедиться в соблюдении базисов.
- ☐ B. Результаты анализа рисков помогают руководству понять и установить приемлемый уровень рисков. Базисы основаны на этом уровне. Метрики используются для отслеживания эффективности контрмер, чтобы убедиться в соблюдении базисов.
- ☐ C. Результаты анализа рисков помогают руководству понять и установить базисы. Приемлемый уровень рисков основан на базисах. Метрики используются для отслеживания эффективности контрмер, чтобы убедиться в соблюдении базисов.
- ☐ D. Результаты анализа рисков помогают руководству понять и установить приемлемый уровень рисков. Базисы основаны на этом уровне. Метрики используются для отслеживания эффективности базисов.

21. Большинство современных систем видеонаблюдения используют CCD. Что из перечисленного ниже не является характерной чертой этого устройства?

- ☐ A. Получает внешний свет через объектив и преобразует его в электрические сигналы
- ☐ B. Улавливает свет в инфракрасном диапазоне
- ☐ C. Обеспечивает изображение повышенного качества
- ☐ D. Записывает данные на жесткий диск, а не ленту

22. Что не является препятствием для установки систем выявления вторжений и мониторинга?

- ☐ A. Их установка стоит дорого
- ☐ B. Их нельзя обойти

- ☐ C. Они требуют реакции людей
- ☐ D. Они являются источником ложных срабатываний

23. Что такое шифро-замок (cipher lock)?

- ☐ A. Замок, который использует криптографические ключи
- ☐ B. Замок, который использует такой тип ключей, которые не могут быть скопированы
- ☐ C. Замок, который использует токены и считыватели по периметру
- ☐ D. Замок, который использует клавиатуру

24. Если шифро-замок (cipher lock) имеет функцию дверного таймера, что это означает?

- ☐ A. Если дверь была открыта больше определенного времени, включается сигнал тревоги
- ☐ B. Он может быть открыт только в аварийных ситуациях
- ☐ C. Он имеет возможность включения сигнала тревоги при открытии под принуждением
- ☐ D. Он имеет возможность замещения ключа контролирующим персоналом

25. Что из приведенного ниже лучше всего описывает различие между замком с нарезкой (warded lock) и цилиндровым замком (tumbler lock)?

- ☐ A. Цилиндровый замок проще замка с нарезкой
- ☐ B. Цилиндровый замок использует внутреннюю задвижку, а замок с нарезкой – внутренние цилиндры
- ☐ C. Цилиндровый замок имеет больше компонентов, чем замок с нарезкой
- ☐ D. Замки с нарезкой в основном используются на внешних дверях, а цилиндровые – на внутренних

Домен 05. Телекоммуникации и сетевая безопасность.

Телекоммуникации и сети используют различные механизмы, устройства, программное обеспечение и протоколы, которые являются взаимосвязанными и интегрированными. Организация сетей является одним из наиболее сложных вопросов в компьютерной области, что связано в основном с большим количеством применяемых концепций и технологий. Сетевой администратор или инженер должен знать, как настраивать сетевое программное обеспечение, протоколы, сервисы, устройства, учитывая при этом вопросы их взаимодействия. Он должен устанавливать, настраивать и использовать телекоммуникационное программное обеспечение и оборудование, эффективно устранять неполадки. Специалист по безопасности должен не только понимать эти вопросы, но и быть способным проанализировать их на несколько уровней глубже, чтобы понять, где в сети могут возникнуть уязвимости. Это может быть крайне сложной задачей, что делает ее более интересной.

Как специалист по безопасности, вы не можете советовать другим, как нужно защищать сеть, если вы сами не в полной мере понимаете, как она работает. Например, для защиты приложения, содержащего уязвимость переполнения буфера, вы должны понимать, что такое переполнение буфера, как эксплуатируется эта уязвимость, как правильно выявить ее наличие и, возможно, как правильно написать код программы для удаления этой уязвимости. Для защиты сетевой архитектуры, вы должны разбираться в различных сетевых платформах, сетевых устройствах, принципах передачи данных по сетям. Вы должны понимать, как работают различные протоколы, каково их назначение, как они взаимодействуют с другими протоколами, каким образом они могут предоставить эксплуатируемые уязвимости, как выбрать и внедрить подходящие типы протоколов в имеющуюся среду. Вы также должны понимать, как работают различные типы межсетевых экранов, маршрутизаторов, коммутаторов и мостов, когда один из них предпочтительнее другого, какую степень безопасности обеспечивает каждый из них.

Множество различных типов устройств, протоколов и механизмов безопасности, имеющихся в сети, предоставляют различную функциональность и многоуровневый подход к безопасности. В рамках безопасности разделение на уровни очень важно, поскольку если атакующий сможет преодолеть один уровень, другой уровень должен задержать его, обеспечив защиту внутренней сети. Во множестве сетей есть маршрутизаторы, межсетевые экраны, системы обнаружения вторжений, антивирусное программное обеспечение и т.п. Каждый из этих компонентов реализует определенную часть безопасности, но все они должны работать совместно для обеспечения многоуровневого подхода к безопасности.

Телекоммуникации – это электрическая передача данных между системами с использованием аналоговых, цифровых или беспроводных видов передачи. Данные могут проходить по медным проводам, коаксиальным кабелям, радиоволнам, телефонным сетям общего пользования (PSTN – public-switched telephone network), оптоволоконным кабелям провайдеров, коммутаторам и маршрутизаторам. Существуют четкие линии, разделяющие среду передачи, технологии, протоколы и оборудование, которое они используют. Однако эти линии смазываются, когда мы рассматриваем, как созданные на пользовательской рабочей станции данные проходят через сложную систему сетевых проводов к маршрутизатору, который отделяет внутреннюю сеть компании от остального мира, через АТМ-коммутатор (АТМ – Asynchronous Transfer Mode – асинхронный режим передачи) провайдера, через облако АТМ, к маршрутизатору сети другой компании и далее к рабочей станции другого пользователя. Каждая часть интересна, но когда они объединены и работают вместе – это приводит в ужас.

Телекоммуникациями обычно называют телефонные системы, провайдеров услуг и службы связи. Большинство телекоммуникационных систем регулируется правительством и международными организациями. В США телекоммуникационные системы (включая

передачу голоса и данных) регулируются Федеральным агентством по связи (FCC – Federal Communications Commission). В Канаде – SITT (Spectrum, Information Technologies and Telecommunications). Отдельные организации разрабатывают международные политики, рекомендательные стандарты, совместно работают для стандартизации и возможности взаимодействия различных технологий.

Основными стандартизирующими организациями являются Международный союз по электросвязи (ITU – International Telecommunication Union) и Международная организация по стандартизации (ISO – International Standards Organization). Разработанные ими модели и стандарты определяют наши сегодняшние технологии, которые рассматриваются в этом Домене.

1. Эталонная модель взаимодействия открытых систем

ISO – это международная организация, которая разрабатывает международные стандарты. В начале 1980-х, ISO работала над набором протоколов, которые могли бы использоваться всеми производителями по всему миру, позволяя взаимодействовать различным сетевым устройствам. Это дало всем производителям продуктов и технологий надежду на возможность взаимодействия через международные и технические границы. Реальный набор протоколов не удалось сделать стандартом, но модель этого набора протоколов (модель OSI) применяется в качестве абстрактной платформы, которую используют большинство операционных систем и протоколов.

Многие люди думают, что описание модели OSI появилось еще на заре компьютерной эры и задало направление для многих, если не всех, сетевых технологий. Однако это не так. Реально эта модель была выпущена в 1984 году, когда основы Интернета были уже разработаны и реализованы, а основной Интернет-протокол уже использовался много лет. Набор протоколов TCP/IP (Transmission Control Protocol / Internet Protocol) в действительности имеет свою собственную модель, которая в наше время часто используется для проверки знаний и понимания сетевых вопросов. На Рисунке 5-1 показаны различия между сетевыми моделями OSI и TCP/IP. В этом Домене мы будем ориентироваться в основном на модель OSI.

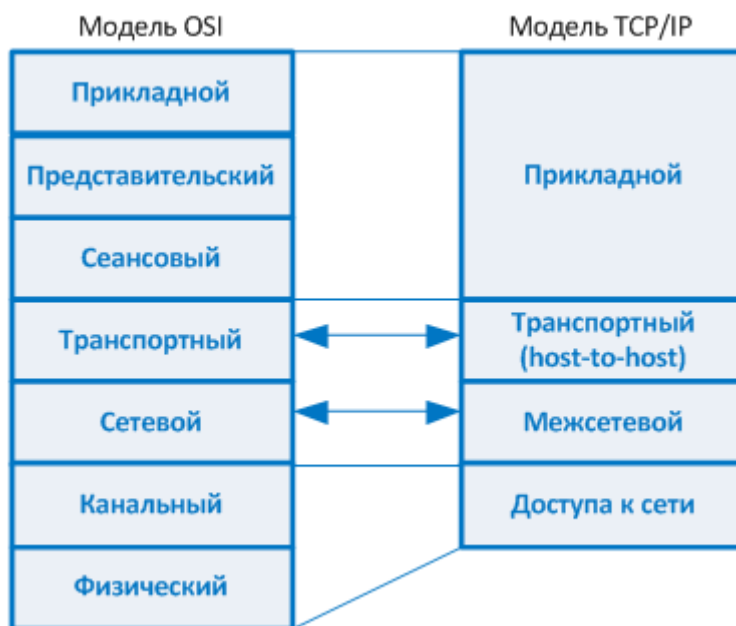


Рисунок 5-1. Сетевые модели OSI и TCP/IP

ПРИМЕЧАНИЕ. Уровень хост-хост иногда называют транспортным уровнем в модели TCP/IP.

1.1. Протокол

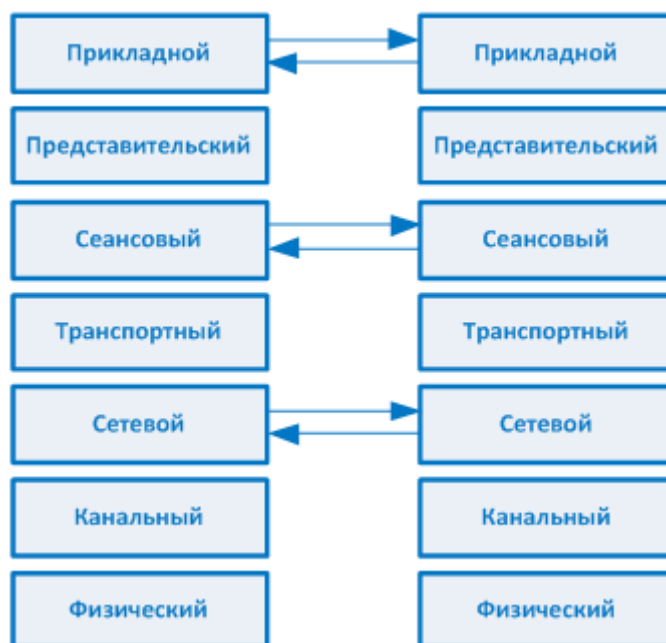
Сетевой *протокол* – это стандартный набор правил, определяющих порядок сетевого взаимодействия систем. Две различные системы, использующие одинаковый протокол, могут взаимодействовать и понимать друг друга, несмотря на их различия, так же как два человека могут разговаривать и понимать друг друга, используя одинаковый язык.

Модель OSI, описанная в стандарте ISO 7498 (часть 1, 3, 4), предоставляет важнейшее руководство, которое используется производителями, инженерами, разработчиками и т.д. Модель разделяет задачи организации сетей, протоколы и службы по различным уровням. Каждый уровень отвечает за свою часть взаимодействия компьютеров в сети. Каждый уровень имеет определенную функциональность, которую реализуют работающие на нем службы и протоколы.

Цель модели OSI – помочь всем производителям разрабатывать продукты, которые будут работать в рамках открытой сетевой архитектуры. *Открытой сетевой архитектурой* не владеют производители, она не является чьей-то собственностью, поэтому она может легко интегрироваться в различные технологии и реализовываться производителями в своих технологиях. Производители используют модель OSI как отправную точку для разработки своих собственных сетевых платформ. Они используют модель OSI как основу и разрабатывают свои собственные протоколы и интерфейсы, имеющие функциональность, отличающуюся или перекрывающую функциональность других производителей. Однако поскольку такие производители в основе своих разработок используют модель OSI, интеграция их продуктов с продуктами других производителей не является сложной задачей – обеспечить функциональную совместимость при этом значительно проще, чем если бы производители не основывались на единой модели.

Хотя компьютеры взаимодействуют на физическом уровне (электрические сигналы, проходящие по проводам от одного компьютера до другого), они взаимодействуют и на более высоком уровне – через логические каналы. Каждый протокол на определенном уровне модели OSI на одном компьютере, взаимодействует с соответствующим протоколом, работающим на том же уровне модели OSI на другом компьютере. Это осуществляется посредством *инкапсуляции*.

Логическое перемещение данных



Инкапсуляция работает следующим образом. Сообщение создается программой на одном компьютере и затем проходит по стеку протоколов сверху вниз. Протокол на каждом уровне добавляет собственную информацию к сообщению; поэтому размер сообщения увеличивается по мере его перемещения по стеку протоколов. Затем сообщение отправляется компьютеру-получателю, который проводит обратное преобразование, выполняя в обратном порядке те же шаги, что и компьютер-отправитель, выполнивший инкапсуляцию. Например, на канальном уровне извлекается только информация, относящаяся к канальному уровню, после чего сообщение отправляется на уровень выше. Затем на сетевом уровне отделяются и обрабатываются только данные сетевого уровня, после чего пакет отправляется дальше и т.д. Так взаимодействуют компьютеры на логическом уровне. Информация, извлекаемая компьютером-получателем, говорит ему о том, как нужно правильно интерпретировать и обрабатывать пакет. Инкапсуляция данных показана на Рисунке 5-2.

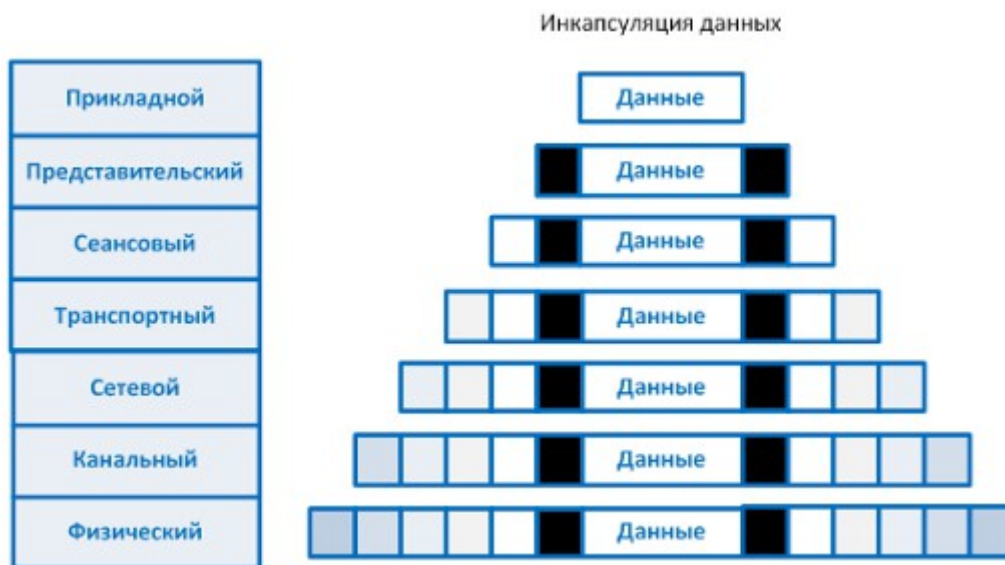


Рисунок 5-2. Каждый уровень OSI добавляет собственную информацию к пакету данных

Протокол на каждом уровне выполняет определенные задачи и контрольные функции, используя известный синтаксис данных. Каждый уровень имеет специальный интерфейс (точку соединения), который позволяет ему взаимодействовать с тремя другими уровнями: 1) взаимодействие с интерфейсом вышестоящего уровня, 2) взаимодействие с интерфейсом нижестоящего уровня, и 3) взаимодействие с интерфейсом того же уровня на компьютере-получателе, адрес которого указан в пакете. Контрольные функции, добавляемые протоколами на каждом уровне, реализуются в форме заголовка (header) и окончания (trailer) пакета.

Преимущество разделения на уровни и обеспечения отдельной функциональности для каждого уровня состоит в том, что различные протоколы, технологии и функции могут взаимодействовать друг с другом и обеспечивать необходимый интерфейс, позволяющий осуществлять такое взаимодействие. Это означает, что приложение может использовать для создания и передачи сообщений по сети, например, прикладные протоколы Novell, транспортные протоколы Apple и сеансовые протоколы IBM. Протоколы, технологии и компьютеры, работающие в рамках модели OSI, считаются *открытыми системами*. Открытые системы могут взаимодействовать с другими открытыми системами, т.к. в них используется стандартные международные протоколы и интерфейсы. Спецификация интерфейса каждого уровня хорошо структурирована, что позволяет производителям разрабатывать дополнительные модули, заменяющие стандартные модули с целью добавления специфичных функций и расширений.

Понимание функциональности каждого уровня OSI и соответствующих протоколов, работающих на нем, поможет вам понять в целом весь процесс взаимодействия между

компьютерами. Когда вы поймете этот процесс, вы сможете более детально рассмотреть каждый протокол, чтобы увидеть весь спектр предоставляемых им функций, а также слабости в защите каждой из них.

Прикладной уровень

Прикладной уровень (application layer, уровень 7) работает в непосредственной близости от пользователя и обеспечивает передачу файлов, обмен сообщениями, терминальные сеансы и многое другое. Этот уровень не содержит реальных приложений, но содержит протоколы, поддерживающие работу приложений. Когда приложению нужно отправить данные по сети, оно передает команды и данные протоколам прикладного уровня. Этот уровень обрабатывает и надлежащим образом форматирует данные, а затем передает их вниз, на следующий уровень модели OSI. Это продолжается до тех пор, пока в созданные на прикладном уровне данные не будет добавлена вся информация каждого уровня, необходимая для передачи данных по сети. Затем данные помещаются в сетевой кабель и передаются, пока не будут доставлены на компьютер-получатель.

Протоколами, работающими на этом уровне, являются, например, SMTP (Simple Mail Transfer Protocol - Простой протокол передачи почты), HTTP (Hypertext Transfer Protocol - Протокол передачи гипертекста), LPD (Line Printer Daemon - Протокол службы построчной печати), FTP (File Transfer Protocol - Протокол передачи файлов), Telnet, TFTP (Trivial File Transfer Protocol - Простой протокол передачи файлов). На Рисунке 5-3 показано, как приложения взаимодействуют с нижестоящими протоколами посредством интерфейсов API. Если пользователь создает запрос на отправку сообщения электронной почты через почтовый клиент Outlook, Outlook отправляет эту информацию по протоколу SMTP. SMTP добавляет свою информацию к пользовательской информации и отправляет ее вниз на представительский уровень.

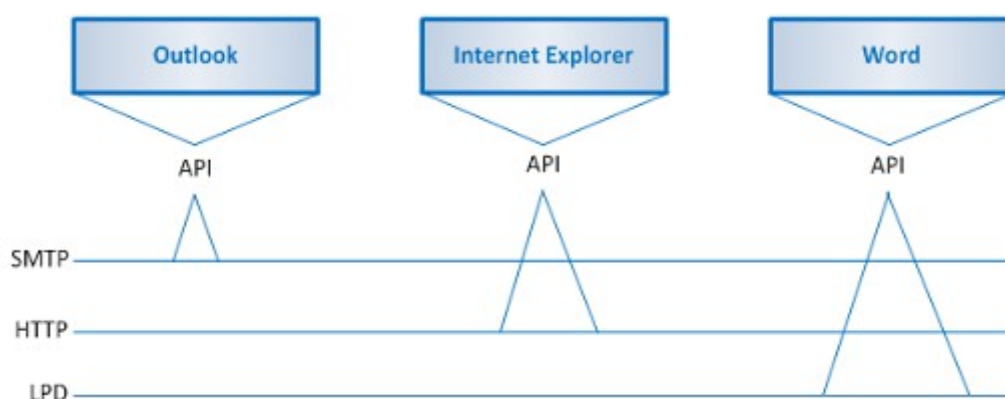


Рисунок 5-3. Приложение отправляет запрос API, который является интерфейсом для поддерживающего протокола

ПРИМЕЧАНИЕ. Прикладной уровень в модели TCP/IP эквивалентен комбинации прикладного, представительского и сеансового уровней в модели OSI (см. Рисунок 5-1).

Представительский уровень

Представительский уровень (presentation layer, уровень 6) получает информацию от протоколов прикладного уровня и преобразует ее в формат, понятный всем компьютерам, использующим модель OSI. Этот уровень обеспечивает преобразование данных в структуру, которая может быть правильно обработана системой получателя. Это означает, что если пользователь создал документ Word и отправил его нескольким людям, им не обязательно иметь именно Word для работы с текстами; каждый из их компьютеров может принять этот файл, определить его формат и представить его пользователю в виде документа. Именно процесс преобразования данных, выполняемый на уровне представления, позволяет осуществлять это. Например, когда компьютер с Windows XP получает файл с компьютера с системой Linux, информация в заголовке файла объясняет ему что это за тип файла.

Операционная система Windows XP имеет список поддерживаемых типов файлов и таблицу, описывающую, какую программу следует использовать для открытия каждого типа файлов. Например, отправитель создал текстовый документ в редакторе Open Office, а получатель использует Word 2003. Получатель все равно может открыть этот файл, так как представительский уровень на системе отправителя преобразует этот файл в формат ASCII, а компьютер получателя знает, что открывать такие файлы нужно в текстовом редакторе Word 2003.

Представительский уровень не затрагивает сами данные – он работает с синтаксисом и форматом этих данных. Это похоже на работу транслятора, который переводит используемый приложением формат в стандартный формат, используемый для передачи сообщений по сети. Если пользователь использует приложение Corel Draw для сохранения изображений, графические файлы могут находиться в различных форматах, например, TIFF, GIF, JPEG. Представительский уровень добавит информацию, которая сообщит компьютеру получателя о типе файла, о том, как его обрабатывать и отображать. Это действует даже в том случае, если на компьютере получателя нет программы Corel Draw, операционная система все равно сможет отобразить отправленное изображение, так как оно было преобразовано в стандартный формат. На Рисунке 5-4 показано преобразование файлов в различные стандартные типы файлов.

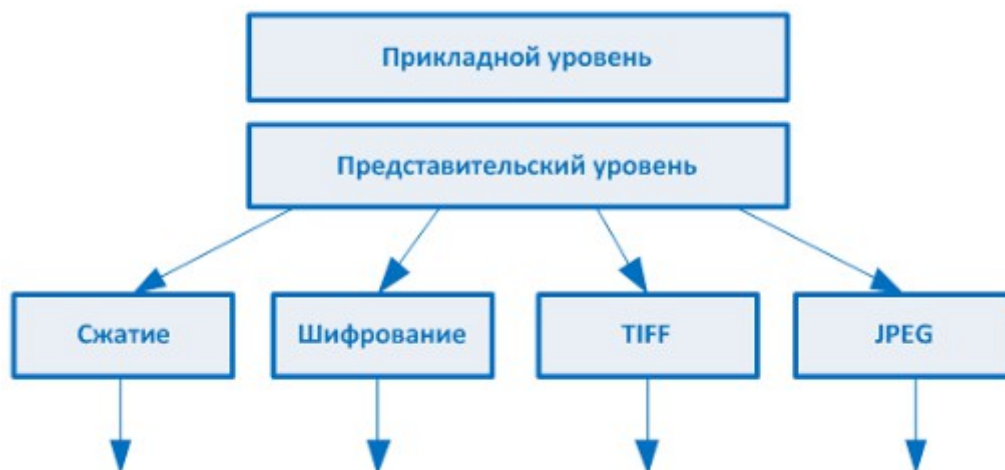


Рисунок 5-4. Представительский уровень получает данные с прикладного уровня и переводит их в стандартный формат

Этот уровень также может выполнять сжатие и шифрование передаваемой информации. Если программа потребовала, чтобы конкретный файл был сжат и зашифрован перед отправкой по сети, представительский уровень предоставит необходимую информацию об этом компьютеру получателя. Эта информация может содержать сведения об использованном типе шифрования или компрессии, порядке правильного представления информации пользователю. К передаваемому пакету данных могут добавляться инструкции, рассказывающие системе получателя как правильно расшифровать или разжать данные.

Сеансовый уровень

Когда двум приложениям требуется взаимодействовать друг с другом или передавать информацию, между ними должен быть установлен сеанс связи. **Сеансовый уровень** (session layer, уровень 5) отвечает за создание соединений между двумя приложениями и поддержку этих соединений в процессе передачи данных, а также контролирует закрытие этих соединений.

Работа сеансового уровня делится на три этапа (работа этого уровня похожа на работу телефонной связи): установление соединения, передача данных, завершение соединения. Он обеспечивает перезапуск сеанса, его восстановление (в случае необходимости), а также

полную поддержку сеанса. Когда взаимодействие окончено, маршрут разрывается, а все параметры возвращаются в свое исходное состояние. Этот процесс известен как *диалог-менеджмент* (dialog management). На Рисунке 5-5 изображены три этапа сеанса. Вот некоторые протоколы, работающие на этом уровне: NFS (Network File System - Сетевая файловая система), SQL (Structured Query Language - Язык структурированных запросов), NetBIOS, RPC (Remote Procedure Call - Удаленный вызов процедур).

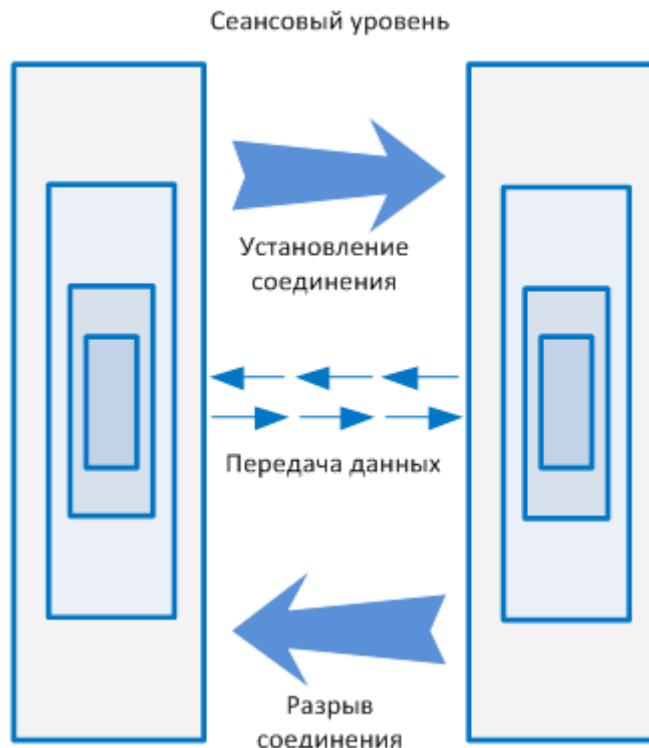


Рисунок 5-5. Сеансовый уровень устанавливает соединение, поддерживает его, а затем разрывает, когда взаимодействие окончено

Протокол сеансового уровня может обеспечивать взаимодействие двух приложений в трех различных режимах:

- **Симплексный** (Simplex). Данные могут передаваться только в одном направлении.
- **Полудуплексный** (Half-duplex). Данные могут передаваться в обоих направлениях, но только одно приложение может отправлять информацию в единицу времени.
- **Дуплексный** (Full-duplex). Данные могут передаваться в двух направлениях, и оба приложения могут одновременно передавать информацию.

Многие тратят много времени, чтобы понять различия между сеансовым и транспортным уровнями, т.к. их определения звучат похоже. Протоколы сеансового уровня управляют взаимодействием приложение-приложение, а протоколы транспортного уровня – компьютер-компьютер. Например, если вы используете продукт, работающий в соответствии с моделью клиент/сервер, реально на вашем компьютере установлена лишь небольшая часть этого продукта (клиентская часть), а остальные части продукта работают на другом компьютере (серверная часть). Для управления взаимодействием между этими двумя частями данного программного обеспечения используется протокол сеансового уровня. Протокол сеансового уровня выполняет функциональность промежуточного программного обеспечения (middleware), позволяя взаимодействовать программному обеспечению на двух различных компьютерах.

Транспортный уровень

Когда два компьютера собираются начать взаимодействие посредством протокола с

предварительным установлением соединения (connection-oriented protocol), они сначала договариваются, как много информации каждый компьютер будет отправлять в единицу времени, как будет проверяться целостность данных при получении, и как будут выявляться потери пакетов в пути. Два компьютера договариваются об этих параметрах на **транспортном уровне** (transport layer, уровень 4) посредством процесса, называемого «рукопожатием» (handshaking). Предварительное соглашение по этим вопросам помогает обеспечить более надежную передачу данных, выявление и исправление ошибок, восстановление, управление потоком, оптимизирует работу сетевых служб, необходимых для выполнения данной задачи. Транспортный уровень обеспечивает сервисы сквозной (end-to-end) передачи данных и устанавливает логическое соединение между двумя взаимодействующими компьютерами.

ПРИМЕЧАНИЕ. Протоколы с предварительным установлением соединения (такие как TCP) обеспечивают более надежную передачу данных по сравнению с протоколами без установления соединения (такими как UDP). Это отличие рассматривается более детально в разделе «TCP/IP» далее в этом Домене.

Функциональность сеансового и транспортного уровней похожа (в том смысле, что они оба устанавливают некий тип сеанса или виртуального соединения для осуществления взаимодействия). Различие протоколов на этих уровнях состоит в том, что на сеансовом уровне устанавливается соединение между *приложениями*, тогда как на транспортном – между *компьютерами*. Например, у нас может быть три различных приложения на компьютере А, взаимодействующих с тремя приложениями на компьютере В. Протокол транспортного уровня похож на автобус. Он не знает и не заботится о том, какие приложения взаимодействуют друг с другом, он просто предоставляет механизм передачи данных от одного компьютера другому.

Транспортный уровень получает данные от множества различных приложений и собирает данные в потоки для их корректной передачи по сети. Основными протоколами, работающими на данном уровне, являются TCP (Transmission Control Protocol - Протокол управления передачей), UDP (User Datagram Protocol - Протокол пользовательских датаграмм) и SPX (Sequenced Packet Exchange - Последовательный обмен пакетами). Информация передается на транспортный уровень от различных сущностей, находящихся на более высоких уровнях, а транспортный уровень должен собрать информацию в поток, как показано на Рисунке 5-6. Потоки состоят из различных сегментов поступивших данных. Также как автобус может перевозить разных людей, так и протокол транспортного уровня может передавать различные типы данных от различных приложений. (Транспортный (host-to-host) уровень в модели TCP/IP аналогичен транспортному уровню в модели OSI, как показано на Рисунке 5-1).

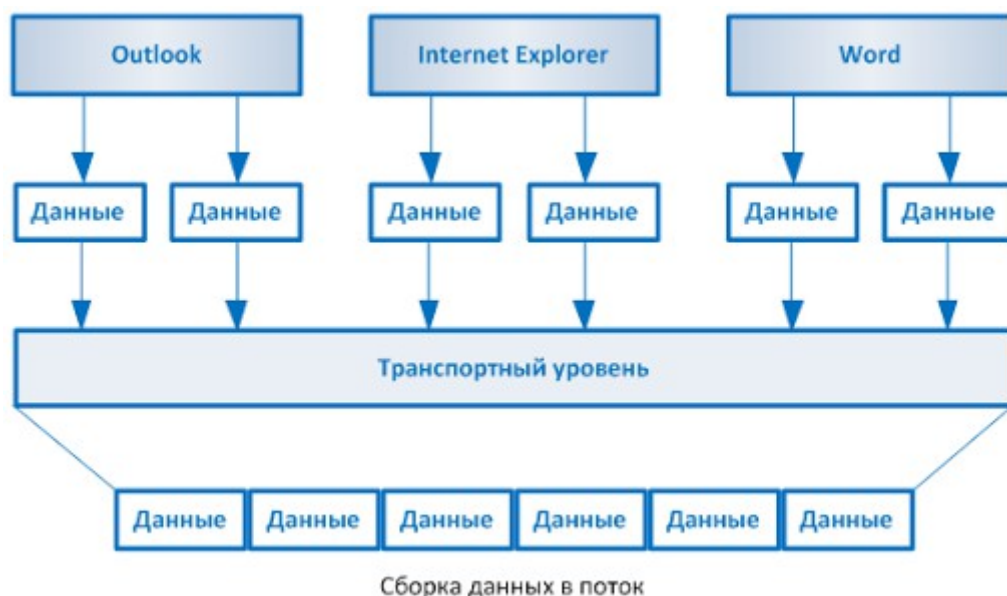


Рисунок 5-6. TCP форматирует данные от приложений в поток, подготавливая их к передаче

ПРИМЕЧАНИЕ. В различной документации некоторые протоколы могут размещаться на разных уровнях. Например, во многих описаниях протокол SSL размещен на сеансовом уровне, в то время как другие описания размещают его на транспортном уровне. Нельзя точно сказать, что правильно, а что нет. Модель OSI пытается отразить реальность с помощью «квадратиков», поэтому некоторые протоколы действительно могут относиться к разным уровням. На самом деле протокол SSL состоит из двух протоколов – один находится на сеансовом уровне, а другой – на транспортном. Для сдачи экзамена CISSP вам нужно разместить протокол SSL на транспортном уровне.

Сетевой уровень

Основной задачей **сетевого уровня** (network layer, уровень 3) является вставка в заголовок пакета информации, необходимой для правильной адресации и маршрутизации этого пакета с целью доставки правильному получателю. В сети множество маршрутов могут вести к одному и тому же получателю. Протоколы сетевого уровня должны выбрать наилучший маршрут для пакета. На этом уровне протоколы маршрутизации создают и поддерживают свои таблицы маршрутизации. Эти таблицы являются картой сети. Когда требуется отправить пакет от компьютера А компьютеру М, протокол проверяет таблицу маршрутизации, добавляет необходимую информацию в заголовок пакета и отправляет его по выбранному маршруту.

Работающие на данном уровне протоколы не контролируют доставку пакетов. Они зависят от протоколов транспортного уровня, которые выявляют любые проблемы и повторно отправляют пакеты при необходимости. Наиболее известный протокол, работающий на данном уровне – IP (Internet Protocol - Межсетевой протокол), хотя на этом уровне работает и масса других протоколов маршрутизации и маршрутизируемых протоколов, например, ICMP (Internet Control Message Protocol - Межсетевой протокол управляющих сообщений), RIP (Routing Information Protocol - Протокол информации маршрутизации), OSPF (Open Shortest Path First - Первоочередной выбор кратчайшего пути), BGP (Border Gateway Protocol - Протокол граничного шлюза), IGMP (Internet Group Management Protocol - Протокол управления межсетевыми группами). На Рисунке 5-7 показано, что пакеты могут использовать различные маршруты, и что сетевой уровень вносит информацию для маршрутизации в заголовок для того, чтобы помочь доставить пакет по назначению. (Межсетевой уровень в модели TCP/IP соответствует сетевому уровню в модели OSI, как показано на Рисунке 5-1).

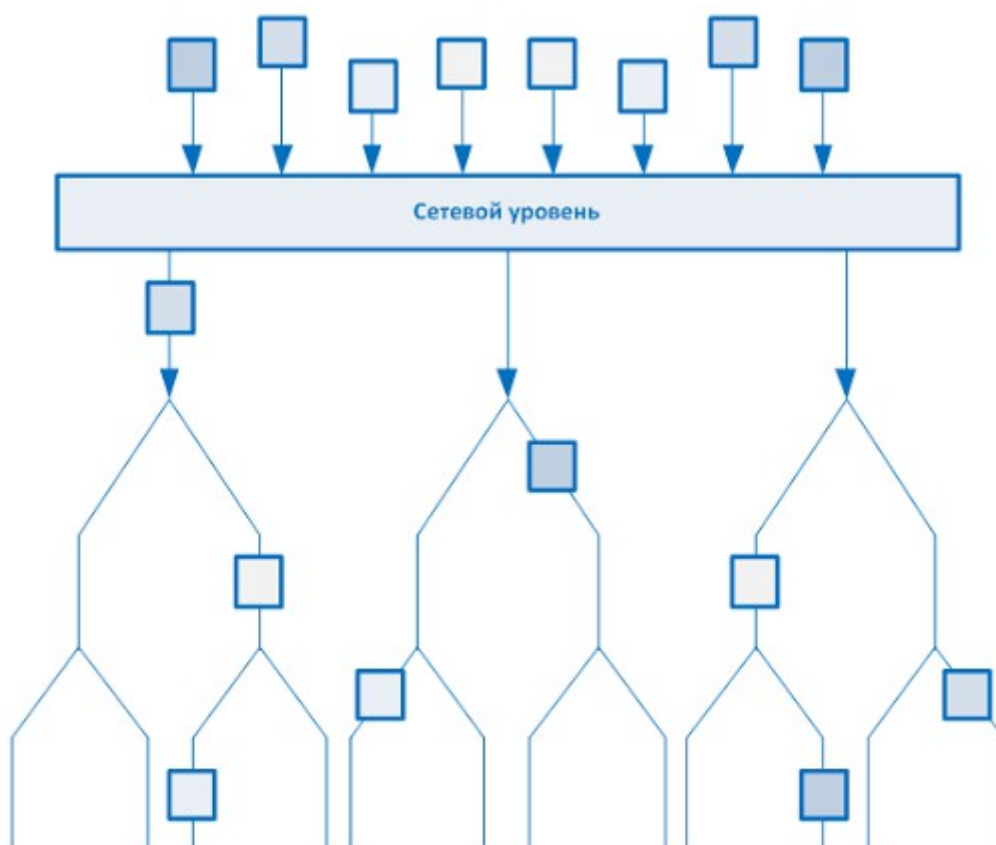


Рисунок 5-7. Сетевой уровень определяет самый эффективный маршрут для каждого пакета

Канальный уровень

Мы продолжаем движение вниз по стеку протоколов, приближаясь непосредственно к сетевым проводам, по которым передаются все данные. Внешний формат пакетов данных незначительно меняется на каждом уровне, но, в конце концов, они приходят к точке, где нужно передавать двоичные данные по технологии LAN (Local Area Network - Локальная вычислительная сеть) или WAN (Wide Area Network - Глобальная вычислительная сеть). Это происходит на **канальном уровне** (data link layer, уровень 2).

Технологии LAN и WAN могут использовать различные протоколы, сетевые карты (NIC – Network Interface Card), кабели и методы передачи. Каждая из этих технологий имеет различные форматы структуры данных, кроме того, эти технологии по-разному интерпретируют электрическое напряжение. Канальный уровень – это уровень, на котором сетевой стек знает, какой должен применяться формат кадра (frame) данных для правильной передачи через Token Ring, Ethernet, ATM (Asynchronous Transfer Mode - Асинхронный режим передачи) или FDDI (Fiber Distributed Data Interface - Распределенный волоконнооптический интерфейс данных) сети. Если сеть является сетью Ethernet, например, все компьютеры ожидают заголовок определенной длины, размещение флагов в определенных местах в пакете, информацию о содержимом пакета в определенных полях, на определенных местах. В сетях Token Ring компьютеры ожидают пакеты с другим расположением большинства параметров и кадры иного формата. Канальный уровень отвечает за обеспечение правильного взаимодействия в рамках используемой технологии, преобразование данных в необходимый формат для передачи на физический уровень. Также канальный уровень управляет переупорядочиванием кадров, которые принимаются не последовательно, и уведомляет вышестоящие протоколы о возникающих условиях для ошибок при передаче (transmission error conditions).

Канальный уровень делится на два функциональных подуровня – **Управление логической связью** (LLC – Logical Link Control) и **Управление доступом к среде** (MAC – Media Access

Control). LLC определен в спецификации IEEE 802.2, он взаимодействует с протоколом, находящимся непосредственно над ним, на сетевом уровне. MAC использует соответствующие загруженные протоколы для взаимодействия с протоколом физического уровня. IEEE-спецификация MAC для Ethernet приведена в 802.3, для Token Ring – в 802.5, для беспроводных сетей – в 802.11 и т.д. Если мы будем смотреть описания в стандарте IEEE (такие как 802.11, 802.16, 802.3 и т.д.), они будут описывать протоколы, работающие на подуровне MAC канального уровня стека протоколов.

Вот некоторые протоколы, которые работают на канальном уровне: SLIP (Serial Line Internet Protocol - Межсетевой протокол последовательной передачи), PPP (Point-to-Point Protocol - Протокол точка-точка), RARP (Reverse Address Resolution Protocol - Обратный протокол преобразования адресов), L2F (Layer 2 Forwarding - Туннельная схема пересылки на уровне 2), L2TP (Layer 2 Tunneling Protocol - Протокол туннелирования на уровне 2), FDDI (Fiber Distributed Data Interface - Распределенный волоконнооптический интерфейс данных), ISDN (Integrated Services Digital Network - Цифровая сеть с интегрированным обслуживанием). На Рисунке 5-8 показано как канальный уровень преобразует информацию в биты, а физический уровень преобразует эти биты в электрические сигналы. (Уровень доступа к сети в модели TCP/IP соответствует комбинации канального и физического уровней модели OSI, как показано на Рисунке 5-1).

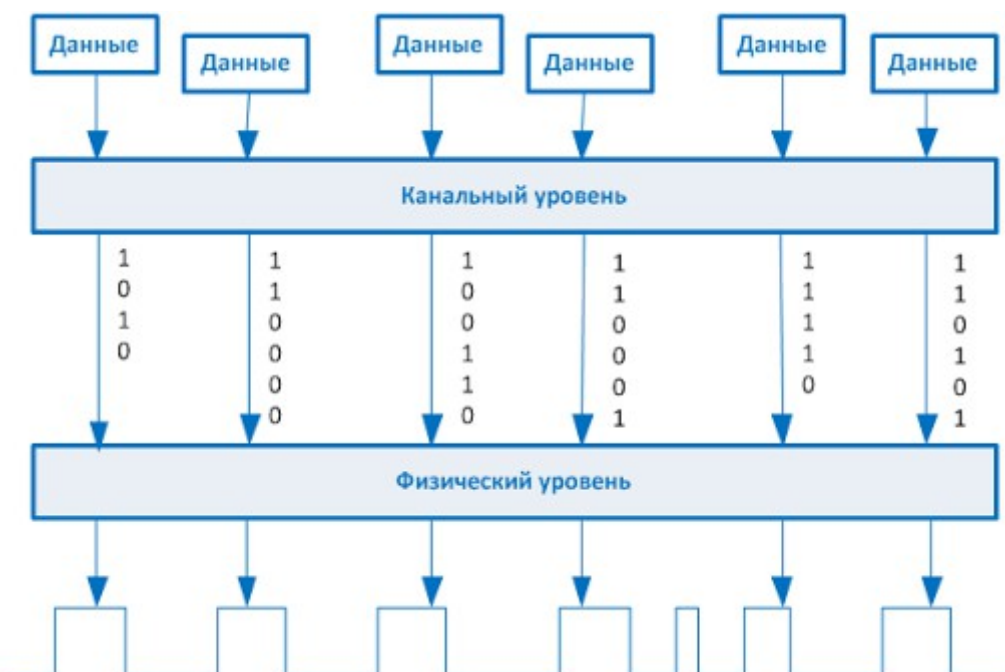


Рисунок 5-8. Канальный уровень преобразует данные в биты для их передачи на физический уровень

Каждая сетевая технология (Ethernet, Token Ring и т.д.) определяет допустимые типы кабелей (коаксиальный, витая пара, оптоволоконный), которые требуются для работы сети. Также каждая сетевая технология определяет электрические сигналы и битовые маски (bit pattern). Это означает, например, что сигнал в 0,5 вольта может представлять 0 в одной технологии и 1 в другой. Протокол канального уровня определяет правильную битовую маску, а протокол физического уровня преобразует эту информацию в электрическую кодировку и электрические переходы состояний. Сетевые карты – это мост между канальным и физическим уровнями. Драйвер сетевой карты кодирует биты на канальном уровне в зависимости от используемой сетевой технологии (Ethernet, Token Ring, FDDI и т.д.). Затем эти биты преобразуются в электрические состояния на физическом уровне и помещаются в сетевой кабель для передачи.

Физический уровень

Физический уровень (physical layer, уровень 1) преобразует биты в напряжения для передачи. Схемы сигналов и напряжений имеют различное значение в различных технологиях LAN и WAN. Если пользователь отправляет данные по телефонной сети через модем, формат данных, электрических сигналов и управление функционированием будут сильно отличаться от случая, когда пользователь отправляет данные через сетевую карту по витой паре в локальной сети. Механизмы, управляющие передачей данных по телефонной сети или по витой паре, работают на физическом уровне. Этот уровень контролирует синхронизацию, скорость передачи, шумы в линии, доступ к среде. Спецификации для физического уровня включают время изменения напряжения, уровни напряжения, физические коннекторы для электрической, оптической или механической передачи.

1.2. Функции и Протоколы модели OSI

Для сдачи экзамена вам нужно знать, какие функции выполняются на различных уровнях модели OSI, а также соответствующие протоколы, работающие на этих уровнях. Далее приводится краткий обзор каждого уровня и его протоколов.

Прикладной уровень

Протоколы прикладного уровня осуществляют передачу файлов, обеспечивают работу виртуальных терминалов, реализуют управление сетью, выполняют сетевые запросы приложений. Ниже указаны несколько протоколов, которые работают на этом уровне:

- File Transfer Protocol (FTP)
- Trivial File Transfer Protocol (TFTP)
- Simple Network Management Protocol (SNMP)
- Simple Mail Transfer Protocol (SMTP)
- Telnet
- Hypertext Transfer Protocol (HTTP)

Представительский уровень

Сервисы представительского уровня обеспечивают перевод в стандартные форматы, компрессию/декомпрессию данных, шифрование/расшифрование данных. На этом уровне нет протоколов, на нем работают только сервисы. Вот некоторые из сервисов представительского уровня:

- American Standard Code for Information Interchange (ASCII)
- Extended Binary-Coded Decimal Interchange Mode (EBCDIC)
- Tagged Image File Format (TIFF)
- Joint Photographic Experts Group (JPEG)
- Motion Picture Experts Group (MPEG)
- Musical Instrument Digital Interface (MIDI)

Сеансовый уровень

Протоколы сеансового уровня устанавливают соединения между приложениями, поддерживают управление диалогом, создают, поддерживают, разрывают коммуникационные каналы. Вот несколько протоколов, работающих на этом уровне:

- Network File System (NFS)
- NetBIOS

- Structured Query Language (SQL)
- Remote procedure call (RPC)

Транспортный уровень

Протоколы транспортного уровня управляют сквозной передачей и сегментацией в потоках данных. Вот протоколы этого уровня:

- Transmission Control Protocol (TCP)
- User Datagram Protocol (UDP)
- Secure Sockets Layer (SSL) / Transport Layer Security (TLS)
- Sequenced Packet Exchange (SPX)

Сетевой уровень

Протоколы сетевого уровня отвечают за сервисы межсетевого взаимодействия, адресацию и маршрутизацию. Вот несколько протоколов этого уровня:

- Internet Protocol (IP)
- Internet Control Message Protocol (ICMP)
- Internet Group Management Protocol (IGMP)
- Routing Information Protocol (RIP)
- Open Shortest Path First (OSPF)
- Novel Internetwork Packet Exchange (IPX)

Канальный уровень

Протоколы канального уровня преобразуют данные в кадры LAN или WAN для передачи, преобразуют сообщения в биты, определяют, как компьютер взаимодействует с сетью. Этот уровень делится на подуровни LLC и MAC. Вот несколько протоколов этого уровня:

- Address Resolution Protocol (ARP)
- Reverse Address Resolution Protocol (RARP)
- Point-to-Point Protocol (PPP)
- Serial Line Internet Protocol (SLIP)

Физический уровень

Сетевые карты и драйверы преобразуют биты в электрические сигналы и контролируют физические аспекты передачи данных, включая оптические, электрические и механические требования. Вот несколько стандартов интерфейсов этого уровня:

- High-Speed Serial Interface (HSSI)
- X.21
- EIA/TIA-232 и EIA/TIA-449

ПРИМЕЧАНИЕ. Среди сервисов безопасности, определенных в модели безопасности OSI, обеспечение целостности данных (защита от модификации и уничтожения), конфиденциальности (защита от разглашения), аутентификации (проверка идентичности источника взаимодействия), а также сервисы управления доступом (позволяющие использовать механизмы предоставления и запрета доступа).

1.3. Соместная работа уровней

Модель OSI используется в качестве основы для множества продуктов множеством производителей. Различные типы устройств и протоколов работают на разных уровнях этой семиуровневой модели. В то время как компьютеры могут интерпретировать и обрабатывать данные на каждом из семи уровней, маршрутизаторы понимают информацию только на сетевом уровне, поскольку их основная функция заключается в маршрутизации пакетов, что не требует знаний другой информации пакета. Маршрутизаторы отделяют информацию заголовка при загрузке данных сетевого уровня, на котором размещается информация о маршрутизации и IP-адресах. Маршрутизатор анализирует эту информацию, чтобы принять решение – куда следует отправлять пакет. Мосты понимают только информацию канального уровня, а повторители – физического. Рисунок 5-9 показывает, что понимают устройства каждого типа из уровней модели OSI.

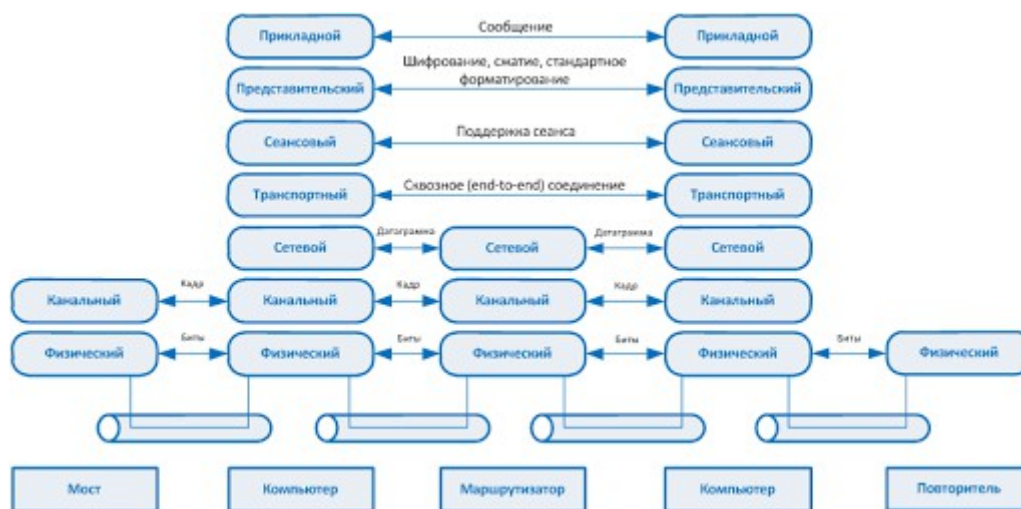


Рисунок 5-9. Каждое устройство работает на отдельном уровне модели OSI

Ссылки по теме:

- [Protocols.com listing of data communications protocols](#)
- [Google listings of protocols](#)
- [Linktionary definition of OSI model](#)
- [Wikipedia entry for OSI model](#)

2. TCP/IP

TCP/IP (Transmission Control Protocol/Internet Protocol) – это набор протоколов, которые управляют перемещением данных от одного устройства к другому. Помимо двух основных одноименных протоколов, TCP/IP также содержит и другие протоколы. IP – это протокол сетевого уровня, который предоставляет сервисы маршрутизации датаграмм. Основной задачей IP является поддержка межсетевой адресации и маршрутизации пакетов. Это протокол, работающий без установления соединения, который оборачивает пришедшие с транспортного уровня данные. Протокол IP адресует датаграммы, используя IP-адреса отправителя и получателя. Протоколы в рамках набора TCP/IP совместно работают для того, чтобы разбить приходящие с прикладного уровня данные на части, которые могут быть переданы по сети. Они работают с другими протоколами для передачи данных компьютеру получателя, а затем, чтобы пересобрать данные в обратном порядке в форму, которая понятна прикладному уровню и которую он может обработать.

IP – это протокол, работающий без установления соединения, который обеспечивает возможности маршрутизации и адресации для каждого пакета данных. Данные, IP и сетевое взаимодействие можно сравнить с взаимодействием между письмом и почтовой системой:

- Данные = Письмо
- IP = Адресованный конверт
- Сеть = Почтовая система

Сообщение – это письмо, которое запечатано в конверт и адресовано IP, сеть и ее службы позволяют отправить сообщение от отправителя получателю, также как почтовая система.

Двумя основными протоколами, работающими на транспортном уровне, являются TCP и UDP. **TCP** – это надежный протокол *с предварительным установлением соединения* (connection-oriented protocol). Это означает, что он убеждается в доставке сообщения компьютеру получателя. Если пакет был потерян в процессе передачи, TCP способен выявить это и повторно отправить потерянный или поврежденный пакет. TCP также поддерживает упорядочивание пакетов (гарантируя получение всех и каждого пакета), контроль потока и перегрузок канала связи, выявление и исправление ошибок. **UDP**, с другой стороны, обеспечивает лучшую скорость и *не использует предварительное установление соединений* (connectionless protocol). Он не поддерживает упорядочивание пакетов, не выявляет перегрузки канала связи, не требует подтверждения получения каждого пакета получателем.

2.1. TCP

TCP называют протоколом с предварительным установлением соединения, поскольку он перед реальной отправкой данных выполняет процедуру «рукопожатия» между двумя системами, собирающимися взаимодействовать. После успешного завершения «рукопожатия» устанавливается виртуальное соединение между двумя системами. UDP – это протокол, не использующий предварительное установление соединения, поскольку он не выполняет этих шагов. Вместо этого UDP отправляет сообщения компьютеру получателя без предварительного контакта с ним, не проверяя, были ли они получены или потеряны. На рисунке 5-10 показаны различия между протоколом с предварительным установлением соединения и протоколом без предварительного установления соединения.

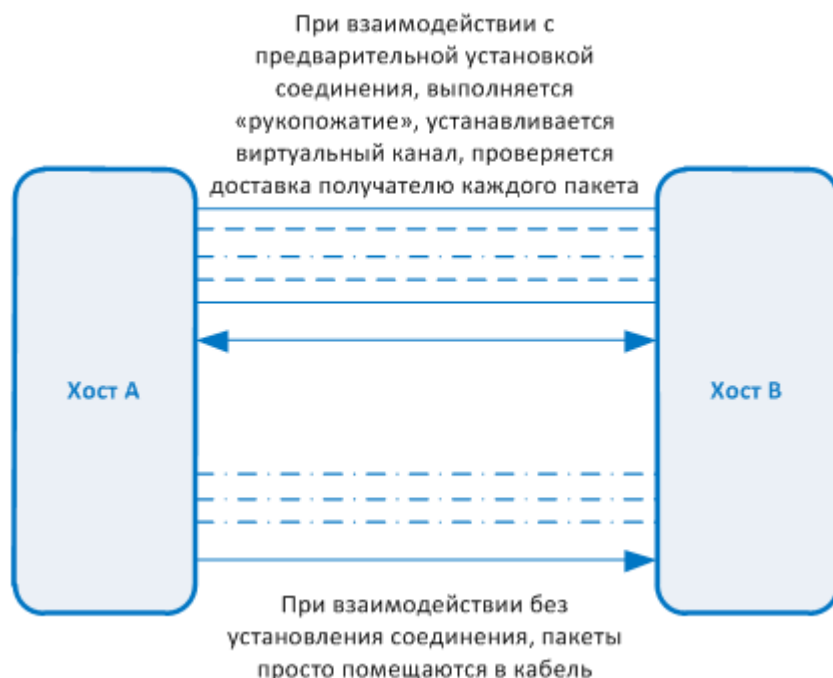


Рисунок 5-10. Функции протокола с установлением соединения по сравнению с функциями протокола без установления соединения

И UDP, и TCP находятся на транспортном уровне, разработчики могут выбирать какой из них использовать в своих приложениях. Часто в качестве транспортного протокола

выбирают TCP, т.к. он обеспечивает надежность и контролирует доставку пакетов. Например, SMTP, используемый для передачи сообщений электронной почты, применяет TCP, поскольку ему нужно убедиться, что данные были доставлены. TCP обеспечивает дуплексный режим, надежный механизм связи, а в случае потери или повреждения любого пакета он его отправляет повторно. Однако TCP больше нагружает систему.

Если программист знает, что потеря данных в процессе передачи не нанесет вреда работе приложения, он может выбрать протокол UDP, поскольку он быстрее и требует меньше ресурсов. Например, UDP является лучшим выбором для отправки сервером информации о своем состоянии всем прослушивающим узлам в сети. Узлу не будет нанесено никакого вреда, если по какой-то причине он не получит эту информацию, т.к. она отправляется каждые 30 минут.

UDP и TCP являются транспортными протоколами, которые используются приложениями для обмена данными по сети. Оба они используют порты для взаимодействия с вышестоящими уровнями OSI и разделения различных взаимодействий, происходящих одновременно. Порты также являются механизмом, позволяющим определить к какой службе обращается другой компьютер. При формировании сообщения TCP или UDP, порты отправителя и получателя указываются в заголовке пакета вместе с адресами отправителя и получателя; адрес и номер порта создают *сокет*. Таким образом, пакеты знают, куда им нужно идти (по адресу) и с какой службой или протоколом взаимодействовать (по номеру порта). IP-адрес работает как дверь к компьютеру, а порт работает как дверь к определенному протоколу или службе. Для правильного взаимодействия пакет должен знать нужные ему двери. На рисунке 5-11 показано, как пакеты взаимодействуют с приложениями и службами через порты.

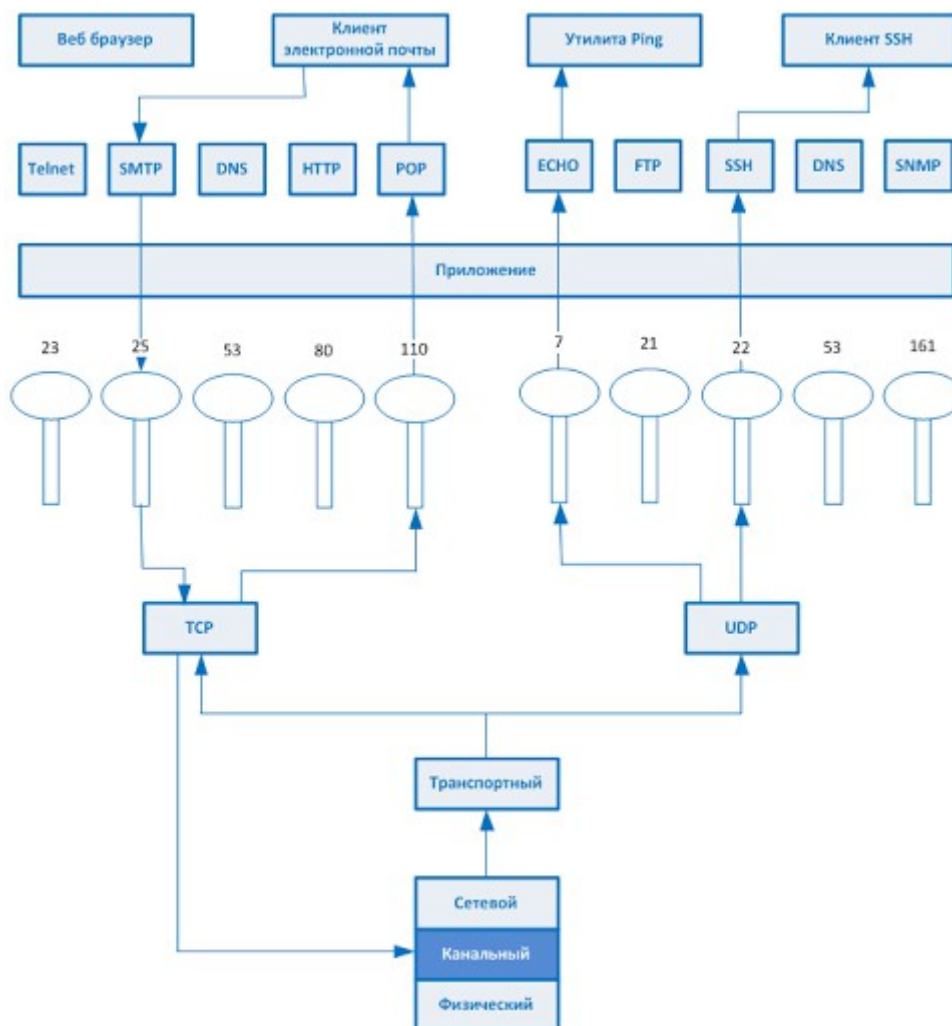


Рисунок 5-11. Пакеты могут взаимодействовать с протоколами вышестоящего уровня и службами посредством портов

Общеизвестные порты. Номера портов до 1023 (0-1023) называются общеизвестными портами (следует отметить, что порты 0-1023 могут использоваться только привилегированными системными процессами или root'ом). Почти на любом компьютере в мире к одним и тем же портам в этом диапазоне «привязаны» одни и те же протоколы. Поскольку они общеизвестны, все используют этот стандартный подход. Это означает, что почти на каждом компьютере протокол SMTP работает на 25-м порту, FTP – на 21-м, а 80-й порт выделен для HTTP и т.д. Эта «привязка» портов с номерами от 0 до 1023 к определенным протоколам является стандартом (де факто), который мы все используем. Однако, поскольку это не является строго обязательным стандартом, администраторы при необходимости могут изменить «привязку» протоколов к портам.

Ниже приведены некоторые наиболее часто используемые протоколы и порты, к которым они «привязаны»:

- Telnet – порт 23
- SMTP – порт 25
- HTTP – порт 80
- SNMP – порты 161 и 162
- FTP – порты 21 и 20

Различие между TCP и UDP можно увидеть также в формате их сообщений. Поскольку TCP предоставляет больше сервисов, чем UDP, он должен включать больше информации в заголовки своих пакетов, как это показано на рисунке 5-12. В таблице 5-1 приведены основные различия между TCP и UDP.



Рисунок 5-12. TCP передает больше информации в заголовках своих пакетов, поскольку он предоставляет больше сервисов, чем UDP

Сервис	TCP	UDP
Надежность	Гарантирует, что пакет получен получателем, отправляет ACK при получении пакета. Это протокол с надежной доставкой.	Не отправляет ACK, не гарантирует получение пакета получателем и не является протоколом с надежной доставкой.
Соединение	Предварительно устанавливает соединение. Он выполняет «рукопожатие», создавая виртуальное соединение с компьютером получателя.	Без предварительной установки соединения. Не выполняет «рукопожатия», не устанавливает виртуальное соединение.
Упорядочивание пакетов	Использует порядковые номера в заголовках, чтобы убедиться в доставке каждого пакета.	Не использует порядковые номера.
Контроль перегрузки канала связи	Компьютер-получатель может сообщить отправителю о перегрузке и необходимости снизить скорость передачи.	Компьютер-получатель не взаимодействует с отправителем в отношении управления потоком.
Применение	Используется, когда требуется надежная доставка.	Используется, когда надежная доставка не требуется, например, для потокового видео, рассылки сообщений о состоянии.
Скорость и служебные данные	Использует больше ресурсов. Скорость TCP ниже, чем у UDP.	Использует меньше ресурсов. Скорость UDP выше, чем у TCP.

Таблица 5-1. Основные различия между TCP и UDP

«Рукопожатие» TCP

TCP должен установить виртуальное соединение между двумя хостами перед отправкой данных. Это означает, что два хоста должны договориться об определенных параметрах, потоке данных, кадрировании, выявлению ошибок и других опциях. Эти параметры настраиваются на этапе «рукопожатия» (handshaking), показанного на рисунке 5-13.

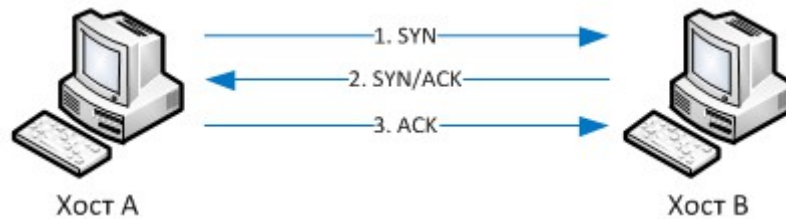


Рисунок 5-13. Трехступенчатый процесс «рукопожатия» TCP

Иницилирующий взаимодействие хост отправляет получателю пакет синхронизации (SYN – synchronous packet). Получатель отвечает на него, отправляя пакет SYN/ACK. Это означает что-то типа «я получил твой запрос и готов к взаимодействию с тобой». Хост отправителя отвечает на это пакетом подтверждения (ACK – acknowledgment packet), что означает «я получил твое подтверждение, давай начнем передачу данных». На этом процедура «рукопожатия» завершается, виртуальное соединение считается установленным и начинается реальная передача данных. Установленное соединение работает в дуплексном режиме, т.е. возможна одновременная передача данных в обоих направлениях.

Структуры данных

Как было сказано ранее, сообщение обычно формируется программным обеспечением и передается им на прикладной уровень. Затем оно отправляется вниз по стеку протоколов. Каждый протокол на каждом уровне добавляет собственную информацию к сообщению и передает его вниз на следующий уровень. Эта концепция обычно называется *инкапсуляцией*. Сообщение, перемещаясь вниз по стеку, проходит своего рода эволюцию, каждый этап которой имеет свое название, указывающее, что на нем происходит. Когда приложение форматирует данные для отправки по сети, эти данные называют *поток* (stream), либо *сообщением* (message). Они отправляются на транспортный уровень, где TCP колдует над данными. Там этот набор данных превращается в *сегмент* (segment). Сегмент отправляется на сетевой уровень. Сетевой уровень добавляет информацию для маршрутизации и адресации, после чего набор данных называется *датаграммой* (datagram). Сетевой уровень передает датаграмму на канальный уровень, который «обрамляет» ее заголовком (header) и окончанием (trailer). При этом датаграмма превращается в *кадр* (frame). Рисунок 5-14 иллюстрирует этот процесс.

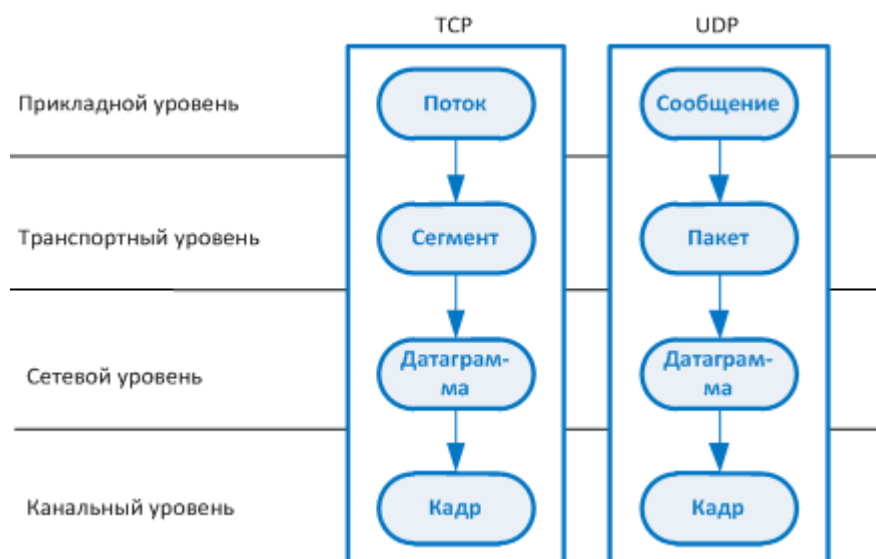


Рисунок 5-14. Данные проходят несколько этапов эволюции, проходя по уровням сетевого стека

2.2. Адресация IP

Каждый узел в сети должен иметь уникальный IP-адрес. В настоящее время в большинстве случаев используется **IP версии 4** (IPv4), но количество устройств в сети уже настолько возросло, что свободные адреса стали стремительно заканчиваться. Чтобы избежать этого, был создан **IP версии 6** (IPv6). Кроме того, IPv6 имеет множество встроенных функций безопасности, которых нет в IPv4.

IPv4 использует 32-битную адресацию, а IPv6 – 128-битную. Поэтому IPv6 предоставляет гораздо больше адресов. Каждый IP-адрес состоит из части, относящейся непосредственно к хосту, и части, относящейся к сети. Адреса группируются в *классы*, а затем в *подсети*. Маски подсетей делят адреса на группы, которые определяют подсети в сети. Классы адресов IPv4 приведены ниже:

Класс А	0.0.0.0 – 127.255.255.255	Первый байт – это сетевая часть, оставшиеся три – часть, относящаяся к хосту
Класс В	128.0.0.0 – 191.255.255.255	Первые два байта – это сетевая часть, оставшиеся два – часть, относящаяся к хосту
Класс С	192.0.0.0 – 223.255.255.255	Первые три байта – это сетевая часть, оставшийся байт относится к хосту
Класс D	224.0.0.0 – 239.255.255.255	Используется для многоадресной (multicast) передачи
Класс E	240.0.0.0 to 255.255.255.255	Зарезервирован для исследований

Все подключенные к сети компании узлы могут иметь различные адреса хоста, но у них должен быть общий адрес сети. Адрес хоста идентифицирует каждый узел в отдельности, тогда как адрес сети идентифицирует сеть, к которой они подключены – поэтому адрес сети должен быть одинаковым для каждого из них. Любой трафик, предназначенный для узлов в этой сети, будет отправлен на адрес этой сети.

Подсеть (subnet) создается из хостовой части IP-адреса для присвоения «под» сети. Это позволяет нам дополнительно разбить хостовую часть адреса на две или более логические группы, как показано на Рисунке 5-15. Сеть может быть логически разделена для упрощения администрирования, повышения производительности сети, а также для безопасности. Разделение компьютеров (узлов) по различным подсетям позволит лучше управлять ими. Маска подсети служит неким барьером, на логическом уровне разделяющим маленькие сети внутри большой сети.

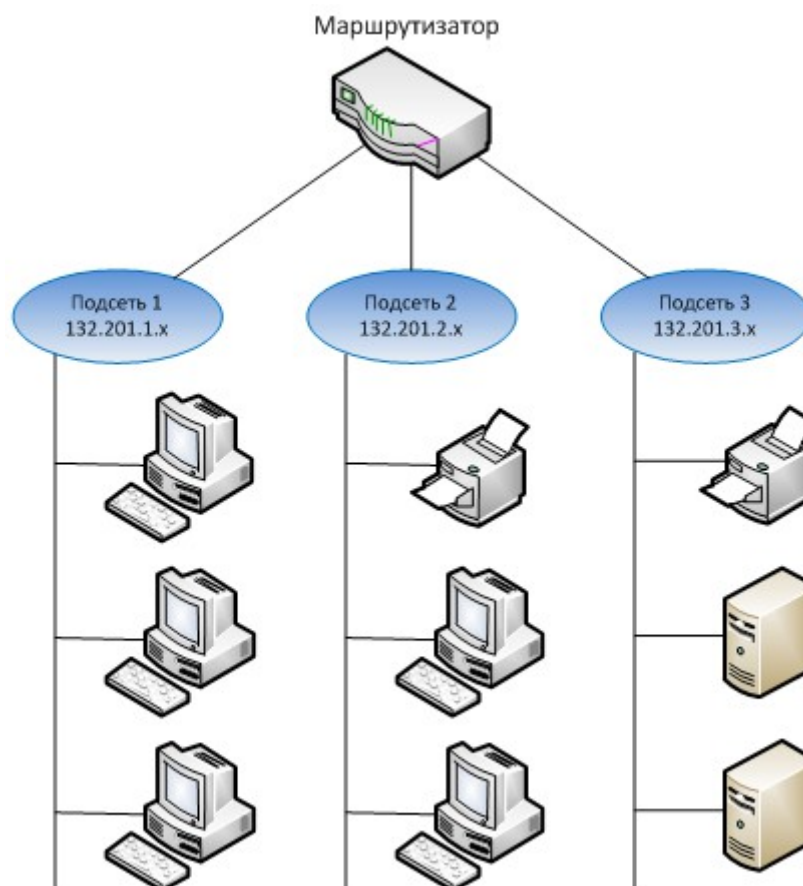


Рисунок 5-15. Подсети создают логические сегменты

Разделение на подсети (subnetting) позволяет разделять широкие диапазоны IP-адресов на более узкие, логические и четкие сетевые сегменты. Рассмотрим компанию с несколькими подразделениями, такими как ИТ, бухгалтерия, отдел кадров и т.д. Создание подсетей для каждого подразделения разделяет сеть на логические части, что позволяет направлять трафик напрямую получателям, не распространяя данные по всей сети. Это резко снижает загруженность сети, уменьшая вероятность ее перегрузки, в частности, большим количеством широковещательных (broadcast) пакетов. Реализация политик безопасности сети также гораздо более эффективна для логически категоризованных подсетей с отдельными периметрами, чем для большой, беспорядочной и сложной сети.

Разделение на подсети особенно выгодно для поддержания небольшого размера таблиц маршрутизации, поскольку при этом внешние маршрутизаторы могут напрямую направлять данные в реальный сетевой сегмент, не беспокоясь о внутренней архитектуре сети и доставке данных к отдельным хостам. Эта работа будет выполняться внутренними маршрутизаторами, которые могут определять конкретные хосты в разделенной на подсети среде, освобождая внешние маршрутизаторы от хлопот, связанных с необходимостью анализа всех 32 бит IP-адреса, – в такой среде внешним маршрутизаторам достаточно смотреть только на биты маски, указывающие на подсеть.

ПРИМЕЧАНИЕ. Чтобы полностью понять вопрос разделения сети на подсети, вам нужно детально разобраться с тем, как работают IP-адреса на двоичном уровне. Более детальную информацию об этом вы можете получить на странице <http://compnetworking.about.com/od/workingwithipaddresses/a/subnetmask.htm>

Использование традиционной маски подсети означает, что используются классические или классифицированные IP-адреса. Если компании нужно создать подсети, не соответствующие этим традиционным размерам, она может использовать неклассифицированные IP-адреса. Это просто означает, что будет использоваться другая маска подсети для разделения в IP-адресе частей, относящихся к хосту и к сети. Когда стало ясно, что доступные адреса в сети

стремительно заканчиваются, поскольку к сети Интернет подключается все больше и больше людей и компаний, была создана безклассовая междоменная маршрутизация (CIDR – classless interdomain routing). Диапазон адресов Класса В обычно слишком велик для большинства компаний, а Класс С, наоборот, слишком мал. В таких случаях CIDR обеспечивает гибкость, позволяя при необходимости увеличить или уменьшить размеры классов.

ПРИМЕЧАНИЕ. Для лучшего понимания работы CIDR посетите страницу www.tcpipguide.com/free/t_IPClasslessAddressingClasslessInterDomainRoutingCI.htm

Хотя каждый узел имеет свой IP-адрес, люди чаще используют имя хоста, чем его адрес. Имена хостов (например, www.logicalsecurity.com) проще запомнить, чем их IP-адреса, такие как 10.13.84.4. Однако использование этих двух систем обозначения требует установления связи между ними (между именем хоста и его адресом), поскольку компьютеры понимают только цифровые схемы. Этот процесс разбирается в разделе «DNS».

ПРИМЕЧАНИЕ. IP обеспечивает адресацию, фрагментацию и таймауты пакетов. Чтобы гарантировать, что пакеты не будут ходить по сети вечно, IP использует значение TTL (время жизни пакета), которое уменьшается каждый раз, когда сообщение проходит через маршрутизатор. IP также может предоставлять функциональность ToS (Type of Service – тип обслуживания), которая обеспечивает приоритезацию пакетов чувствительных к задержкам сервисов.

2.3. IPv6

IPv6, также известный как **IP следующего поколения** (IPng), не только имеет более широкое адресное пространство, но и множество возможностей, которых нет в IPv4. Хотя все эти новые возможности IPv6 выходят за рамки этой книги, мы немного поговорим о некоторых из них, т.к. за IPv6 будущее. IPv6 позволяет использовать ограничивать области адресов (scoped addresses), что дает администратору возможность ограничить определенные адреса для файловых серверов или общих принтеров, например. В IPv6 IPSec встроен в стек протоколов, что обеспечивает безопасную сквозную передачу и аутентификацию. Этот протокол предоставляет возможность автоматической конфигурации, что сильно упрощает администрирование и не требует использования NAT для расширения адресного пространства.

NAT (network address translation – трансляция сетевых адресов) был разработан только потому, что в IPv4 свободные адреса быстро заканчивались. Хотя технология NAT очень полезна, она является причиной дополнительных издержек и многих проблем передачи данных, т.к. она ломает клиент-серверную модель, используемую многими современными приложениями. IPv6 имеет большую гибкость и лучшие возможности маршрутизации, позволяя устанавливать значения приоритетов QoS (Quality of Service – качество обслуживания) для чувствительных к задержкам сервисов. Однако, хотя IPv6 имеет значительно больше преимуществ, чем IPv4, он пока не применяется массово, его принятие индустрией идет очень медленно, что связано в основном с вопросами взаимодействия между IPv4 и IPv6, а также тем, что разработка протокола NAT существенно снизила скорость уменьшения количества свободных адресов.

ПРИМЕЧАНИЕ. NAT рассматривается далее, в разделе «NAT».

В спецификации IPv6 (см. RFC 2460) приведены отличия и преимущества IPv6 над IPv4. Вот некоторые из них:

- В IPv6 увеличен размер IP-адреса с 32 бит до 128 бит для поддержки большего количества уровней иерархии адресов, обеспечения гораздо большего количества адресов для узлов и упрощенной автоконфигурации адресов.
- Масштабируемость многоадресной маршрутизации (multicast routing) улучшена за счет добавления поля «Область» (scope) для групповых адресов (multicast addresses).

Также, определены новые типы адресов, названные *альтернативными адресами* (anycast address), которые используются для отправки пакетов любому узлу из группы узлов.

- Некоторые поля заголовка IPv4 удалены или сделаны опциональными для снижения общих затрат на обработку пакетов и увеличения полосы пропускания за счет уменьшения размера заголовка пакета IPv6. Это показано на Рисунке 5-16.

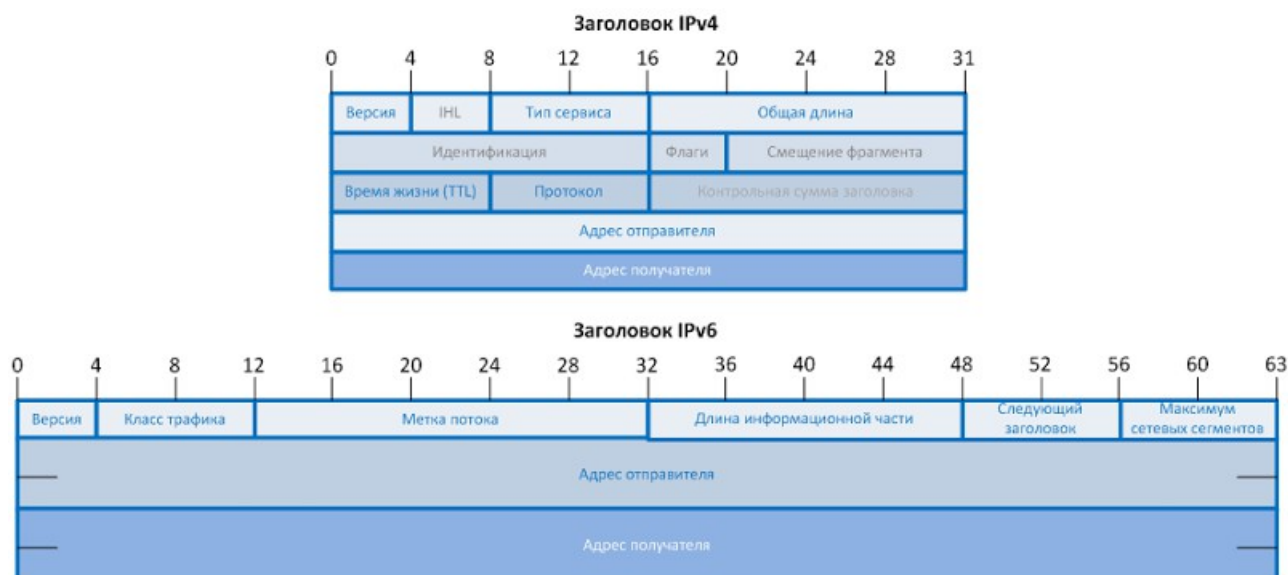


Рисунок 5-16. Заголовки IPv4 и IPv6

- Изменения в способе кодирования информации заголовка пакета IP обеспечивают более эффективную пересылку, менее жесткие ограничения на длину опций, значительно повышают гибкость для введения в будущем новых опций.
- Добавлены новые возможности, позволившие маркировать пакеты, для которых отправителем запрошена специальная обработка, выделяя их из общего потока трафика. Это может быть необходимо, например, для установки нестандартного QoS или для сервиса реального времени.
- В IPv6 также предусмотрены расширения для поддержки аутентификации, обеспечения целостности данных и (опционально) конфиденциальности данных.

IPv4 не обеспечивает таких аспектов безопасности, как аутентификация, целостность и конфиденциальность данных, но как это делает IPv6? Все дело в использовании IP security (IPSec). IPSec – это набор протоколов, который защищает данные, передающиеся по IP-сетям. Хотя использование IPSec доступно и для IPv4, он не интегрирован полностью в сетевой стек IP так, как это сделано в IPv6. Более подробно IPSec рассматривается в Домене 06.

3. Типы передачи

Передача данных может осуществляться различными способами (аналоговыми или цифровыми), использовать различные схемы управления (синхронные или асинхронные), использовать для каждого канала отдельный провод (baseband – однополосная передача), либо размещать в каждом проводе по несколько каналов (broadband – широкополосная передача). Все эти типы передачи и их характеристики будут описаны в следующих разделах.

3.1. Аналоговая и цифровая

Аналоговая передача сигналов – это постоянно изменяющаяся электромагнитная волна,

которая может распространяться по воздуху, воде, витой паре, коаксиальному или оптоволоконному кабелю. С помощью процесса *модуляции*, данные объединяются с сигналом несущей, имеющим определенную частоту. Модуляция сигнала различается по *амплитуде* (высота сигнала) и *частоте* (число волн в определенный период времени), как показано на Рисунке 5-17. Это означает, что данные размещаются на «спине» сигнала несущей частоты. Сигналы несущей частоты предоставляют многие радиостанции, частотные диапазоны и коммуникационные каналы. Каждая радиостанция имеет определенный сигнал несущей и частоту, используемую ей для передачи. Поэтому, например, три различные радиостанции в городе могут предоставлять три различных радиоканала.

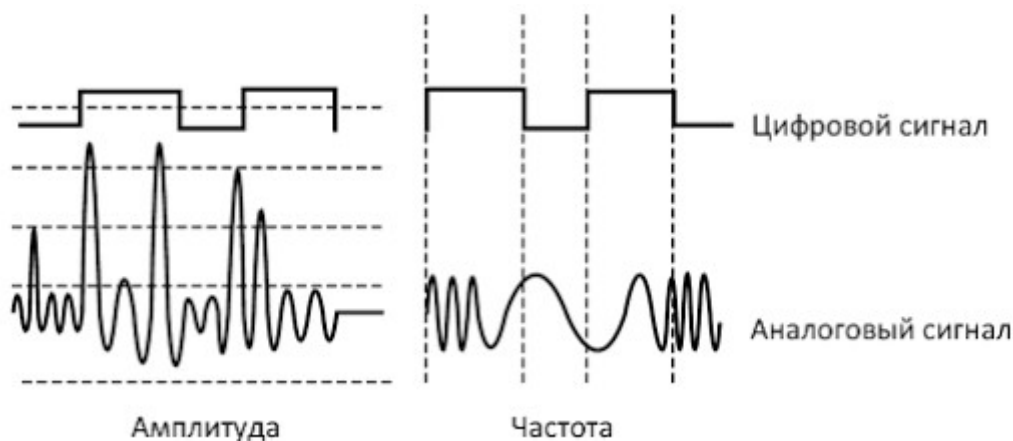


Рисунок 5-17. Аналоговые сигналы измеряются по амплитуде и частоте, тогда как цифровые сигналы представляют собой двоичные цифры в виде электрических импульсов

Внутри компьютеров используются цифровые сигналы для передачи данных от одного компонента к другому. Когда компьютер подключается к телефонной линии посредством коммутируемого (dial-up) соединения, модем (*модулятор/демодулятор*) должен преобразовать эти цифровые данные в аналоговый сигнал, используемый в телефонной линии. Фактически модем модулирует цифровые данные в аналоговый сигнал. Когда данные получены компьютером получателя, он должен преобразовать их обратно в цифровую форму, которую он понимает. **Цифровые сигналы** представляют собой двоичные цифры в виде электрических импульсов. Каждый отдельный импульс – это элемент сигнала, представляющий собой 1 или 0. **Полоса пропускания** (bandwidth) при цифровой передаче – это число электрических импульсов, которые могут быть переданы через соединение в секунду, а эти электрические импульсы передают отдельные биты информации.

Цифровые сигналы более надежны, чем аналоговые, при передаче на большое расстояние и обеспечивают более четкий и эффективный метод передачи сигнала, т.к. гораздо проще сравнить наличие напряжения (1) с его отсутствием (0), чем сравнивать волны для интерпретации аналогового сигнала. Цифровой сигнал проще очистить от шумов и отраженных сигналов. Аналоговый сигнал гораздо сложнее отделить от фонового шума, поскольку амплитуда и частота волн постепенно теряют форму. Это связано с тем, что аналоговый сигнал может иметь бесконечное число значений и состояний, тогда как цифровой сигнал находится в дискретных состояниях. Цифровой сигнал – это волны прямоугольной формы, которые не имеют множества возможных значений, различной амплитуды и частоты как в аналоговом сигнале. Цифровые системы более совершенны, чем аналоговые, они могут передавать больше телефонных звонков и данных по одной и той же линии, обеспечивая при этом лучшее качество при передаче на большие расстояния.

Раньше передача голоса и данных осуществлялась в основном посредством аналоговых сигналов по телекоммуникационным каналам, но сейчас большинство коммуникаций оцифровано. Телефонные компании оцифровывают телефонные звонки, многие корпоративные телефонные системы являются цифровыми. Радиоволны продолжают

использоваться в радиостанциях, радиолюбительской связи, частных домах и центральных офисах телефонных компаний. Эта часть телекоммуникационных сетей называется *абонентской линией* (local loop) или «*последней милей*» (last mile).

3.2. Асинхронная и синхронная

Два устройства могут взаимодействовать синхронным или асинхронным образом в зависимости от типа связи и наличия какой-либо синхронизации двух этих устройств.

Асинхронная связь применяется тогда, когда два устройства никак не синхронизированы. Отправитель может отправлять данные в любое время, а принимающая сторона должна быть все время готова к приему. **Синхронная связь** происходит между двумя синхронизированными (обычно с помощью часового механизма) устройствами.

Обычно для передачи больших объемов данных используют синхронную связь, а для малых объемов – асинхронную. Однако такое решение обычно принимается не в момент передачи данных, а гораздо раньше, при разработке системы. Системы, предназначенные для частой передачи больших объемов данных, заранее разрабатываются и настраиваются на использование синхронной связи, а системы, предназначенные для передачи небольших объемов данных, разрабатываются как асинхронные.

Примером асинхронного соединения может быть связь между терминалом и терминальным сервером. Пользователь использует на своем компьютере терминальное программное обеспечение для доступа к удаленному рабочему столу, который ему предоставляет сервер. Пользователь просто видит рабочий стол на своем компьютере, а в действительности он работает на терминальном сервере. При этом каждое нажатие кнопки «мыши» или клавиши на клавиатуре передается по сети терминальному серверу, а терминальный сервер выполняет соответствующую команду. Результат отображается на рабочем столе, изображение которого передается пользователю. Пользователь видит все происходящее так, как будто это его компьютер выполнил команду, хотя фактически она была выполнена терминальным сервером, который может находиться на другом этаже или вообще в другом здании. При использовании этой технологии обычно передаются небольшие объемы данных за единицу времени, поэтому она использует асинхронную передачу данных.

Модемы также используют асинхронную передачу данных. Поскольку данные могут передаваться в любое время и иметь любую длину, должны использоваться разделители начала и конца, сообщающие принимающей стороне, откуда начинать процесс обработки и где его заканчивать. Каждый символ, который в действительности представляет собой последовательность нулей и единиц, имеет бит начала и бит конца, прикрепленные соответственно до и после битов самого символа. Это несколько увеличивает передаваемый объем информации, но это необходимо при асинхронной передаче.

Синхронная связь, с другой стороны, передает данные как поток битов, не разделяя их на кадры со стартовыми и стоповыми битами. При этом между двумя системами должна быть обеспечена синхронизация, например, путем использования одинакового часового механизма или сигнала, который может быть закодирован в потоке данных, чтобы получатель мог синхронизироваться с отправителем сообщения. Синхронизация должна быть обеспечена до отправки первого сообщения. Передающая система может отправить цифровой импульс времени принимающей системе, что можно перевести как "мы начинаем сейчас и будем работать по этой схеме синхронизации".

3.3. Однополосная и широкополосная

При *однополосной* (baseband) передаче весь коммуникационный канал используется для одной этой передачи, тогда как при *широкополосной* (broadband) передаче коммуникационный канал делится на отдельные независимые каналы, которые могут передавать одновременно различные типы данных. Однополосная передача позволяет передавать только один сигнал в единицу времени, тогда как широкополосная – много

сигналов по различным каналам. Например, коаксиальный телевизионный кабель (CATV – Coaxial cable TV) использует широкополосную технологию, позволяющую передавать множество телевизионных каналов по одному проводу. Причем посредством того же кабеля можно дополнительно предоставить доступ в Интернет домашним пользователям, но эти данные будут передаваться на другой частоте по сравнению с телевизионными каналами. Ethernet – это однополосная технология, в которой весь провод используется для передачи только одного канала.

Широкополосную передачу используют и многие другие технологии, но общим правилом для них является обеспечение передачи данных на скорости выше 56 кбит/с, которую обеспечивает стандартный модем, работающий на обычном коммутируемом соединении. Широкополосные коммуникации предоставляют каналы для передачи данных и могут использоваться множеством пользователей. Типами широкополосных систем связи, доступных сегодня, являются выделенные линии связи (T1, T3), широкополосный ISDN, ATM, DSL (Digital Subscriber Line - цифровая абонентская линия), широкополосные беспроводные каналы и CATV.

4. Организация локальных вычислительных сетей

Ниже представлены четыре основные причины для организации сети:

- Чтобы обеспечить взаимодействие между компьютерами
- Чтобы совместно использовать информацию
- Чтобы совместно использовать ресурсы
- Чтобы обеспечить централизованное администрирование

Большинству пользователей сети нужно использовать однотипные ресурсы, такие как информация, принтеры, файловые серверы, плоттеры, факсы, доступ в Интернет и т.д. Почему бы не объединить все эти ресурсы и не сделать доступными всем? Прекрасная идея, для ее реализации нам нужно *организовать сеть*!

Организация сети может дать прекрасные возможности за короткое время. В начале компьютерной эры использовались мейнфреймы. Они были изолированы в серверных помещениях и доступ к ним осуществлялся посредством простых терминалов. Однако это еще не сеть в чистом виде. В конце 1960-х – начале 1970-х годов исследователи нашли способ объединения всех мейнфреймов и Unix-систем для обеспечения их взаимодействия. Это были первые шаги Интернета.

Микрокомпьютеры использовались во многих офисах и на рабочих местах. Терминалы стали немного «умнее» и полезнее при совместном использовании офисных ресурсов. А затем был разработан Ethernet, позволивший создать настоящую сеть.

ПРИМЕЧАНИЕ. Значительной частью организации сети являются вопросы идентификации и аутентификации, рассмотренные в Домене 02. Однако, сейчас важно отметить, что аутентификация самого по себе узла сети не может быть использована для установления доверительных отношений между пользователями сети. В распределенной сети тема доверия является основной проблемой безопасности.

Ссылки по теме:

- IEEE Standards Working Group
- Introduction to Networking and Data Communications, by Eugene Blanchard (2001)
- “Ultimate Guide to Networking: Part One,” by Michael Furdyk, HardwareCentral tutorial (June 3, 2005)
- “How Analog and Digital Recording Works: Analog Wave,” by Marshall Brain, HowStuffWorks

4.1. Топология сети

Физическое размещение компьютеров и устройств называется **сетевой топологией**.

Топология указывает на способ, которым *физически* соединена сеть, показывает размещение ресурсов и систем. Существует разница между физической топологией сети и логической топологией. Сеть может быть сконфигурирована физически как звезда, а логически – как кольцо, использующее технологию Token Ring.

Выбор лучшей топологии для конкретной сети зависит от таких вещей, как предполагаемый порядок взаимодействия узлов, используемые протоколы, типы приложений, надежность, расширяемость, физическое размещение в здании, а также от уже внедренных технологий. Неверная топология (или комбинация топологий) может негативно сказаться на производительности сети, ее продуктивности и возможностях расширения.

В этом разделе описаны основные типы сетевых топологий. Большинство сетей значительно более сложны и реализованы с использованием комбинации топологий.

Топология «кольцо»

Топология «кольцо» (ring topology) – это последовательное соединение устройств однонаправленными линиями связи, как показано на Рисунке 5-18. Эти связи образуют замкнутое кольцо, не имеющее подключения к центральной системе (имеющейся в топологии «звезда»). В физическом кольце каждый узел зависит от предшествующих узлов. В простой системе, в случае неисправности одной системы, она окажет негативное влияние на все остальные системы, поскольку все они взаимосвязаны. Сегодня большинство сетей обладают избыточностью или другими механизмами, которые могут защитить сеть в случае неисправности одной рабочей станции, но некоторые неудобства при этом вероятно все равно возникнут.

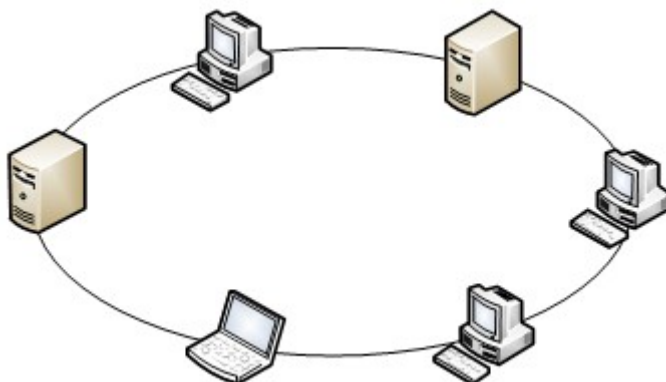


Рисунок 5-18. Топология «кольцо» образует замкнутое соединение

Топология «шина»

В простой **топологии «шина»** (bus topology), единственный кабель проходит по всей длине сети. Узлы подключаются к сети «в разрыв» кабеля. Данные передаются по всей длине кабеля, и каждый узел может просмотреть любой передаваемых пакетов. Каждый узел решает, принять ему пакет или проигнорировать его, ориентируясь на указанный в пакете адрес компьютера-получателя.

Существует два основных типа топологии «шина»: линейная и древовидная. **Линейная топология «шина»** имеет один кабель, к которому подсоединены все узлы. **Древовидная топология «шина»** имеет отдельные ответвления от единого кабеля, к каждому ответвлению может быть подключено множество узлов.

В простой реализации топологии «шина», если одна рабочая станция выходит из строя, она оказывает негативное влияние на другие системы, т.к. они в определенной степени взаимозависимы. Подключение всех узлов к одному кабелю – это единая точка отказа.

Традиционно Ethernet использует топологию «звезда».

Топология «звезда»

В **топологии «звезда»** (star topology) все узлы подключаются к центральному устройству, такому как коммутатор (switch). Каждый узел имеет выделенное подключение к центральному устройству. Центральное устройство должно обеспечивать достаточную пропускную способность, чтобы не стать «бутылочным горлышком» для всей сети. Использование центрального устройства потенциально является единой точкой отказа, поэтому должна быть обеспечена некоторая избыточность. Коммутаторы могут быть настроены в плоской или иерархической реализации, которую могут использовать крупные компании.

Когда одна рабочая станция выходит из строя в топологии «звезда», это не оказывает воздействия на другие системы, как в топологиях «шина» или «кольцо». В топологии «звезда» каждая система независима от других, но она зависит от центрального устройства. Эта топология обычно требует меньше проводов, чем другие топологии, и, как следствие, существует меньше шансов разрыва провода, а задача выявления проблем существенно упрощается.

Не многие сети используют в чистом виде топологию линейной «шины» или «кольцо» в локальной сети. Топология «кольцо» может быть использована для магистральной сети, но большинство локальных вычислительных сетей (LAN) создается на базе топологии «звезда», поскольку это повышает отказоустойчивость сети и позволяет ей не зависеть от проблем отдельных узлов. Помните, что существует разница между физической топологией и методами доступа к среде передачи информации. Даже если сеть построена как Token Ring или Ethernet, это говорит только о том, как подключен к среде передачи информации каждый узел этой сети и как проходит трафик. Хотя Token Ring обычно работает через «кольцо», а Ethernet подразумевает реализацию «шины», эти термины относятся только к логической организации сети, реализующейся на канальном уровне. Если при этом физически проще организовать «звезду», то так и делают.

Полносвязная топология

В **полносвязной топологии** (mesh topology) все системы и ресурсы подключены друг к другу иными способами по сравнению с вышеуказанными топологиями, как показано на рисунке 5-19. Эта схема обычно представляет собой сеть связанных друг с другом маршрутизаторов и коммутаторов, обеспечивающих множественные маршруты передачи данных между всеми узлами в сети. При полной реализации полносвязной топологии (full mesh), каждый узел напрямую соединен с каждым другим из других узлов, что обеспечивает наивысшую степень отказоустойчивости. При частичной реализации полносвязной топологии (partial mesh), не все узлы связаны напрямую. Интернет – это пример сети с частичной реализацией полносвязной топологии.

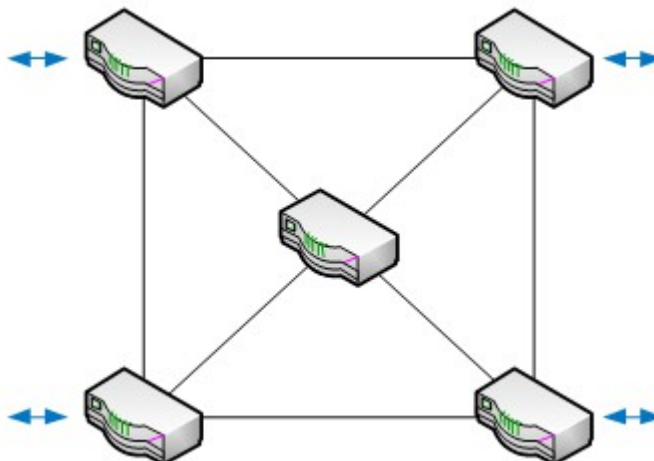


Рисунок 5-19. В полносвязной топологии все узлы соединены друг с другом, что обеспечивает наличие избыточных связей

Резюме по различным сетевым топологиям и их наиболее важные характеристики представлены в таблице 5-2.

Топология	Характеристики	Проблемы	Используемые технологии
Шина	Использует единый линейный кабель для подключения всех компьютеров. Весь трафик передается по всей длине кабеля и может быть просмотрен любым подключенным компьютером	Если на одном из компьютеров возникает проблема, она оказывает влияние на окружающие компьютеры, подключенные к тому же кабелю	Ethernet
Кольцо	Все компьютеры подключены к двунаправленной линии передачи, при этом кабель образует замкнутое кольцо	Если на одном из компьютеров возникает проблема, она оказывает негативное влияние на окружающие компьютеры, подключенные к тому же кольцу	FDDI
Звезда	Все компьютеры подключены к центральному устройству, что обеспечивает большую устойчивость сети	Центральное устройство является единой точкой отказа	Логическая шина (Ethernet) и кольцевые топологии (Token ring)
Дерево	Топология шина, которая вместо одного линейного кабеля имеет несколько ответвлений		Ethernet
Полносвязная	Компьютеры подключены друг к другу, что обеспечивает отказоустойчивость	Требуются повышенные затраты на прокладку кабелей и огромные усилия при поиске неисправности в кабеле	Internet

Таблица 5-2. Резюме по сетевым топологиям

Независимо от используемой топологии, большинство сетей LAN имеет магистраль (backbone), являющуюся комбинацией кабелей и протоколов, которая связывает отдельные сетевые сегменты. Магистраль работает на более высокой скорости, чем отдельные сетевые сегменты, что позволяет быстро передавать данные из одной сети в другую. В то время как для сетевых сегментов лучше использовать UTP и Ethernet, для магистрали лучше подходит FDDI или Fast Ethernet. В качестве аналогии можно привести пример городских улиц и автомобильных магистралей. На улицах (в сетевых сегментах) машины (данные) движутся медленно, но улицы соединены с магистралями, которые позволяют машинам быстро перемещаться из одного места в другое. Точно также магистраль позволяет данным быстро перемещаться на большие расстояния.

ПРИМЕЧАНИЕ. При использовании топологии «кольцо» или «шина» все узлы между системами отправителя и получателя имеют доступ к передаваемым данным. Это упрощает для атакующего задачу получения потенциально критичных данных.

4.2. Технологии доступа к среде LAN

LAN – это сеть, которая предоставляет общие коммуникации и ресурсы на относительно небольшой площади. Различия между LAN и WAN определяются физической средой, протоколами инкапсуляции и функциональностью. Например, LAN может использовать кабели 10Base-T, протоколы IPX/SPX и позволять взаимодействовать пользователям, находящимся в пределах здания. WAN, в свою очередь, может использовать оптоволоконные кабели, протокол L2TP и может позволять пользователям одного здания взаимодействовать с пользователями другого здания или даже другого штата (или страны). WAN соединяет сети LAN на больших расстояниях. Наиболее существенные отличия между этими двумя технологиями находятся на канальном уровне.

Вопрос: Говорят, что LAN охватывает относительно небольшую площадь. При каких размерах сеть перестают быть LAN?

Ответ: Когда две отдельные сети LAN соединены маршрутизатором, в результате образуется объединенная сеть (internetwork), которая не является большой LAN. Каждая отдельная LAN

имеет собственную схему адресации, широковещательный домен (broadcast domain) и коммуникационные механизмы. Если две сети LAN соединены с помощью других технологий канального уровня, таких как Frame Relay или X.25, они образуют WAN.

Термин «локальная» в контексте LAN означает не столько географическую область, сколько ограничения LAN с точки зрения общей среды передачи данных, количества подключенных к ней устройств и компьютеров, скорости передачи данных, используемых типов кабелей и устройств. Если сетевой администратор строит очень большую LAN, предпочтительнее организовать ее в виде нескольких LAN, т.к. большой объем трафика нанесет удар по производительности, либо кабели будут слишком длинными и скажется фактор *затухания сигнала* (attenuation). Сеть, в которой установлено слишком много узлов, маршрутизаторов, мостов, коммутаторов может быть очень сложна – в особенности с точки зрения администрирования, что станет открытой дверью для ошибок, конфликтов и «дыр» в безопасности. Сетевой администратор должен следовать спецификациям используемой им технологии, и когда он достигнет предела, ему следует подумать о реализации двух или более небольших LAN вместо одной большой LAN. Сети LAN определяет их физическая топология, технологии канального уровня, протоколы и используемые устройства. Об этом мы поговорим в следующих разделах.

Ссылки по теме:

- IEEE LAN/MAN Standards Committee
- Internetworking Technology Handbook, Chapter 2, “Introduction to LAN Protocols,” Cisco Systems, Inc.

Ethernet

Ethernet – это сетевая технология (LAN-sharing), позволяющая нескольким устройствам взаимодействовать в рамках одной сети. Ethernet обычно использует топологию «звезда» или «шина». Если используется топология линейной шины, все устройства подключаются к одному кабелю. Если используется топология «звезда», каждое устройство кабелем соединяется с центральным устройством (например, с коммутатором). Ethernet был разработан в 1970-х годах и стал доступен для применения в бизнесе в 1980 году. Он был назван стандартом IEEE 802.3.

В своей короткой Ethernet истории прошел эволюцию с реализации на коаксиальном кабеле, работающем на скорости 10 Мб/с, до 5-й категории витой пары, работающей на скоростях 100 Мб/с, 1 Гб/с и даже 10 Гб/с.

Ethernet определяется следующими характеристиками:

- Общая среда (все устройства используют среду поочередно, возможно возникновение коллизий)
- Использует широковещательные (broadcast) и коллизионные (collision) домены
- Использует метод множественного доступа с контролем несущей и обнаружением коллизий (CSMA/CD – Carrier sense multiple access with collision detection)
- Поддерживает полный дуплекс при реализации на витой паре
- Может использовать среду с коаксиальным кабелем или витой парой
- Определен стандартом IEEE 802.3

Ethernet определяет, каким образом компьютеры совместно используют общую сеть и как они обрабатывают коллизии, а также вопросы целостности данных, механизмы коммуникаций, управление передачей. Это обычные характеристики Ethernet, но кроме того Ethernet поддерживает множество типов кабельных схем и скоростей передачи. Существует несколько типов реализации Ethernet, приведенных в таблице 5-3. В следующих разделах

будут обсуждаться реализации 10Base2, 10Base5 и 10Base-T, которые используются чаще всего.

Тип Ethernet	Тип кабеля	Скорость
10Base2, ThinNet	Коаксиальный	10 Мбит/с
10Base5, ThickNet	Коаксиальный	10 Мбит/с
10Base-T	Витая пара (UTP)	10 Мбит/с
10Base-TX, Fast Ethernet	Витая пара (UTP)	100 Мбит/с
100Base-T, Gigabit Ethernet	Витая пара (UTP)	1 000 Мбит/с

Таблица 5-3. Типы Ethernet

10Base2. 10Base2, ThinNet использует коаксиальный кабель. Максимальная длина кабеля составляет 185 метров, обеспечивается скорость передачи 10 Мбит/с, требуются BNC-коннекторы (British Naval Connector) для сетевых устройств.

10Base5. 10Base5, ThickNet использует толстый коаксиальный кабель. При использовании ThickNet могут применяться более длинные сегменты кабеля, чем для ThinNet, поэтому ThickNet часто используется для магистральной сети. ThickNet более устойчив к электрическим помехам, чем ThinNet, поэтому обычно он предпочтительнее при прокладке кабеля через подверженное электрическим помехам пространство. При использовании ThickNet также требуются BNC-коннекторы, т.к. он тоже использует коаксиальный кабель.

10Base-T. 10Base-T использует витую пару с медными проводами вместо коаксиального кабеля. Витая пара использует один провод для передачи данных, а другой – для приема. 10Base-T обычно применяется в топологии «звезда», позволяющей легко настраивать сеть. В топологии «звезда» все системы подключены к центральному устройству в плоской или иерархической конфигурации.

Сети 10Base-T используют коннектор RJ-45, который используется для подключения компьютеров. Провода чаще всего прокладывают по стенам и подключают к коммутационной панели. Коммутационная панель обычно подключается к концентратору 10Base-T, который открывает дверь к магистральному кабелю или центральному коммутатору. Этот тип конфигурации показан на рисунке 5-20.

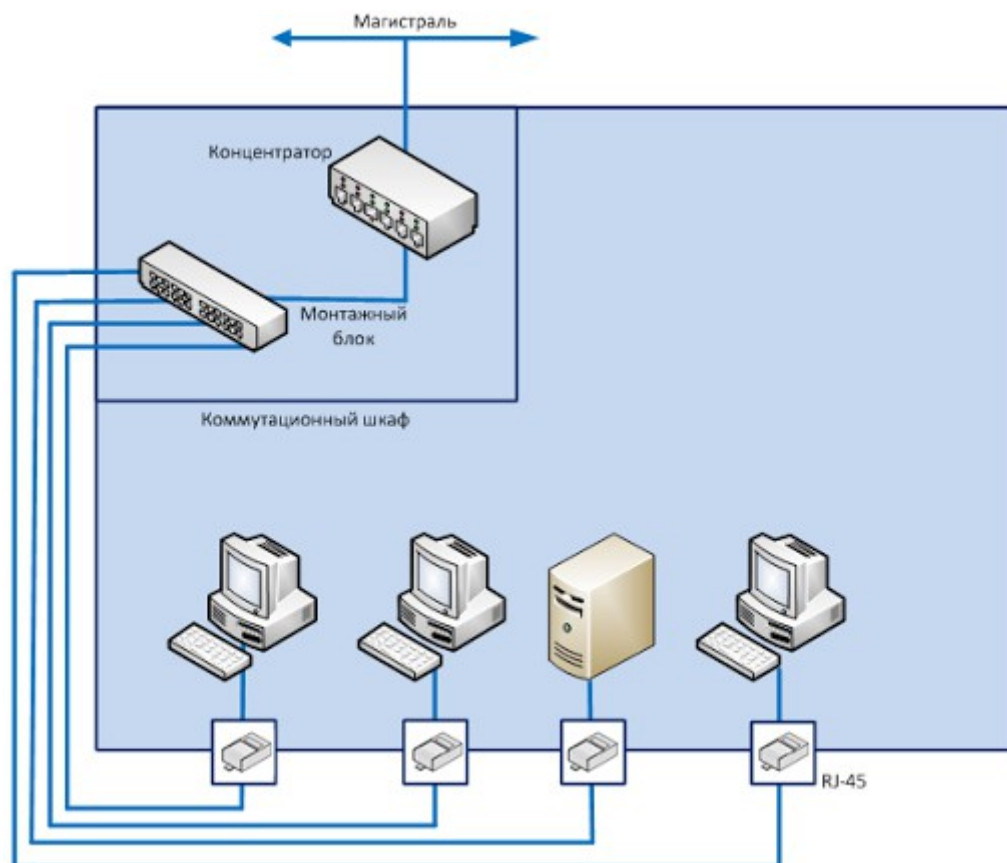


Рисунок 5-20. Ethernet-узлы подключены к коммутационной панели, соединенной с магистральным кабелем через концентратор или коммутатор

Fast Ethernet: Ускоренный Ethernet. Не удивительно, что когда-то скорость 10 Мбит/с казалась заоблачной, но сейчас большинству пользователей требуется значительно большая скорость. Для реализации этой потребности был разработан Fast Ethernet.

Fast Ethernet – это обычный Ethernet, но работающий на скорости 100 Мбит/с по витой паре. Примерно в то же время, когда появился Fast Ethernet, была разработана другая технология 100 Мбит/с – 100-VG-AnyLAN. Эта технология не использовала традиционный CSMA/CD Ethernet, она работала по-другому.

Fast Ethernet использует традиционный CSMA/CD (о ней рассказывается дальше в этом домене) и оригинальный формат кадра Ethernet. Именно поэтому он используется многими корпоративными средами LAN в настоящее время. В одной среде могут работать одновременно сетевые сегменты со скоростью 10 и 100 Мбит/с, соединенные через 10/100 концентратор или коммутатор.

В настоящее время существует четыре основных типа Fast Ethernet, они отличаются используемыми кабелями и дальностью передачи. Для более подробной информации о них перейдите по приведенным ниже ссылкам.

Ссылки по теме:

- [University of New Hampshire InterOperability Laboratory: Fast Ethernet Consortium](#)
- [Dan Kegel's Links for Fast Ethernet documents](#)
- [Charles Spurgeon's Ethernet Web Site](#)

Token Ring

Как и Ethernet, **Token Ring** – это технология LAN, позволяющая передавать информацию и совместно использовать сетевые ресурсы. Технология Token Ring была разработана IBM и определена в стандарте IEEE 802.5. Она использует технологию передачи маркеров

(токенов) в топологии, организованной в виде «звезды». Часть «ring» в имени технологии связана с потоком прохождения сигналов, образующим логическое кольцо. Каждый компьютер подключен к центральному концентратору, называемому **модулем множественного доступа** (MAU – Multistation Access Unit). Физически топология может быть организована в виде «звезды», однако сигналы проходят по логическому кольцу.

Технология передачи маркеров – это единственная технология, в которой данные не помещаются в сетевой провод, не имея специального **маркера** (token), являющегося управляющим кадром, путешествующим по логическому кольцу и захватываемым, когда какой-либо системе нужно взаимодействовать с другой системой. Это отличается от Ethernet, в котором все устройства пытаются взаимодействовать одновременно, из-за чего Ethernet зовут «болтливым протоколом», такой подход приводит к коллизиям. В Token Ring не бывает коллизий, так как только одна система может передавать информацию в каждый момент времени, однако из-за этого скорость передачи данных снижается по сравнению с Ethernet.

В начале технология Token Ring позволяла передавать данные со скоростью 4 Мбит/с, затем она была увеличена до 16 Мбит/с. Когда кадр помещается в провод, каждый компьютер просматривает его, чтобы проверить, не адресован ли он ему. Если кадр адресован не ему, компьютер помещает его обратно в сетевой провод, усиливая сигнал, который идет к следующему компьютеру по кругу.

Token Ring применяет пару механизмов для обработки проблем, которые могут возникнуть в сети этого типа. Механизм **активного мониторинга** удаляет кадры, которые продолжают кружить по сети. Это происходит, когда компьютер-получатель по какой-либо причине отключается от сети и маркер не может быть доставлен ему. Используя механизм **сигнализации** (beaconing), компьютер отправляет кадр сигнализации при обнаружении проблемы с сетью. Этот кадр создает сбойный домен (failure domain), который передается другим компьютерам с целью их информирования о проблемном участке. Компьютеры и устройства в рамках сбойного домена попытаются перенастроиться, чтобы попытаться обойти проблемный участок. На рисунке 5-21 показана сеть Token Ring с физической конфигурацией «звезда».

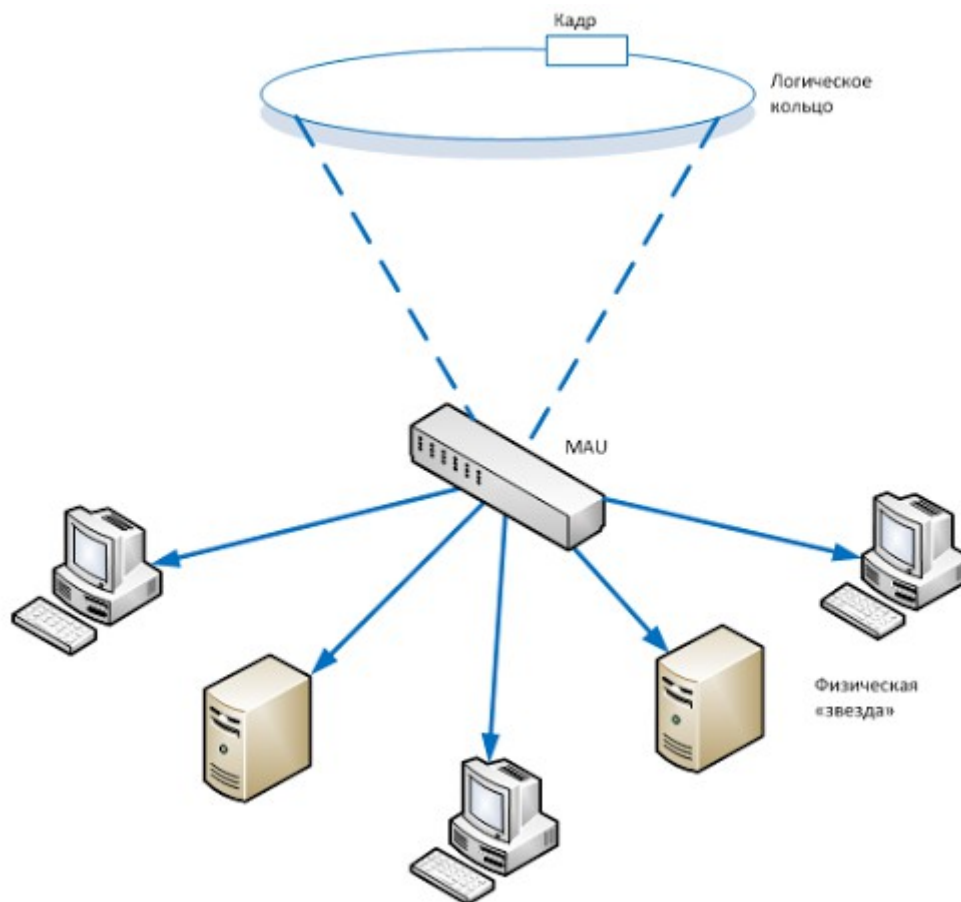


Рисунок 5-21. Сеть Token Ring

Сети Token Ring были популярны в 1980-х – 1990-х годах, некоторые из них еще используются и в настоящее время, однако Ethernet стала гораздо более популярной и взяла первенство на рынке сетей LAN.

Вопрос: В чем заключается различие между Ethernet, Token Ring и FDDI?

Ответ: Все это технологии канального уровня. Реально, канальный уровень состоит из подуровня MAC и подуровня LLC. Эти технологии работают на подуровне MAC и используют интерфейс для взаимодействия с подуровнем LLC. Эти технологии LAN отличаются тем, как они взаимодействуют со стеком протоколов, а также предоставляемой ими функциональностью.

Ссылки по теме:

- Token Ring FAQ
- Token-Ring Technical Summary, TechFest

FDDI

Технология FDDI (Fiber Distributed Data Interface), разработанная ANSI (American National Standards Institute) – это высокоскоростная технология доступа к среде, использующая передачу маркера. FDDI имеет скорость передачи данных до 100 Мбит/с и, как правило, применяется в качестве магистральной сети с использованием волоконно-оптических кабелей. FDDI обеспечивает отказоустойчивость путем создания второго кольца, вращающегося в обратную сторону. В основном кольце данные «крутятся» по часовой стрелке, именно это кольцо обычно используется для передачи данных. Во втором кольце данные «крутятся» против часовой стрелки, оно используется только тогда, когда основное кольцо не работает. Датчики следят за функционированием основного кольца и, если оно перестает работать, они переключают передачу данных на второе кольцо, задействуя для этого механизм восстановления целостности кольцевой сети (ring wrap). Каждый узел в сети FDDI имеет ретранслятор (relay), подключенный к обоим кольцам, поэтому в случае разрыва

в кольце, оба кольца могут быть соединены.

FDDI обычно используется как магистральная сеть, объединяющая несколько различных сетей, как показано на рисунке 5-22.

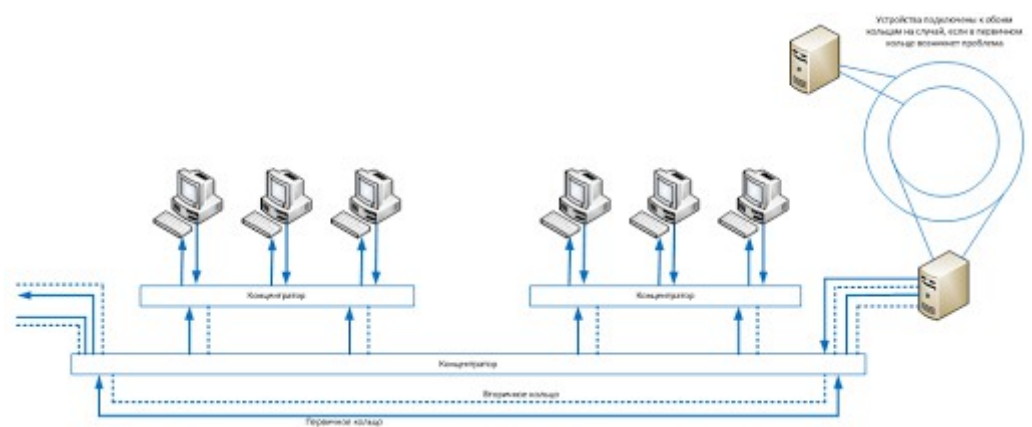


Рисунок 5-22. Кольца FDDI могут использоваться в качестве магистрали для объединения различных сетей LAN

До появления на рынке Fast Ethernet и Gigabit Ethernet, именно FDDI чаще всего использовалась в магистральных сетях. Поскольку FDDI может использоваться на расстояниях до 100 километров, эта технология часто используется для сетей MAN (metropolitan area networks). Преимущество технологии FDDI заключается в том, что она может работать на больших расстояниях с высокими скоростями и с минимальными помехами. Это позволяет нескольким маркерам одновременно находиться в кольце, что дает возможность одновременно передавать по сети несколько сообщений, а также обеспечивает предсказуемые задержки, что помогает соединенным сетям и устройствам знать, чего ожидать и когда.

Технология CDDI (Copper Distributed Data Interface), являющаяся разновидностью FDDI, может работать на витой паре (UTP). CDDI может использоваться в рамках LAN для соединения сетевых сегментов, тогда как FDDI предпочтительнее использовать в MAN.

ПРИМЕЧАНИЕ. FDDI-2 обеспечивает фиксированную полосу пропускания, которая может быть выделена для конкретных приложений. Это делает ее работу похожей на широкополосную связь, которая позволяет по одной линии передавать голос, видео и данные.

Таблица 5-4 резюмирует наиболее важные характеристики технологий, описанных в предыдущих разделах.

Реализация LAN	Стандарт IEEE	Характеристики
Ethernet	802.3	<ul style="list-style-type: none"> - Общая среда – все устройства должны по очереди использовать одну и ту же среду и выявлять коллизии - Использует широковещательные и коллизийные домены - Использует метод CSMA/CD - Может использовать коаксиальный кабель или витую пару - Скорость передачи от 10 Мбит/с до 1 Гбит/с
Token Ring	802.5	<ul style="list-style-type: none"> - Все устройства подключены к <u>центральному</u> MAU - Метод доступа к среде с передачей маркера - Скорость передачи от 4 до 16 Мбит/с - Использует механизмы активного мониторинга и сигнализации
FDDI	802.8	<ul style="list-style-type: none"> - Метод доступа к среде с передачей маркера - Два вращающихся в разные стороны кольца для повышения отказоустойчивости - Скорость передачи до 100 Мбит/с - Работает на больших расстояниях с высокими скоростями и поэтому используется в качестве магистрали - CDDi работает через UTP

Таблица 5-4. Методы доступа к среде LAN

4.3. Кабели

Прокладка сетевых кабелей играет важную роль при создании новой сети или расширении уже существующей. Типы используемых кабелей должны соответствовать используемым технологиям канального уровня. Кабели различаются по скорости передачи данных, максимальной длине, а также способам подключения к сетевым картам. В 1970-х – 1980-х годах, единственным вариантом был коаксиальный кабель, но в конце 1980 года, появилась витая пара, которая по сей день является наиболее популярным типом используемых сетевых кабелей.

На поток электрических сигналов, проходящих через сетевые кабели, может негативно влиять множество факторов внешнего окружения, таких как, двигатели, флюоресцентное освещение, магнитные поля и различные электрические приборы. Эти факторы могут повредить проходящие через кабель данные. Именно поэтому применяются различные стандарты кабелей, определяющие тип кабеля, защитную оболочку, скорость передачи данных и возможную длину кабеля.

Кабели имеют различную ширину полосы пропускания и связанную с ней скорость передачи данных. Хотя два этих термина связаны между собой, в действительности они разные.

Полоса пропускания (bandwidth) кабеля указывает максимальный диапазон частот, которые он использует, например, 10Base-T использует 10 МГц, а 100Base-TX использует 80 МГц. Это отличается от реального объема данных, которые могут быть переданы через кабель.

Скорость передачи данных (data throughput rate) является фактический объем данных, который в единицу времени проходит через кабель после сжатия и кодирования. 10Base-T имеет скорость передачи данных 10 Мбит/с, а 100Base-TX – 100 Мбит/с. Ширину полосы пропускания можно представить в виде размера трубы, а скорость передачи данных – в виде фактического объема данных, проходящего через нее в единицу времени.

Коаксиальный кабель

Коаксиальный кабель (coaxial cable) состоит из медного центрального провода (токопроводящей жилы), окруженного изоляционным слоем и заземляющим проводом, как показано на Рисунке 5-23. Это все помещено во внешнюю защитную оболочку. По сравнению с витой парой, коаксиальный кабель более устойчив к электромагнитным помехам (EMI), обеспечивает более широкую полосу пропускания, а также позволяет использовать кабель большей длины. Так почему же витая пара более популярна? Дело в том, что витая пара дешевле и удобнее в использовании, а переход на коммутируемую среду, предоставляющую иерархическую схему проводки кабелей, позволил решить проблемы с длиной кабеля витой пары.



Рисунок 5-23. Коаксиальный кабель

Два основных вида коаксиального кабеля, используемого в сетях LAN, это 50 Ом-ный кабель (используется для передачи цифрового сигнала) и 75 Ом-ный кабель (используется для высокоскоростной передачи цифрового сигнала и аналогового сигнала). Разновидностями коаксиального кабеля являются 10Base2 (ThinNet) и 10Base5 (ThickNet). По коаксиальному кабелю можно передавать данные, используя *однополосную* (baseband) передачу, когда по кабелю передается только один канал, или *широкополосную* (broadband) передачу, когда по

кабелю передается одновременно несколько каналов.

Витая пара

Витая пара состоит из изолированных медных проводов, заключенных во внешнюю защитную оплетку. **Экранированная витая пара** (STP – shielded twisted pair) имеет внешнюю защитную оплетку из фольги, которая улучшает защиту от радиочастотных и электромагнитных помех. Если витая пара не имеет дополнительного внешнего экранирования, она называется **неэкранированной витой парой** (UTP – unshielded twisted pair).

Кабель состоит из медных проводов, скрученных друг с другом, как показано на Рисунке 5-24. Это переплетение проводов защищает передаваемые сигналы от радиочастотных и электромагнитных помех, а также от перекрестных помех. Каждый провод образует уравновешенную схему, так как напряжение в каждой паре использует ту же амплитуду, но с противоположной фазой. Тугое переплетение проводов обеспечивает большую устойчивость кабеля от помех и затухания сигнала. В UTP входит несколько категорий кабелей, каждый из которых имеет свои уникальные характеристики. Разница в категориях кабеля основана на том, насколько туго сплетен кабель.

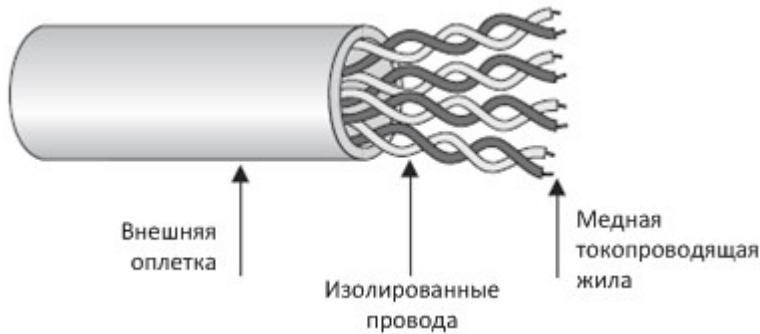


Рисунок 5-24. Витая пара

Скорость передачи данных определяется переплетением проводов, типом используемой изоляции, качеством токопроводящего материала и экранированием провода. Категории UTP указывают на характеристики этих компонентов, использованные при изготовлении кабеля.

Категория	Характеристики	Использование
Категория 1	Телефонный кабель для передачи голоса	Не рекомендуется для использования в сетях, но модемы могут работать через такой кабель
Категория 2	Передача данных на скорости до 4 Мбит/с	Используется для терминального доступа к мейнфреймам и миникомпьютерам, но не рекомендуется для использования в высокоскоростных сетях
Категория 3	10 Мбит/с для Ethernet и 4 Мбит/с для Token Ring	Используется в реализациях сети 10Base-T
Категория 4	16 Мбит/с	Обычно используется в сетях Token Ring
Категория 5	100 Мбит/с для сетей 100Base-TX и CDDi; имеет плотное переплетение проводов, что снижает перекрестные помехи	Используется в реализациях 100Base-TX, CDDi, Ethernet и ATM; широко используется при создании новых сетей
Категория 6	10 Гбит/с	Используется при создании новых сетей, требующих высокоскоростную передачу данных; является стандартом для Gigabit Ethernet
Категория 7	10 Гбит/с	Используется при создании новых сетей, требующих высокоскоростную передачу данных

Таблица 5-5. Категории кабеля UTP

Медный кабель используется повсеместно в течение многих лет. Он недорог и прост в использовании. Сегодня большинство телефонных систем используют медные кабели категории, подходящей для передачи голоса. Витая пара предпочтительна при создании

сетей, но у нее есть свои недостатки. Медь обладает электрическим сопротивлением, что приводит к деградации сигнала после прохождения им определенного расстояния. Именно поэтому даются рекомендации по максимальной длине медного кабеля. Если эти рекомендации не будут соблюдены, в сети могут происходить потери сигнала и повреждения данных. Кроме того, медь излучает энергию, что дает возможность злоумышленникам перехватывать передаваемую информацию. По сравнению с коаксиальным и оптоволоконным кабелем, UTP наименее безопасна. Если компании требуется более высокая скорость, более высокий уровень безопасности, а также большая длина кабелей, чем это позволяют медные кабели, наилучшим выбором для нее может быть оптоволоконный кабель.

Оптоволоконный кабель

Витая пара и коаксиальный кабель используют медные провода в качестве среды передачи данных, а оптоволоконный кабель использует разновидность стекла, через которое передаются световые волны. Эти световые волны переносят данные. Стеклоядро окружено защитной оболочкой, которая, в свою очередь, помещена в наружную оплетку.

Оптоволоконные кабели обеспечивают передачу сигналов на большие расстояния и с более высокой скоростью, поскольку для передачи данных используются световые волны. Оптоволоконные кабели не подвержены затуханию сигнала и электромагнитным помехам (EMI) в отличие от кабелей, в которых используются медные провода. Оптоволоконные кабели не излучают сигналы, в отличие от кабелей UTP, информацию с них трудно перехватить, поэтому оптоволоконные кабели гораздо более безопасны по сравнению с UTP, STP или коаксиальными кабелями.

Преимущества оптоволоконного кабеля звучат прекрасно, даже удивительно, зачем при этом продолжают использовать UTP, STP или коаксиальный кабель. К сожалению, оптоволоконный кабель чрезвычайно дорог и с ним трудно работать. Оптоволоконные кабели обычно используются в магистральных сетях и средах, которые требуют высокой скорости передачи данных. Большинство сетей используют UTP и подключены к магистральям, работающим на оптоволоконном кабеле.

Проблемы, связанные с кабелями

Кабели имеют чрезвычайно важное значение для сетей. Когда с сетевыми кабелями происходят проблемы, эти проблемы могут затронуть всю сеть. В этом разделе рассматриваются некоторые из наиболее распространенных проблем, связанных с кабелями, с которыми сталкивается большинство сетей.

Помехи

Помехи в кабелях обычно вызываются окружающим их оборудованием или характеристиками внешней среды. Помехи могут быть вызваны работой моторов, компьютеров, копировальных устройств, флуоресцентных ламп, микроволновых печей и т.п. Фоновые помехи могут накладываться на передаваемые по кабелю данные и искажать сигнал, как показано на Рисунке 5-25. Чем больше помех вокруг кабеля, тем более вероятно, что данные не дойдут до получателя или дойдут в искаженном виде. (Аналогичные проблемы воздействия на линии электропередач были рассмотрены в Доме 04).

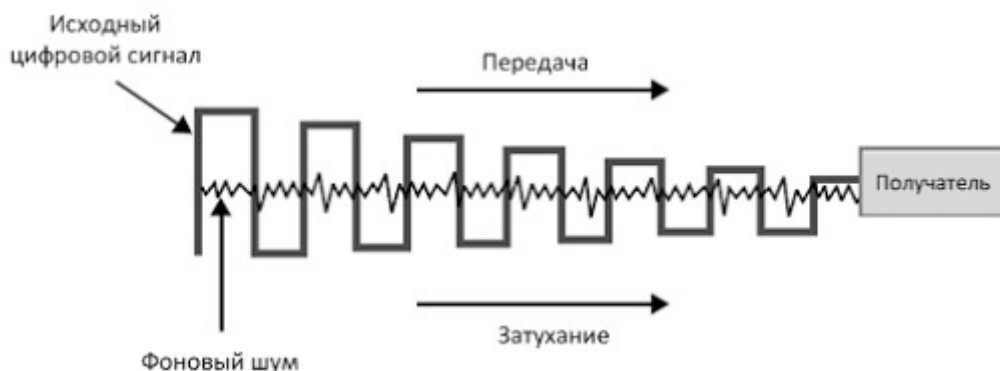


Рисунок 5-25. Фоновый шум может накладываться на электрический сигнал и нарушать целостность данных

Магистраль. Если используется кабель UTP и его длина превышает 185 метров, в нем может происходить затухание сигнала. Обычно данные находятся в форме электронов, «плывущих» по медному проводу. Однако это похоже на плавание против течения, т.к. существует сопротивление движению электронов в этой среде. После прохождения определенного расстояния электроны начинают терять темп, а формат их кодировки теряет форму. Если эта форма слишком деградирует, система получателя не сможет ее декодировать. Поэтому стандарт устанавливает ограничение по максимальной длине кабеля – не более 185 метров. Если сетевому администратору требуется кабель большей длины, он должен установить повторители (repeater) или другие аналогичные устройства, усиливающие сигнал и гарантирующие, что он дойдет до получателя в надлежащем виде.

Затухание

Затухание (attenuation) сигнала – это потеря мощности сигнала в процессе передачи. Чем длиннее кабель, тем больше затухание, что может привести к повреждению данных. Поэтому в стандарты включены рекомендации по максимальной длине кабеля; когда данные прошли определенное расстояние, сопротивление потоку электронов накапливается и сигнал теряет целостность.

Эффект затухания увеличивается на высоких частотах, поэтому у 100Base-TX, работающей на 80МГц, затухание больше, чем у 10Base-T, работающей на 10МГц. В связи с этим кабели, используемые для передачи данных на высоких частотах, следует делать короче, чтобы снизить воздействие затухания.

Затухание сигнала может также быть вызвано неисправностью кабеля. Поэтому кабели необходимо проверять. Если предполагается, что проблема вызвана затуханием сигнала в кабеле, тестировщик вводит на вход кабеля сигналы и считывает их на выходе из него, анализируя произошедшие изменения.

Перекрестные помехи

Как было сказано ранее, кабель UTP подвержен перекрестным помехам (crosstalk), при которых электрический сигнал из одного провода накладывается на сигнал в другом проводе. Когда различные электрические сигналы смешиваются, их целостность нарушается, и передаваемые данные могут быть повреждены. Кабель UTP в большей степени подвержен перекрестным помехам, чем STP или коаксиальный кабель, т.к. UTP не имеет дополнительного уровня экранирования, который помогает защититься от этого.

Как было сказано ранее, два провода в витой паре образуют уравновешенную схему, поскольку они оба имеют одинаковую амплитуду, но различные фазы. Перекрестные и фоновые помехи могут нарушить это равновесие и провода начнут работать как антенна, захватывающая все помехи из окружающего пространства.

Пожарные рейтинги кабелей

Поскольку здание должно соответствовать определенным пожарным кодексам, им должна

соответствовать и проводка. Многие компании прокладывают провода над подвесными потолками – в пространстве между подвесным и настоящим потолком, либо под фальшполом. Это скрывает кабели и позволяет людям не спотыкаться и не ходить по ним. Однако при возгорании кабелей, протянутых в таких местах, существует большая вероятность, что никто этого не заметит. Некоторые кабели при горении выделяют ядовитые газы, которые быстро распространяются по зданию. При прокладке сетевых кабелей в таких замкнутых пространствах, необходимо учитывать соответствующий пожарный рейтинг, чтобы обеспечить отсутствие ядовитых газов при пожаре. Следует учитывать, что система вентиляции здания обычно размещается в этих же замкнутых пространствах, поэтому если токсичные газы попадут в них, они могут за несколько минут распространиться по всему зданию.

Не предназначенные для замкнутых пространств кабели (nonplenum cable) обычно имеют оплетку из поливинилхлорида (PVC), а предназначенные – фторополимерную. Когда создается новая сеть или расширяется существующая, важно понимать, какие типы кабелей требуются в этой конкретной ситуации.

Вы должны учитывать следующие факторы при выборе сетевых кабелей: бюджет, выделенный компанией на создание (расширение, модернизацию) сети, простота использования, возможные помехи, необходимая длина кабелей, необходимая скорость передачи данных, требуемая безопасность и пожарный рейтинг.

Кабели следует прокладывать в недоступных местах (например, в коробах), чтобы никто не ходил по ним, чтобы они не были повреждены или прослушаны. Кабели следует прокладывать по стенам и в защищенных пространствах над подвесным потолком. В некоторых случаях кабели прокладывают в трубах под давлением, и если кто-то попытается получить доступ к проводу и нарушит целостность трубы, зазвучит тревога и администратору будет автоматически направлено соответствующее уведомление.

Если в окружающем пространстве установлено много станков или других устройств, создающих электромагнитные поля, следует использовать кабель STP или оптоволоконный. Если для компании важнее всего безопасность, следует использовать оптоволоконные кабели.

4.4. Методы передачи

Может потребоваться отправить пакет только одной рабочей станции, группе рабочих станций или одновременно всем рабочим станциям в подсети. Если пакет нужно отправить только одному получателю, используется метод **одноадресной** (unicast) передачи. Если пакет нужно отправить определенной группе получателей, отправляющая система использует метод **многоадресной** (multicast) рассылки. Если необходимо, чтобы сообщение получили все компьютеры в подсети, нужно использовать **широковещательный** (broadcast) метод.

Одноадресная передача самая простая, т.к. используется только адрес отправителя и адрес получателя. Данные просто идут из точки A в точку Z, от одного компьютера – другому (один-к-одному). Многоадресная рассылка немного отличается от одноадресной. С помощью многоадресной рассылки один компьютер может отправить данные выбранной группе компьютеров. Хорошим примером многоадресной рассылки является прослушивание сетевой радиостанции на компьютере. Существует программное обеспечение, которое позволяет пользователю выбрать желаемое направление музыки. Пользователь выбирает жанр, а программное обеспечение сообщает драйверу сетевой карты, что нужно принимать пакеты, содержащие определенный адрес групповой рассылки.

Разница между широковещательной и многоадресной передачей данных заключается в том, что широковещательная рассылка – это передача данных один-ко-всем, в то время как многоадресная рассылка – один-к-нескольким, выбранным в качестве получателей данных. Но каким образом сервер при многоадресной рассылке направляет данные определенному

компьютеру в конкретной сети? Пользователь, который выбрал получение рассылки, в действительности сообщил своему локальному маршрутизатору, что он хочет получать проходящие через этот маршрутизатор кадры с этим конкретным адресом групповой рассылки. Этот локальный маршрутизатор сообщает об этом вышестоящему маршрутизатору и так далее, пока каждый маршрутизатор между отправителем и получателем не будет знать, куда нужно передавать данные этой конкретной многоадресной рассылки. При этом в действительности пользователь не взаимодействует непосредственно с маршрутизаторами, за него это делает используемое им программное обеспечение.

IP-протоколы многоадресной рассылки используют адреса класса D, которые являются специальным адресным пространством, выделенным для многоадресной рассылки данных. Их можно использовать для отправки информации, мультимедиа-данных, а также голоса и видео в режиме реального времени.

Для сообщения маршрутизаторам информации о членах групп многоадресной рассылки используется протокол IGMP (Internet Group Management Protocol - Протокол управления группами интернета). Когда пользователь включает прием многоадресного трафика, он становится членом определенной группы многоадресной рассылки. IGMP является механизмом, который позволяет компьютерам информировать локальные маршрутизаторы, что они (компьютеры) являются частью определенной группы и что следует отправлять им трафик с соответствующим адресом многоадресной рассылки.

Определения связи. Ниже представлено краткое резюме по основным концепциям передачи данных, которые нужно знать:

- **Цифровые сигналы** – двоичные цифры, представленные в виде дискретных электрических импульсов.
- **Аналоговые сигналы** – непрерывные сигналы, отличающиеся друг от друга амплитудой и частотой.
- **Асинхронная передача** – последовательная передача данных, использующая стартовые и стоповые биты, и требующая, чтобы взаимодействующие устройства работали на одной скорости
- **Синхронная передача** – высокоскоростная передача данных, управляемая сигналами электронных часов
- **Однополосная передача** – вся полоса пропускания используется только для одного канала, имеет низкую скорость передачи данных
- **Широкополосная передача** – делит полосу пропускания на несколько каналов, позволяя одновременно передавать различные типы данных, обеспечивает высокую скорость передачи
- **Одноадресная передача** – пакет передается с одного компьютера-отправителя на один компьютер-получатель
- **Многоадресная передача** – пакет передается с одного компьютера-отправителя, на несколько определенных компьютеров-получателей
- **Широковещательная передача** – пакет отправляется с одного компьютера-отправителя на все компьютеры определенного сетевого сегмента

Ссылки по теме:

- Internetworking Technology Handbook, Chapter 43, “Internet Protocol Multicast,” Cisco Systems, Inc.

4.5. Технологии доступа к среде

Физическая топология сети – это нижний уровень, фундамент сети. Она определяет тип используемой среды и порядок соединения средой передачи данных различных систем. Технологии доступа к среде определяют порядок взаимодействия систем через среду,

обычно они представляют собой протоколы, драйверы сетевых карт и интерфейсы. Технологии доступа к LAN устанавливают правила взаимодействия компьютеров по сети, правила обработки ошибок, использования физической среды, устанавливают максимальный размер MTU кадра и многое другое. Эти правила позволяют всем компьютерам и устройствам взаимодействовать, исправлять ошибки, а пользователям – позволяют эффективно выполнять свои сетевые задачи. Каждая отдельная система должна знать, как правильно взаимодействовать с другими системами, чтобы они понимали ее передачи, команды и запросы. Обо всем этом заботится технология доступа к среде LAN.

ПРИМЕЧАНИЕ. Параметр MTU указывает, как много данных может содержать кадр в конкретной сети. Различные типы сетевых технологий могут требовать различный размер MTU, из-за этого кадры часто бывают фрагментированными.

Передача маркера

Маркер (token) – это 24-битный управляющий кадр, используемый для управления взаимодействием компьютеров и интервалами этого взаимодействия. Маркер передается с компьютера на компьютер, и только тот компьютер, который имеет маркер, может отправлять кадры с данными по сети. Маркер дает компьютерам право на взаимодействие. Маркер содержит передаваемые данные, а также информацию об адресах отправителя и получателя. Если компьютеру нужно передать данные, он ждет маркер. После получения маркера, компьютер прикрепляет к нему свое сообщение и помещает его в сетевой провод. Каждый компьютер, который затем получит это сообщение, проверяет – не адресовано ли оно ему. Это продолжается до тех пор, пока компьютер-получатель не получит сообщение. Компьютер-получатель делает себе копию сообщения и устанавливает в кадре специальный бит, чтобы сообщить компьютеру-отправителю, что он получил сообщение. Когда этот пакет возвращается компьютеру-отправителю, он удаляет этот кадр из сети. Обратите внимание, что компьютер-получатель делает себе копию сообщения, но он не удаляет сообщение из сети. Только компьютер-отправитель сообщения может удалить его из маркера и сети.

Если компьютер получает маркер, но у него нет сообщений для отправки, он просто передает маркер следующему компьютеру в сети. Пустой маркер имеет заголовок, поле данных и окончание. При добавлении к пустому маркеру сообщения, он получает новый заголовок, адреса отправителя и получателя, а также новое окончание.

Такой метод доступа к сети применяется технологиями Token Ring и FDDI.

ПРИМЕЧАНИЕ. Некоторые приложения и сетевые протоколы работают лучше, если они могут взаимодействовать через определенные интервалы времени, а не «когда придут данные». В технологиях с передачей маркера передача трафика приходит имеет детерминированную природу, т.к. все системы не могут взаимодействовать одновременно – взаимодействовать могут только системы, имеющие маркер.

CSMA

Протоколы Ethernet определяют порядок взаимодействия узлов, исправления ошибок, использования общего сетевого кабеля. Ethernet использует CSMA для доступа к сетевому кабелю. Существует два различных типа CSMA: CSMA/CD и CSMA/CA.

Передача данных называется *несущей* (carrier), поэтому, если компьютер передает кадры, он выполняет функции несущей. Если компьютеры используют протокол **множественного доступа с контролем несущей и с выявлением коллизий** (CSMA/CD – Carrier Sense Multiple Access with Collision Detection), они отслеживают передачу данных (или несущую) в проводе, чтобы определить наилучшее время для передачи данных. Каждый узел постоянно контролирует провод и ждет, пока он освободится, чтобы этот узел мог передать свои данные. В качестве аналогии представьте себе беседу нескольких людей. Если один человек хочет что-то сказать, он обычно сначала слушает уже говорящего и ждет паузы, чтобы начать говорить самому. Если он не будет дожидаться, пока договорит первый и будет

говорить одновременно с ним, люди вокруг не смогут понять ни одного из говорящих.

При использовании метода доступа CSMA/CD, компьютеры прослушивают кабель и ждут момента, когда в нем не будет сигнала несущей, что означает, что никто не передает данные. Если два компьютера зафиксируют отсутствие несущей и одновременно начнут передачу своих данных, может произойти конфликт и коллизия. При возникновении **конфликта** (contention) все узлы должны прекратить совместное использование среды передачи данных. **Коллизия** (collision) происходит при столкновении двух или более кадров, что приводит к повреждению обоих кадров. Если компьютер отправил кадр по проводу, но он столкнулся с кадром другого компьютера, передача обоих компьютеров прерывается, и все остальные компьютеры оповещаются о возникновении коллизии. При получении информации о коллизии все компьютеры запускают коллизионный таймер на случайно выбранное время, которое будет являться задержкой, по прошествии которой они снова начнут предпринимать попытки передать данные. Этот коллизионный таймер, запускаемый на случайное время, называется **алгоритмом выдержки** (back-off algorithm). (Число коллизий обычно снижается при разделении сети мостами или коммутаторами).

Множественный доступ с контролем несущей и предотвращением коллизий (CSMA/CA – Carrier Sense Multiple Access with Collision Avoidance) – это метод доступа, при использовании которого каждый компьютер сигнализирует о своем намерении передавать данные до начала реальной передачи. При получении такого сигнала все остальные компьютеры в сети понимают, что им пока нельзя передавать данные, т.к. это может привести к коллизии. Система прослушивает общую среду, чтобы определить, свободна ли она. Как только система определяет, что линия свободна и готова к передаче данных, эта система отправляет широковещательное сообщение всем остальным системам, говоря им, что она собирается передавать данные. После получения такого сообщения, каждая из остальных систем ждет определенный промежуток времени перед попыткой самой начать передачу данных, чтобы избежать коллизий. Технологии беспроводных сетей 802.11 используют технологию CSMA/CA для доступа к среде.

Методы доступа с контролем несущей и передачей маркера. В целом методы доступа с контролем несущей работают быстрее методов с передачей маркера, но для них актуальна проблема коллизий. В сетевом сегменте с большим количеством устройств может возникать большое количество коллизий, снижая производительность сети. Для технологий с передачей маркера проблемы коллизий не существует, но они не обладают такой скоростью, как технологии с контролем несущей. Сетевые коммутаторы могут существенно помочь в изоляции сетевых ресурсов для обоих методов (CSMA/CD и передача маркера), поскольку это снижает конкуренцию.

Коллизионные домены

Как было указано в предыдущем разделе, коллизии происходят в сетях Ethernet, когда два компьютера пытаются передавать данные одновременно. Другие компьютеры в сети выявляют эту коллизию, т.к. при этом наложившиеся друг на друга сигналы увеличивают напряжение сигнала выше определенного уровня. Чем больше устройств в конкурентной (contention-based) сети, тем выше вероятность коллизий, что увеличивает **латентность** такой сети (задержки при передаче данных). **Коллизионный домен** (collision domain) – это группа компьютеров, которые конкурируют (или соперничают) за доступ к среде передачи данных.

Неприемлемое количество коллизий может быть вызвано перенаселенностью сети, повреждением сетевого кабеля или коннектора, слишком большим количеством повторителей, либо кабелем, длина которого превышает рекомендуемый максимум. Если кабель длиннее, чем это рекомендуется спецификацией Ethernet, два компьютерам на противоположных концах кабеля могут начать передавать данные одновременно. При этом они могут не узнать о том, что произошли коллизии, поскольку они находятся слишком далеко друг от друга. Если кабель слишком длинный, компьютеры не могут прослушивать

сеть достаточно хорошо, чтобы выявлять коллизии. При получении компьютером-получателем разрушенного кадра, он отправляет запрос отправителю на повторную передачу сообщения, что становится причиной увеличения трафика.

Проблемы такого типа в основном связаны с реализацией коллизионных доменов. Сеть Ethernet может иметь широковещательные и коллизионные домены. Одна подсеть будет находиться в одном коллизионном и широковещательном домене, если она не разделена маршрутизаторами или мостами. Если подсеть разделена мостами, мосты могут позволить широковещательному трафику проходить между различными частями подсети, но не коллизиям, как показано на рисунке 5-26. Так создаются коллизионные домены. Изоляция коллизионных доменов снижает количество происходящих в сети коллизий и увеличивает общую производительность сети.

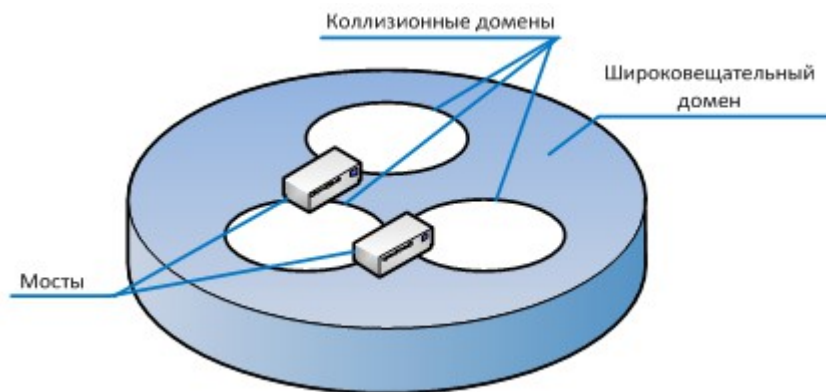


Рисунок 5-26. Коллизионные домены в рамках одного широковещательного домена

Другое преимущество ограничения и управления широковещательными и коллизионными доменами состоит в том, что это затрудняет перехват злоумышленником критичной информации, передающейся по сети (сниффинг). Атакующие часто используют тактику установки на скомпрометированный компьютер троянской программы, выполняющей функции сетевого сниффера. Этот сниффер обычно настраивается на перехват определенной информации, такой как имена пользователей и пароли. Если широковещательный и коллизионный домен находятся под таким воздействием, скомпрометированная система может получить доступ к широковещательному и коллизионному трафику только в рамках отдельной подсети или широковещательного домена. Скомпрометированная система не сможет прослушивать трафик в другом широковещательном и коллизионном домене, что значительно снизит объем трафика и информации, доступной атакующему.

Опрос

Третья разновидность методов доступа к среде LAN – это **опрос** (polling). При использовании этого метода, некоторые системы настроены как первичные станции, а другие – как вторичные. Через предопределенные интервалы времени первичные станции спрашивают у вторичных станций – есть ли у них данные для передачи. Вторичные станции могут передавать данные только в этот момент.

Опрос – это метод мониторинга множества систем и контролируемой передачи данных по сети. Если используется метод опроса для мониторинга устройств, первичное устройство взаимодействует с каждым вторичным устройством, через определенные интервалы времени для проверки их состояния. Затем первичное устройство фиксирует в журнале полученный ответ и переходит к следующему устройству. При использовании опроса для доступа к сети, первичная станция спрашивает каждое устройство, есть ли у него данные для передачи другому устройству. Метод опроса обычно используется в среде мейнфреймов.

4.6. Протоколы LAN

Некоторые протоколы, такие как UDP, TCP и IP, обсуждались в предыдущих разделах. Сети

используют эти и многие другие протоколы, для реализации определенного объема функциональности. Наиболее широко используются такие протоколы TCP/IP, как ARP, DHCP и ICMP. Они будут обсуждаться в следующих разделах.

ARP

В сети TCP/IP каждому компьютеру и сетевому устройству требуется уникальный IP-адрес и уникальный физический аппаратный адрес. Каждая сетевая карта имеет уникальный физический аппаратный адрес, который прошивается производителем в микросхеме ее ПЗУ. Этот физический адрес также называют **MAC-адресом** (Media Access Control). Сетевой уровень работает с IP-адресами и понимает только их, а канальный уровень - MAC-адреса. Но как обеспечить совместную работу двух этих адресов, использующихся на разных уровнях?

ПРИМЕЧАНИЕ. MAC-адрес является уникальным, поскольку первые 24 бита в нем являются кодом производителя, а последние 24 бита – представляют собой уникальный серийный номер устройства, присвоенный производителем.

Когда данные приходят с прикладного уровня, они поступают на транспортный уровень для присвоения порядкового номера, создания сеанса и включения в поток. Затем данные переходят на сетевой уровень, где к каждому пакету добавляется информация маршрутизации и IP-адреса отправителя и получателя. Затем данные направляются на канальный уровень, который должен найти MAC-адрес и добавить его в заголовок кадра. Когда кадр физически попадает в провод, он направляется по MAC-адресу, указанному в заголовке кадра. Работающие на этом уровне модели OSI механизмы не понимают IP-адреса. Поэтому, если компьютер не может при передаче пакета на нижний уровень сопоставить IP-адресу соответствующий MAC-адрес, взаимодействие с компьютером-получателем невозможно.

ПРИМЕЧАНИЕ. Кадр – это полностью инкапсулированные данные со всеми необходимыми заголовками и окончаниями.

Чтобы сетевое взаимодействие было возможно, MAC- и IP-адреса должны быть корректно сопоставлены. Это производится с помощью протокола **ARP** (Address Resolution Protocol - Протокол определения адреса). Когда канальный уровень получает кадр, сетевой уровень уже прикрепил к нему IP-адрес получателя, но канальный уровень не понимает IP-адресов, поэтому он призывает на помощь ARP. ARP делает широковещательный запрос MAC-адреса, соответствующего указанному IP-адресу получателя. Каждый компьютер в подсети получает этот запрос и игнорирует его, если его IP-адрес не соответствует запрашиваемому. Только компьютер с соответствующим запрашиваемому IP-адресом отвечает на запрос, указывая в ответе свой MAC-адрес. Теперь ARP знает, какой аппаратный адрес соответствует этому IP-адресу. Канальный уровень добавляет к кадру MAC-адрес и передает его на физический уровень, который помещает его в провод для отправки компьютеру-получателю. ARP таким образом связывает аппаратный адрес с соответствующим ему IP-адресом и сохраняет информацию об этой связи в специальной таблице, чтобы в дальнейшем использовать ее повторно и сэкономить время. В случае, если новый кадр будет адресован тому же IP-адресу, такое кэширование позволяет ARP не выполнять повторный широковещательный запрос – он просто берет информацию из своей таблицы.

Иногда атакующие изменяют ARP-таблицу системы, внося в нее некорректную информацию. Это называется **«отравлением ARP»** (ARP table poisoning). Целью атакующего в этом случае является получение пакетов, предназначенных для другого компьютера. Это разновидность атаки **маскарадинга** (masquerading attack). Например, компьютер А имеет IP-адрес 10.19.34.3 и MAC-адрес X, и эта информация сохранена в ARP-таблице компьютера В. Атакующий может изменить эту ARP-таблицу и указать, что IP-адрес 10.19.34.3 соответствует MAC-адресу Y (MAC-адресу компьютера атакующего). Тогда все пакеты компьютера В, которые он будет пытаться отправить на компьютер А, в действительности

будут отправляться на компьютер атакующего.

Ссылки по теме:

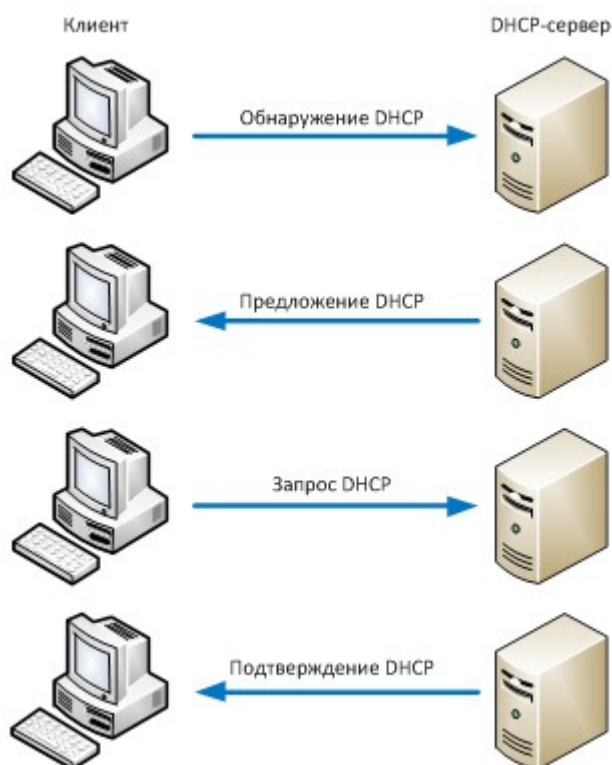
- Address Resolution Protocol (ARP), explanation by Gorrry Fairhurst
- Connected: An Internet Encyclopedia, “ARP Protocol Overview,” Freesoft

DHCP

Компьютер при первоначальной загрузке может получить IP-адреса несколькими различными способами. Если он имеет статический адрес, ему не нужно присваивать никаких других адресов. Если компьютеру требуется DHCP-сервер (Dynamic Host Configuration Protocol - Протокол динамической конфигурации узла) для получения корректного IP-адреса, при загрузке он направляет запрос DHCP-серверу, который присваивает ему IP-адрес.

DHCP – это протокол, основанный на UDP, который позволяет серверам присваивать сетевым клиентам IP-адреса в режиме реального времени. В отличие от статических IP-адресов (настроенных вручную), DHCP автоматически проверяет доступные IP-адреса и присваивает их клиентам. Это исключает возможные конфликты адресов, вызванные тем, что две системы в сети имеют одинаковый IP-адрес. Кроме того, DHCP значительно снижает объем работы, необходимой для управления большими сетями.

DHCP присваивает IP-адреса из определенного диапазона в режиме реального времени, при подключении клиента к сети. Это отличается от использования статических адресов, когда каждой системе изначально присвоен IP-адрес. В стандартной сети, использующей DHCP, для получения IP-адреса клиентский компьютер отправляет по сети широковещательное сообщение DHCPDISCOVER (обнаружение DHCP), чтобы найти DHCP-сервер. Когда соответствующий DHCP-сервер получает запрос DHCPDISCOVER, он отвечает на него пакетом DHCPOFFER (предложение DHCP), предлагая клиенту IP-адрес. Сервер присваивает клиенту свободный IP-адрес в соответствии с административными политиками сети. Отправляемый сервером пакет DHCPOFFER содержит информацию о присвоенном IP-адресе и конфигурационных параметрах для настройки служб на стороне клиента.



После получения клиентом направленных сервером в пакете DHCP OFFER настроек, он отвечает серверу пакетом DHCP REQUEST, подтверждая принятие этих настроек. Сервер в ответ отправляет подтверждение пакетом DHCP ACK, указывая в нем период действия (*аренды*) предоставленных параметров.

К сожалению, при этом обе стороны (и клиент, и сервер) уязвимы к фальсификации. На стороне клиента атакующий может сделать собственную систему похожей на легитимного клиента сети. Это позволит его системе стать частью сети компании и попытаться проникнуть на другие системы этой сети. Также, атакующий может создать поддельный DHCP-сервер в сети, который будет взаимодействовать с настоящими клиентами, ищущими IP-адреса. Управляемый атакующим DHCP-сервер может скомпрометировать настройки клиентской системы, выполнить атаку «человек посередине» (man-in-the-middle), несанкционированно пересылать трафик в другую сеть и т.п., и в конечном итоге подвергнуть опасности всю сеть.

Эффективным методом защиты сетей от неуполномоченных клиентов DHCP является использование *перехвата DHCP* (DHCP snooping) на сетевых коммутаторах. Перехват DHCP обеспечивает, что DHCP серверы могут присваивать IP-адреса только выбранным системам, идентифицированным по их MAC-адресам. Также, современные коммутаторы имеют возможность направления клиентов непосредственно легитимным серверам DHCP для получения IP-адресов, исключая возможность для злоумышленника сделать свою систему DHCP-сервером сети.

Но что если в сети есть бездисковые рабочие станции, которые не имеют операционной системы? Бездисковые рабочие станции имеют достаточно кода только для того, чтобы включиться и направить широковещательный запрос на получение IP-адреса. Также они хранят указатель на сервер, содержащий операционную систему. Бездисковые рабочие станции знают свой аппаратный адрес и включают его в широковещательный запрос. Для этого аппаратного адреса DHCP-сервер выдает IP-адрес. Аналогично ARP, кадры **RARP**

(Reverse Address Resolution Protocol - Обратный протокол определения адреса) отправляются всем системам в подсети, но только RARP-сервер отвечает на них. После получения RARP-сервером запроса, он просматривает свою таблицу в поисках IP-адреса, соответствующего аппаратному адресу, указанному в широковещательном запросе. Затем сервер отправляет сообщение с найденным IP-адресом обратно запросившему компьютеру. После получения системой IP-адреса она может работать в сети.

Протокол **BOOTP** (Boot Protocol) был создан для расширения функциональности RARP, предоставляемой им бездисковым рабочим станциям. Бездисковые рабочие станции могут получить от BOOTP-сервера свой IP-адрес, адрес сервера имен для последующего разрешения имен, а также адрес шлюза по умолчанию. BOOTP предоставляет больше функциональности для бездисковых рабочих станций, чем RARP.

Этот протокол развивался следующим образом: RARP был преобразован в BOOTP, а BOOTP – в DHCP.

Различия между ARP и RARP. ARP знает IP-адрес и отправляет широковещательный запрос, чтобы получить соответствующий ему аппаратный MAC-адрес. RARP, наоборот, знает аппаратный адрес и отправляет широковещательный запрос для получения IP-адреса.

ICMP

Протокол ICMP (Internet Control Message Protocol - Межсетевой протокол управляющих сообщений) передает статусные сообщения, отчеты об ошибках, ответы на некоторые запросы, отчеты по информации маршрутизации. Он используется для проверки соединений и выявления проблем в IP-сетях.

Проще всего понять протокол ICMP с помощью утилиты *ping*. Эта утилита нужна для проверки соединения с другой системой. Она отправляет проверяемой системе кадр ICMP ECHO REQUEST, а та отвечает на него кадром ICMP ECHO REPLY, получение которого утилита *ping* отображает на экране. Если ответ не был получен в течение определенного периода времени, утилита *ping* повторно отправляет кадры ECHO REQUEST. Если ответа снова нет, утилита *ping* сообщает, что проверяемая система недоступна.

ICMP также указывает на возникновение проблем с каким-либо маршрутом в сети и сообщает окружающим маршрутизаторам о лучших маршрутах на основе информации о функционировании и перегрузке различных маршрутов. Маршрутизаторы используют ICMP для отправки сообщений в ответ на датаграммы, которые не удалось доставить. При этом маршрутизатор выбирает соответствующий ICMP-ответ и отправляет его обратно запросившей системе, говоря ей, что возникла проблема с передачей ее запроса.

ICMP используется другими протоколами, не устанавливающими соединений, а не только IP, поскольку такие протоколы не имеют способов выявления и реакции на ошибки передачи данных, аналогичных используемым протоколами, устанавливающими соединения. Протоколы, не устанавливающие соединения, могут использовать ICMP для отправки сообщений об ошибках системам-отправителям для информирования их о сетевых проблемах.

Атака Loki. Протокол ICMP был разработан для отправки сообщений о статусе, но не для хранения или передачи пользовательских данных. Но сейчас появились способы вставки данных внутрь ICMP-пакета, что может использоваться для взаимодействия с ранее скомпрометированной системой. Loki – это настоящая клиент-серверная программа, используемая хакерами для установления «черного хода» в системы. Атакующий взламывает компьютер и устанавливает на него серверную часть Loki, которая слушает порт, являющийся «черным ходом» и используемый злоумышленником для доступа в систему. Для получения доступа и открытия удаленной командной строки на этом компьютере, атакующий отправляет команды внутри ICMP-пакетов. Обычно они успешно доходят до цели, поскольку маршрутизаторы настроены на разрешение трафика ICMP на вход и выход из сети, т.к. он считается безопасным – ведь он был разработан не для того, чтобы содержать данные или иную полезную нагрузку.

Ссылки по теме:

- RFC 792 – Internet Control Message Protocol
- “LOKI ICMP Tunneling Back Door,” Internet Security Systems’ X-Force Database
- Securing Routers Against Hackers and DoS (PowerPoint Presentation)

5. Протоколы маршрутизации

Отдельные сети в Интернет называются **автономными системами** (AS – autonomous system). Эти AS независимо управляются различными корпорациями и организациями. AS создаются с помощью маршрутизаторов, самостоятельно администрируемых каждой компанией, которой принадлежит соответствующая AS. Границы AS определяются граничными маршрутизаторами, на границах используется протокол IGP (Interior Gateway Protocol - Внутренний шлюзовой протокол). Граничные маршрутизаторы одной AS соединяются с граничными маршрутизаторами других AS, при этом применяются внутренние и внешние протоколы маршрутизации. Внутренние маршрутизаторы соединяются с другими маршрутизаторами в рамках одной AS, при этом применяются внутренние протоколы маршрутизации. Таким образом, Интернет – это в действительности просто сеть, созданная из автономных систем и протоколов маршрутизации.

Архитектура Интернет, поддерживающая эти различные AS, создана таким образом, чтобы при подключении к отдельной AS не требовалось знать и понимать используемые внутри нее протоколы. Для взаимодействия с AS используются те же внешние протоколы маршрутизации (см. Рисунок 5-27). В качестве аналогии представьте, что вам нужно передать пакет другу, который живет в другой стране. Вы передаете пакет брату, который собирается ехать на поезде в город на границе с этой страной, чтобы он отправил пакет по почте на границе. При этом вы знаете, как ваш брат доедет до границы – на поезде, но вы не знаете, как почтовая служба доставит ваш пакет вашему другу домой (на грузовике, автобусе или машине), и это вас не волнует. Пакет достигнет цели без вашего участия. Точно также при взаимодействии сетей, одна сеть кладет пакеты данных на внешний протокол (поезд), после чего этот пакет перемещается на граничный маршрутизатор другой сети (граница страны). Таким образом, данные будут переданы в любом случае, какой бы внутренний протокол не использовался в сети получателя. Протоколы маршрутизации используются маршрутизаторами для определения маршрута между системами отправителя и получателя.

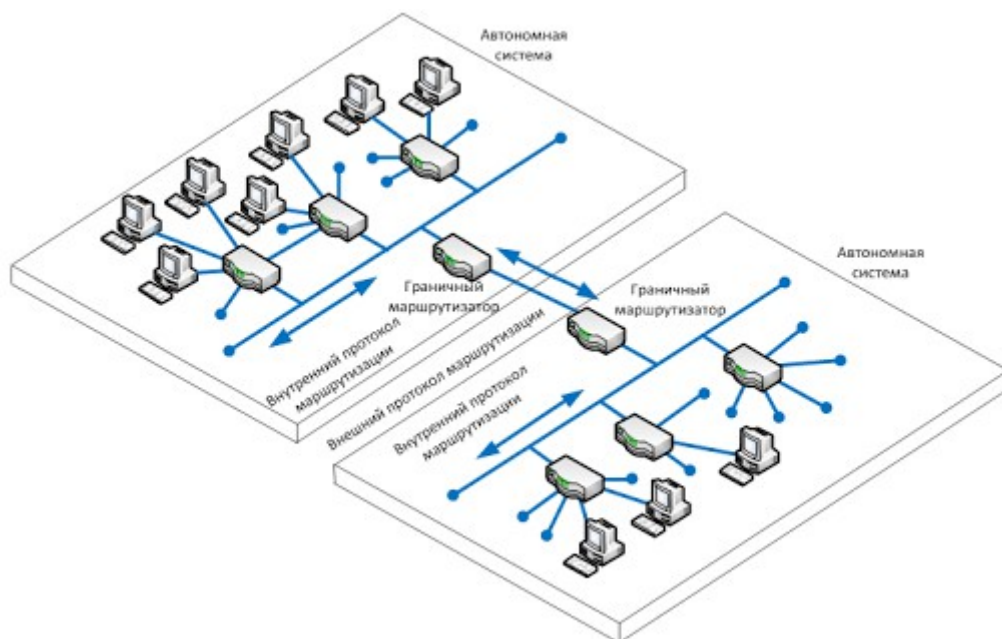


Рисунок 5-27. Автономные системы

Протоколы маршрутизации могут быть динамическими или статическими. **Динамический протокол маршрутизации** (dynamic routing protocol) может находить маршруты и строить таблицу маршрутизации. Маршрутизаторы используют таблицы маршрутизации для принятия решения о наилучшем маршруте для поступивших пакетов. Динамический протокол маршрутизации может изменять записи в таблице маршрутизации на основе изменений различных маршрутов. Когда маршрутизатор, использующий динамический протокол маршрутизации, обнаруживает, что какой-либо маршрут не работает или перегружен, он отправляет соответствующее уведомление об этом другим маршрутизаторам вокруг него. Получившие это уведомление маршрутизаторы обновляют на основе него свои таблицы маршрутизации, что необходимо для реализации эффективной маршрутизации. **Статический протокол маршрутизации** (static routing protocol) требует ручной настройки таблицы маршрутизации администратором.

Такие изменения маршрутов, вызванные постоянным изменением доступности маршрутизаторов, называются *колебанием маршрутов* (route flapping). Если маршрутизатор не получил уведомление о неработоспособности какого-либо маршрута, он продолжит пересылать пакеты по этому маршруту, который в этом случае будет называться *черной дырой* (black hole).

Двумя основными типами используемых протоколов маршрутизации являются протокол дистанционно-векторной маршрутизации (distance-vector routing protocol) и протокол маршрутизации по состоянию канала (link-state routing protocol). **Протоколы дистанционно-векторной маршрутизации** принимают решения о маршрутах на основе расстояния (или количества «прыжков» (hop)) и вектора (направления). Протокол использует эти переменные в алгоритме определения лучшего маршрута для пакета. **Протоколы маршрутизации по состоянию канала** строят более точную таблицу маршрутизации, поскольку они строят топологическую базу данных сети. Эти протоколы анализируют большее количество переменных, чем просто количество «прыжков» между двумя системами – в своих алгоритмах определения наилучшего маршрута для пакета они используют такие параметры, как размер пакета, скорость соединения, задержки, загруженность и надежность.

Таким образом, протокол дистанционно-векторной маршрутизации смотрит только на число «прыжков» между двумя системами, считая все «прыжки» одинаковыми. Протоколы маршрутизации по состоянию канала рассматривают больше частей этой головоломки – они анализируют не просто количество «прыжков», но и понимают состояние каждого из этих «прыжков», учитывая это при принятии решения о маршруте. Как вы узнаете далее, RIP – это пример протокола дистанционно-векторной маршрутизации, а OSPF – пример протокола маршрутизации по состоянию канала. OSPF более предпочтителен и чаще используется в больших сетях. RIP также применяется, но его используют в основном в маленьких сетях.

Ниже приведены примеры некоторых внутренних протоколов маршрутизации, используемых в настоящее время:

- **RIP** (Routing Information Protocol - Протокол информации маршрутизации) – это стандарт, который описывает, как маршрутизаторы обмениваются данными таблиц маршрутизации. Он основан на протоколе дистанционно-векторной маршрутизации, т.е. он рассчитывает самое короткое расстояние между отправителем и получателем. Он считается унаследованным протоколом, поскольку имеет низкую производительность и небольшую функциональность. Его следует использовать только в небольших сетях. Следует также учитывать, что RIP версии 1 не имеет аутентификации, RIP версии 2 отправляет пароли открытым текстом, либо хэширует с помощью MD5.
- **OSPF** (Open Shortest Path First - Протокол маршрутизации по принципу выбора кратчайшего пути) использует для отправки данных таблиц маршрутизации

алгоритмы, учитывающие состояние канала. Применение этих алгоритмов позволяет выполнять небольшие, но более частые обновления таблиц маршрутизации, что обеспечивает большую стабильность сети (по сравнению с RIP), но требует больше памяти и ресурсов процессора для поддержания этих процессов. OSPF позволяет использовать иерархическую маршрутизацию сети, которая имеет магистраль, соединяющую вместе все подсети. На сегодняшний день OSPF заменил RIP во множестве сетей. В OSPF аутентификация может осуществляться с помощью пароля, передаваемого открытым текстом, либо хэш-функции пароля, но также существует возможность отключить аутентификацию на маршрутизаторах с использованием этого протокола.

- **IGRP** (Interior Gateway Routing Protocol - Протокол внутренней маршрутизации между шлюзами) – это протокол дистанционно-векторной маршрутизации, который был разработан Cisco и является ее собственностью. В то время как RIP использует только один критерий для поиска наилучшего маршрута между отправителем и получателем, IGRP использует 5 критериев для принятия решения о наилучшем маршруте. Сетевой администратор может настроить вес этих различных критериев, чтобы обеспечить максимально эффективную работу этого протокола в конкретной сети.

ПРИМЕЧАНИЕ. Несмотря на то, что большинство протоколов маршрутизации имеет функции аутентификации, на большинстве маршрутизаторов аутентификация не включена.

Внешние протоколы маршрутизации, используемые для соединения маршрутизаторами различных AS, обычно называются внешними шлюзовыми протоколами (EGP – Exterior Gateway Protocol). BGP (Border Gateway Protocol – Граничный шлюзовой протокол) позволяет маршрутизаторам различных AS совместно использовать информацию маршрутизации для обеспечения оптимальной и эффективной маршрутизации между различными сетями. BGP обычно применяется интернет-провайдерами для маршрутизации данных в Интернете из одного места в другое.

ПРИМЕЧАНИЕ. Ранее существовал внешний протокол маршрутизации, который назывался EGP, но он был повсеместно заменен на BGP и сейчас термин «граничный шлюзовой протокол» и аббревиатура «EGP» используются в основном для указания типа протокола, а не самого протокола.

BGP использует комбинацию алгоритмов дистанционно-векторной маршрутизации и маршрутизации по состоянию канала. Он строит базу данных сетевой топологии, используя для этого свою функциональность контроля состояния канала, и передает обновления таблицы маршрутизации на периодической основе, аналогично протоколам дистанционно-векторной маршрутизации, а не непрерывно. Сетевой администратор может указать вес для различных переменных, используемых этим протоколом для контроля состояния канала при выборе наилучшего маршрута. Эти настройки в общем виде называются *политикой маршрутизации* (routing policy).

Существует несколько типов атак на маршрутизаторы, которые проводятся посредством используемых ими протоколов маршрутизации. Целью большинства таких атак является нарушение маршрутов трафика путем использования ложных ICMP-сообщений. Атакующий может выдавать свой компьютер за другой маршрутизатор и изменить с помощью него таблицу маршрутизации на маршрутизаторе-жертве. После применения маршрутизатором этой ложной информации, он может начать отправлять трафик неправильным подсетям или компьютерам, либо вообще несуществующим адресам (черная дыра). Эти атаки чаще всего оказываются успешными, если на маршрутизаторе не включена аутентификация для протокола маршрутизации. Если аутентификация не выполняется, маршрутизатор принимает и применяет обновления таблиц маршрутизации, не зная, является ли отправитель этих обновлений легитимным маршрутизатором. Атакующий может изменить маршруты трафика компании для получения конфиденциальной информации или просто нарушения потока трафика, что будет аналогично DoS-атаке.

Существуют и другие типы DoS-атак, такие как флудинг порта маршрутизатора, переполнение буфера, SYN-флуд. Существует множество различных типов атак, а также множество соответствующих им контрмер, о которых следует знать для эффективного противодействия этим атакам. Большинство этих контрмер используют аутентификацию и шифрование передаваемых данных маршрутизации с использованием общих ключей или IPSec. Хорошее описание этих атак и соответствующих контрмер приведено в документе SAFE компании Cisco: «SAFE: Best Practices for Securing Routing Protocols».

Атака «червоточина» (wormhole). Атакующий может перехватить пакеты в одном месте в сети и посредством туннеля передать их в другое место в сети. В атаке такого типа участвуют двое атакующих – по одному на каждом конце туннеля. Например, один атакующий может перехватить аутентификационный токен, отправленный на сервер аутентификации, и передать этот токен другому атакующему, который может воспользоваться им для получения несанкционированного доступа к ресурсу. Это может происходить как в проводных, так и в беспроводных сетях, но проще делать это в беспроводных сетях, поскольку атакующему для этого не нужно физически подключаться к сетевому проводу.

Контрмерой для этого типа атак является использование «поводка» (leash), который реально является определенными данными, помещенными в заголовок отдельных пакетов. Такой «поводок» ограничивает максимальное расстояние передачи пакетов. «Поводок» может быть географическим, гарантируя, что пакет остается в пределах определенного расстояния от отправителя, либо временным, ограничивая время жизни пакета. Это все равно, что одеть ошейник на собаку, чтобы она не смогла отойти от вас дальше, чем на метр.

6. Сетевые устройства

В средах LAN, MAN, WAN используется большое количество различных типов устройств, обеспечивающих взаимодействие между компьютерами и сетями. Применение тех или иных устройств зависит от их функциональности, возможностей, интеллектуальности и положения в сети. Мы рассмотрим следующие устройства:

- Повторители
- Мосты
- Маршрутизаторы
- Коммутаторы

6.1. Повторители

Повторитель (repeater) обеспечивает простейший тип связи, поскольку он только повторяет и усиливает электрические сигналы между сегментами кабеля, позволяя расширить сеть. Повторители работают на физическом уровне и являются дополнительными устройствами, позволяющими расширить сеть, увеличив расстояние, на которое могут передаваться данные. Это устройство усиливает сигналы, затухающие при прохождении по сети.

Повторители могут также работать в качестве стабилизаторов, очищая сигнал от помех. При усилении цифрового сигнала это работает намного лучше, чем при усилении аналогового сигнала, т.к. цифровой сигнал является дискретным, что упрощает извлечение из него фонового шума. При усилении аналогового сигнала, вместе с ним также прекрасно усиливаются все сопутствующие шумы, что может привести к разрушению сигнала.

Концентратор (hub, хаб) – это многопортовый повторитель. Концентратор является физическим коммуникационным устройством, позволяющим нескольким компьютерам и другим устройствам взаимодействовать друг с другом. Концентратор не понимает и не работает с IP- или MAC-адресами. Когда одна система отправляет сигнал другой системе, подключенной к концентратору, сигнал рассылается широковещательно на все порты и, соответственно, на все системы, подключенные к этому концентратору.

6.2. Мосты

Мост (bridge) – это устройство, используемое для соединения сегментов LAN. Мост работает на канальном уровне и поэтому использует MAC-адреса. Повторитель не использует адреса – он просто пересылает все получаемые им сигналы. Когда кадр поступает на мост, мост определяет, принадлежит ли его MAC-адрес локальному сетевому сегменту. Если MAC-адрес не принадлежит локальному сетевому сегменту, он пересылает кадр в нужный ему сетевой сегмент.

Мост используется для разделения перегруженной сети на небольшие сегменты, для обеспечения наилучшего использования полосы пропускания и управления трафиком. Мост усиливает электрические сигналы, также как это делает повторитель, но он интеллектуальнее повторителя. Мост применяется для расширения LAN, а также позволяет администратору фильтровать кадры, определяя какой кадр куда отправить.

При использовании мостов вам нужно внимательно следить за **«широковещательными штормами»** (broadcast storm). Поскольку мосты пересылают весь трафик, они будут пересылать и все широковещательные пакеты. Это может перегрузить сеть в результате «широковещательного шторма», что приведет к деградации пропускной способности и производительности сети.

Используется три основных типа мостов: локальный, удаленный и транслирующий. **Локальный мост** (local bridge) соединяет два или более сегментов LAN в пределах локальной сети. **Удаленный мост** (remote bridge) может соединить два и более сегментов LAN через MAN, используя телекоммуникационные каналы. Удаленный мост оборудуется телекоммуникационными портами, позволяющими ему соединить два и более сегментов LAN, которые разделяет большое расстояние. При этом такой мост может использовать, в том числе, телефонные линии. **Транслирующий мост** (translation bridge) необходим для соединения двух LAN различных типов, использующих разные стандарты и протоколы. Например, представим себе соединение между сетью Token Ring и сетью Ethernet. Кадры в сети каждого типа имеют разный размер, поля в них содержат информацию различных протоколов, а сами сети работают на разных скоростях. Если между ними будет установлен обычный мост, кадры Ethernet попадут в сеть Token Ring и наоборот, и никто не сможет понять сообщения, пришедшие из другого сетевого сегмента. Транслирующий мост, в соответствии со своим именем, транслирует пакеты между двумя сетями различных типов.

Ниже приведен список функций моста:

- Сегментирует большую сеть на небольшие и лучше управляемые
- Использует фильтрацию на основе MAC-адресов
- Объединяет различные типы сетевых соединений, сохраняя тот же широковещательный домен
- Изолирует коллизийные домены в рамках одного широковещательного домена
- Может обеспечивать функции моста как локально – в пределах одной LAN, так и для соединения территориально удаленных LAN
- Может обеспечивать трансляцию кадров между протоколами различных типов

ПРИМЕЧАНИЕ. Не путайте мосты и маршрутизаторы. Маршрутизаторы работают на сетевом уровне и фильтруют пакеты на основе IP-адресов, тогда как мосты работают на канальном уровне и фильтруют кадры на основе MAC-адресов. Маршрутизаторы обычно не пропускают широковещательные пакеты, а мосты – пропускают.

Таблицы пересылки

Мост должен знать, как передать кадр получателю, для этого он должен знать на какой порт отправить кадр и где находится система получателя. Раньше сетевые администраторы

вводили в мосты статические маршруты, указывая, как они должны передавать кадры, направленные различным получателям. Это была очень нудная задача, часто приводившая к ошибкам. Сегодня мосты используют *прозрачную маршрутизацию* (transparent bridging).

При использовании прозрачной маршрутизации, мост начинает изучать сетевую среду сразу после включения, а также после изменений в сети. Для этого мост анализирует кадры и делает записи в таблицах пересылки (forwarding table). Когда мост получает кадр от нового компьютера-отправителя, он связывает адрес этого отправителя и порт, на который пришел кадр от него. Мост делает это для всех компьютеров, отправляющих кадры в сеть. В конечном счете, мост узнает адреса всех компьютеров в различных сетевых сегментах, а также к какому порту каждый из них подключен. Если мост получает кадр, который должен быть отправлен получателю, отсутствующему в его таблице пересылки, он отправляет специальный запрос в каждый сегмент сети, за исключением сегмента, откуда пришел кадр. Только компьютер получателя отвечает на этот запрос. Получив ответ, мост обновляет свою таблицу пересылки, указывая в ней новый компьютер и порт, к которому он подключен, а затем пересылает ему кадр.

Многие мосты используют *алгоритм связующего дерева* (STA – Spanning Tree Algorithm), повышающий их интеллектуальность. STA обеспечивает, что кадры не циркулируют по сети вечно, предоставляет избыточные пути в случае отключения моста, присваивает уникальные идентификаторы каждому мосту, присваивает значения приоритета каждому мосту, рассчитывает стоимость маршрутов. Это позволяет реализовать более эффективный процесс пересылки кадров каждым мостом. STA также позволяет администратору указать предпочтительные маршруты передачи трафика.

Если используется *маршрутизация от источника* (source routing) вместо прозрачной маршрутизации, кадры содержат всю необходимую информацию, чтобы сообщить мосту, куда их нужно отправить. Кадры содержат достаточную информацию для их пересылки, поэтому мостам и маршрутизаторам не нужно навязывать таким кадрам свой вариант маршрута. Но как компьютеру-отправителю узнать правильный маршрут до получателя, если он хочет самостоятельно указать маршрут пересылки и не зависеть от моста? Для этого компьютер-отправитель сначала отправляет специальные пакеты-исследователи, которые доставляются на компьютер-получатель. Такие пакеты содержат информацию о маршруте до пункта назначения, включая информацию о том, через какие мосты нужно проходить. После получения этих пакетов компьютером-получателем, он отправляет их обратно, а компьютер-отправитель извлекает из них информацию маршрутизации, вставляет ее в свой кадр и отправляет его компьютеру-получателю.

Вопросы и ответы.

Вопрос: В чем заключается отличие между двумя LAN, соединенными мостом, от двух LAN, соединенных маршрутизатором?

Ответ: Если две LAN соединены мостом, эти LAN, по сути, были просто расширены, поскольку обе они находятся в одном широковещательном домене. Маршрутизатор может быть настроен таким образом, чтобы он не пересылал широковещательные данные, поэтому результатом соединения двух LAN посредством маршрутизатора является объединенная сеть (internetwork). Объединенная сеть – это группа сетей, объединенных таким способом, который позволяет любому узлу в любой сети взаимодействовать с любым другим узлом. Интернет является примером объединенной сети.

ПРИМЕЧАНИЕ. Внешним устройствам и граничным маршрутизаторам не следует принимать пакеты с информацией маршрутизации от источника в заголовке, т.к. эта информация будет перекрывать информацию в таблицах пересылки и маршрутизации этих промежуточных устройств. Поскольку вы хотите управлять прохождением трафика по сети, вам не нужно, чтобы пакеты могли направляться куда хотят. Маршрутизация от источника может использоваться атакующим для обхода правил фильтрации моста и маршрутизатора.

Ссылки по теме:

- IETF Bridge MIB Charter

- Ethernet Bridges and Switches, explanation by Gorrry Fairhurst

6.3. Маршрутизаторы

Мы передвигаемся вверх по уровням модели OSI в процессе обсуждения различных сетевых устройств. Повторители работают на физическом уровне, мосты – на канальном, а маршрутизаторы – на сетевом уровне. Устройства на каждом вышестоящем уровне более интеллектуальны и функциональны и глубже заглядывают в кадры. Повторитель смотрит только на электрический сигнал. Мост смотрит на MAC-адрес в заголовке. Маршрутизатор снимает с кадра первый (внешний) заголовок и заглядывает в кадр глубже в поисках IP-адреса и другой информации маршрутизации. Чем выше уровень устройства в модели OSI, тем глубже устройство может изучить кадр и принять больше решений на основе содержащейся в нем информации. Позже мы увидим, что шлюзы (gateway) могут просмотреть весь кадр вплоть до прикладного уровня, а не только адреса и информацию маршрутизации.

Маршрутизаторы (router) работают на третьем (сетевом) уровне. Они используются для соединения похожих или различных сетей (например, с помощью них можно соединить две сети Ethernet или сеть Ethernet с сетью Token Ring). Маршрутизатор – это устройство, которое имеет два или более интерфейсов и таблицу маршрутизации, с помощью которой он принимает решения о том, как отправлять пакеты по назначению. Маршрутизатор может фильтровать трафик на основе списков контроля доступа (ACL), фрагментировать пакеты при необходимости. Поскольку маршрутизаторы больше знают о сети, они могут выполнять более высокоуровневые функции, такие как расчет самого короткого и экономичного пути между узлами отправителя и получателя.

Маршрутизатор получает информацию о маршрутах и изменениях в сети с помощью протоколов маршрутизации (RIP, BGP, OSPF и т.п.). Эти протоколы сообщают маршрутизатору об отключениях каналов, перегрузках маршрутов, а также уведомляют о других, более экономичных, маршрутах. Также, с помощью этих протоколов обновляются таблицы маршрутизации и передаются сообщения о возникновении у маршрутизатора проблем или его отключении.

Мост использует одни и те же сетевые адреса для всех своих портов, а у маршрутизатора используются отдельные адреса для каждого порта, что позволяет ему соединять различные сети.

Маршрутизатор может быть выделенным устройством (appliance) или «двухканальным» (dual-homed) компьютером с сетевой операционной системой. Когда пакеты поступают на один из интерфейсов, маршрутизатор проверяет параметры этих пакетов по своим спискам ACL. Эти списки показывают, какие пакеты считаются разрешенными, а какие – нет. При этом решение принимается на основе IP-адресов и портов отправителя и получателя, типа протокола. Например, администратор может блокировать все пакеты, исходящие из сети 10.10.12.0, пакеты FTP, а также любые пакеты, отправленные на порт 80 компьютера с адресом 10.10.15.1. Все это реализуется с помощью ACL, которые администратор должен запрограммировать и по мере необходимости обновлять.

Что в действительности происходит внутри маршрутизатора, когда он получает пакет? Давайте пройдем по шагам:

1. Кадр принят одним из интерфейсов маршрутизатора. Маршрутизатор анализирует данные маршрутизации.
2. Маршрутизатор отыскивает в датаграмме IP-адрес (подсеть) получателя.
3. Маршрутизатор просматривает свою таблицу маршрутизации в поисках своего порта, соответствующего IP-адресу (подсети) получателя.
4. Если маршрутизатор не находит в своей таблице информации об адресе получателя,

он отправляет компьютеру-отправителю ICMP-сообщение об ошибке, говоря ему, что не может найти получателя.

5. Если маршрутизатор находит нужный маршрут в своей таблице маршрутизации, он уменьшает значение TTL и смотрит, не отличается ли значение MTU в сети получателя. Если в сети получателя требуется MTU меньшего размера, маршрутизатор производит фрагментацию датаграммы.
6. Маршрутизатор изменяет информацию в заголовке кадра, чтобы он смог пройти к следующему маршрутизатору или компьютеру-получателю (если маршрутизатор непосредственно подключен к сети, в которой находится компьютер-получатель).
7. Маршрутизатор направляет кадр в очередь на отправку соответствующего своего интерфейса.

В Таблице 5-6 представлен краткий обзор отличий между маршрутизатором и мостом.

Мост	Маршрутизатор
Читает информацию из заголовка, но не изменяет ее	Создает новый заголовок для каждого кадра
Создает таблицы пересылки на основе MAC-адресов	Создает таблицы маршрутизации на основе IP-адресов
Использует один сетевой адрес для всех портов	Присваивает всем портам различные сетевые адреса
Фильтрует трафик на основе MAC-адресов	Фильтрует трафик на основе IP-адресов
Пересылает широковещательные пакеты	Не пересылает широковещательные пакеты
Пересылает трафик, если целевой адрес неизвестен мосту	Не пересылает трафик, если целевой адрес не известен маршрутизатору

Таблица 5-6. Основные различия между мостами и маршрутизаторами

В каких случаях какое устройство является наилучшим выбором? Повторитель применяется в тех случаях, когда администратору нужно расширить сеть и усилить сигналы, чтобы они не затухали в длинном кабеле. Однако повторитель пересылает информацию о коллизиях и широковещательную информацию, поскольку не имеет достаточного интеллекта, чтобы отличать различные виды трафика.

Мосты работают на канальном уровне и имеют немного больше интеллекта, чем повторители. Мосты могут выполнять простую фильтрацию, они разделяют коллизионные, но не широковещательные домены. Мост следует использовать в тех случаях, когда администратору нужно разделить сеть на сегменты для снижения нагрузки от трафика и снижения количества коллизий.

Маршрутизатор разделяет сеть на коллизионные домены и широковещательные домены. Маршрутизатор позволяет более четко разделить сеть на сегменты, чем повторители или мосты. Если администратор хочет иметь возможность более четкого управления прохождением трафика, ему следует использовать маршрутизатор, поскольку маршрутизатор позволяет настроить более сложные правила фильтрации, а если маршрутизатор используется для деления сети на сегменты, результат может быть более контролируемым.

Маршрутизаторы применяются для деления сети по подразделениям, рабочим группам или иным бизнес-группам. Мосты применяются для деления сети на сегменты на основе видов трафика и загруженности.

Ссылки по теме:

- Internetworking Technology Handbook, Chapter 5, “Routing Basics,” Cisco Systems, Inc.
- Active IETF Working Groups

6.4. Коммутаторы

Коммутаторы (switch) сочетают в себе функциональность повторителей и мостов. Коммутатор усиливает электрический сигнал, как повторитель, и имеет встроенные

компоненты и интеллектуальность моста. Это многопортовое устройство, позволяющее соединить отдельные компьютеры или другие коммутаторы и концентраторы. Любое устройство, подключенное к одному из портов коммутатора, может взаимодействовать с другими устройствами, подключенными к другим его портам, используя при этом собственные виртуальные частные соединения. В чем заключается отличие коммутаторов от концентраторов или мостов? Когда концентратор получает кадр, он отправляет его по всем своим портам. Когда кадр получает мост, он отправляет его в порт, к которому подключен сетевой сегмент получателя. Когда кадр получает коммутатор, он отправляет его напрямую компьютеру-получателю (или сети получателя), что позволяет уменьшить трафик. На Рисунке 5-28 показана конфигурация сети, в которой компьютеры напрямую подключены к соответствующим коммутаторам.

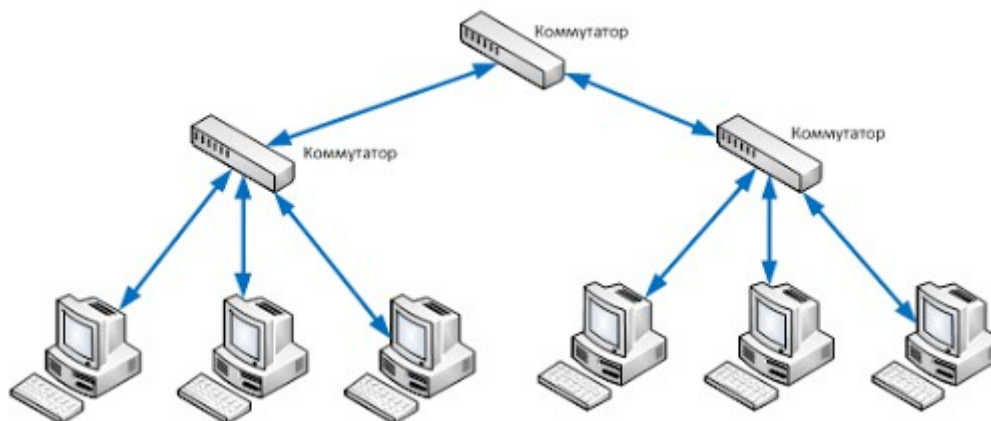


Рисунок 5-28. Коммутаторы позволяют устройствам взаимодействовать друг с другом посредством собственных виртуальных соединений

В сетях Ethernet компьютеры используют общую сетевую среду. Каждый компьютер должен прослушивать сетевую активность и передавать данные, когда он думает, что сетевая среда свободна. Конфликты и коллизии вызывают задержки трафика и повышают загрузку полосы пропускания. Если используются коммутаторы, конфликты и коллизии перестают быть проблемой, что повышает эффективность использования полосы пропускания и снижает латентность сети. Коммутаторы снижают или полностью устраняют совместное использование сетевой среды, исключая связанные с этим проблемы.

Коммутатор – это многопортовое устройство сопряжения, каждый порт которого предоставляет выделенную полосу пропускания для подключенного к нему устройства. Порт соединяется с другим портом, что позволяет двум устройствам создать частное соединение точка-точка. Коммутатор применяет полнодуплексную связь, одна пара проводов в нем используется для отправки, а другая – для приема. Это позволяет двум подключенным устройствам не соревноваться за использование общей полосы пропускания.

Обычные коммутаторы работают на канальном уровне и пересылают трафик на основе MAC-адресов. Однако на сегодняшний день существуют коммутаторы, работающие на 3, 4 и других уровнях, и обеспечивающие более широкую функциональность, чем коммутаторы 2 уровня. Такие высокоуровневые коммутаторы зачастую обладают функциями маршрутизации, анализа пакетов, приоритизации трафика, функциональностью QoS. Эти коммутаторы называют многоуровневыми (multilayered switch), поскольку они сочетают в себе функциональность канального, сетевого и других уровней.

Многоуровневые коммутаторы обладают большой вычислительной мощностью, что позволяет им глубже анализировать сетевые пакеты, принимать больше решений на основе такого анализа, выполнять маршрутизацию и задачи управления трафиком. Часто такой объем работы приводит к некоторым перегрузкам и задержкам передачи трафика, но многоуровневые коммутаторы выполняют эти действия с помощью ASIC (Application

Specific Integrated Circuit – Специализированная интегральная микросхема). Это означает, что большинство функций коммутатора выполняется на аппаратном уровне (в чипе), что значительно быстрее, чем выполнение этих функций на программном уровне.

Коммутаторы уровней 3 и 4

Интеллектуальности коммутаторов второго уровня достаточно только для пересылки кадров на основе MAC-адресов, но они не имеют представления о сети в целом.

Коммутаторы третьего уровня обладают интеллектуальностью маршрутизатора. Они не только могут маршрутизировать пакеты на основе их IP-адресов, они также могут выбирать маршруты на основе их доступности и производительности. Коммутаторы третьего уровня – это «продвинутые» маршрутизаторы, т.к. они переводят функции анализа маршрутов на более эффективный аппаратный уровень.

Основным различием между коммутаторами 2, 3 и 4 уровней является информация заголовка, которую они анализируют для принятия решения о пересылке или маршрутизации. Коммутаторы 3 и 4 уровней могут использовать теги, которые связаны с каждой целевой сетью или подсетью. Когда пакет приходит на коммутатор, он сравнивает адрес получателя со своей Базой информации о тегах (Tag Information Base), которая является списком всех подсетей и соответствующих им номеров тегов. Коммутатор добавляет тег к пакету и отправляет его следующему коммутатору. Все коммутаторы между первым (ближайшим к отправителю) коммутатором и узлом получателя просто просматривают информацию о теге для выбора маршрута, а не анализируют весь заголовок. Когда пакет приходит на последний (ближайший к получателю) коммутатор или маршрутизатор, этот тег удаляется и пакет отправляется получателю. Использование тегов увеличивает скорость маршрутизации пакетов от одного места к другому.

Использование такого типа тегов, называемое MPLS (Multiprotocol Label Switching - Мультипротокольная коммутация по меткам), позволяет не только увеличить скорость маршрутизации, но и учесть требования по уровню сервиса для различных типов пакетов. Некоторый чувствительный ко времени трафик (такой как видео-конференции) требует определенного уровня сервиса (QoS), гарантирующего минимальную скорость доставки, которая соответствует требованиям приложения или пользователя. При использовании MPLS, информация о приоритетах указывается в тегах, помогая обеспечить чувствительному ко времени трафику более высокий приоритет, чем всему остальному, как показано на Рисунке 5-29.

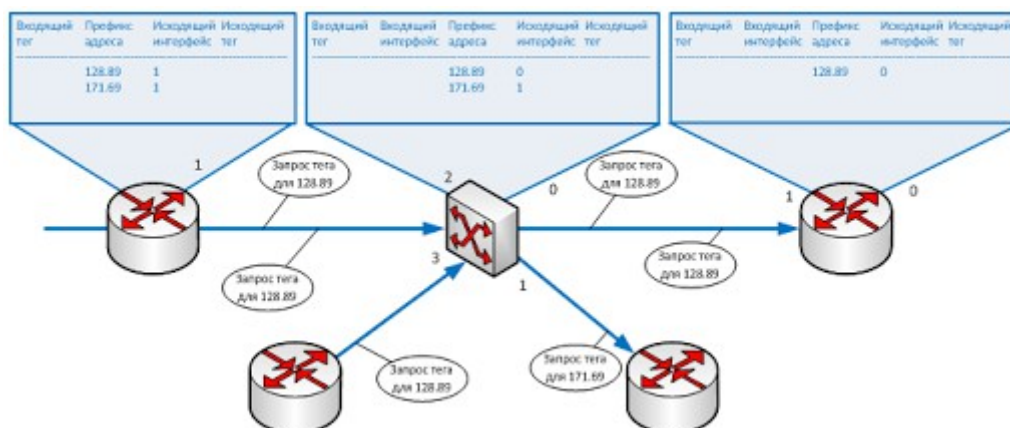


Рисунок 5-29. MPLS используется для чувствительного ко времени трафика

Многие компании в настоящее время используют коммутируемые сети, в которых компьютеры подключены к выделенным портам Ethernet-коммутаторов, Gigabit Ethernet-коммутаторов, АТМ-коммутаторов и т.д. Текущий уровень развития коммутаторов, расширение их функциональности, возможности выполнения ими функций повторителей, мостов и маршрутизаторов, сделали коммутаторы важной частью современного сетевого

мира.

Поскольку безопасность требует контроля доступа к определенным ресурсам, более интеллектуальные устройства обеспечивают более высокий уровень защиты, т.к. они могут принять больше ориентированных на детали решений о доступе к ресурсам. Если устройство может заглянуть глубже в пакеты, оно может узнать больше информации для принятия решения о возможности предоставления доступа, что обеспечивает более детальный контроль.

Как было указано ранее, использование коммутируемых сетей создает злоумышленникам больше сложностей в части возможного перехвата и мониторинга сетевого трафика, т.к. в таких сетях широковещательная и коллизийная информация не передается постоянно через всю сеть. Коммутаторы, в отличие от других устройств предоставляют также сервисы безопасности. **Виртуальные сети** (VLAN – Virtual LAN) также являются важной частью коммутируемой сети, поскольку они позволяют администраторам получить больше возможностей для управления сетевой средой. С помощью VLAN'ов можно изолировать пользователей и группы в логические и управляемые единицы. Виртуальные сети описаны в следующем разделе.

Ссылки по теме:

- Internetworking Technology Handbook, Chapter 5, “Routing Basics,” Cisco Systems, Inc.
- Internetwork Design Guide, Chapter 12, “Designing Switched LAN Internetworks,” Cisco Systems, Inc.

Виртуальные сети (VLAN)

Технологии в лице коммутаторов позволили реализовать функциональность виртуальных сетей (VLAN). VLAN'ы позволяют администраторам разделить и сгруппировать компьютеры логически, на основе необходимых им ресурсов, безопасности или потребностей бизнеса, вместо стандартной группировки систем на основе их физического размещения, которая применяется при использовании повторителей, мостов и маршрутизаторов. Рисунок 5-30 показывает как компьютеры, физически размещенные рядом друг с другом, могут быть логически сгруппированы в различные VLAN'ы. Администраторы могут сформировать такие группы на основе потребностей пользователей и компании, а не на основе физического расположения систем и ресурсов.

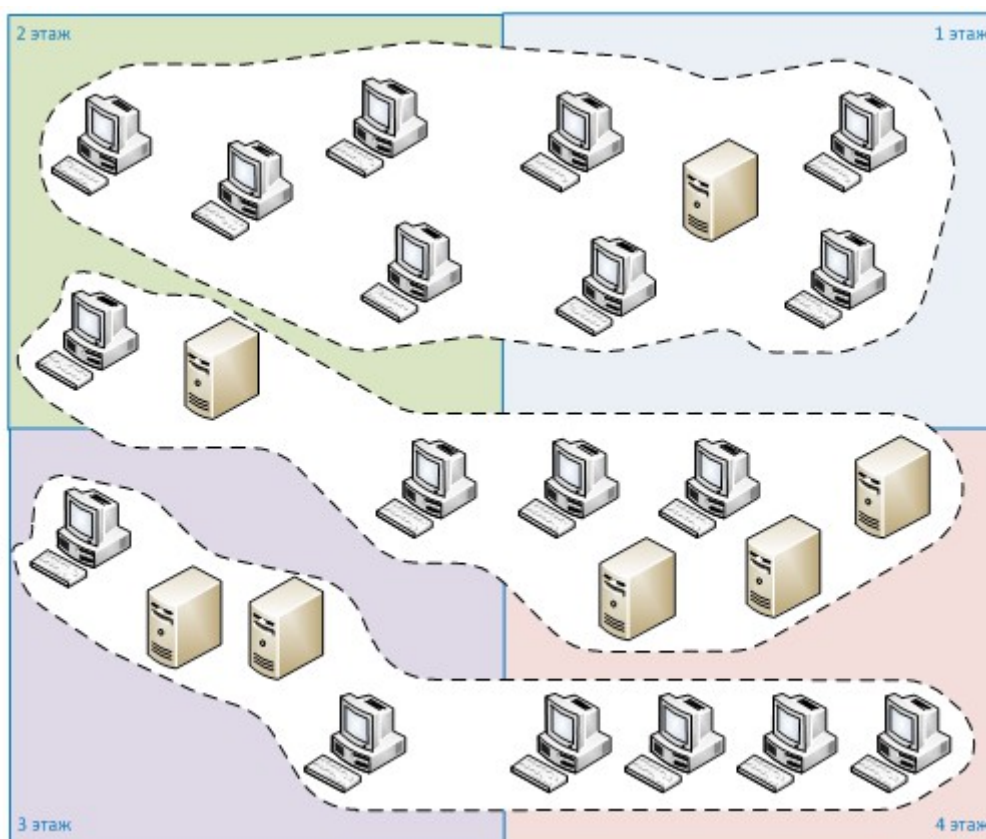


Рисунок 5-30. VLAN'ы позволяют администраторам размещать системы в логических сетях

Администратору может понадобиться, например, разместить компьютеры всех пользователей Департамента маркетинга в одном VLAN. При этом все эти пользователи будут получать одни и те же широковещательные сообщения и смогут использовать одни и те же типы ресурсов. Такое распределение позволяет администратору включить в тот же VLAN несколько пользователей, размещенных в другом здании или на другом этаже. VLAN'ы также позволяют администратору применять различные политики безопасности для различных логических групп. Например, если требуется максимальная безопасность для бухгалтерии, администратор может разработать политику, добавить все бухгалтерские системы в отдельный VLAN и применить к нему эту политику безопасности.

VLAN расположен над физической сетью, как это показано на рисунке 5-31. Если рабочая станция P1 хочет взаимодействовать с рабочей станцией D1, передаваемые между ними сообщения должны пройти маршрутизацию, даже если эти рабочие станции физически находятся в непосредственной близости друг от друга. Дело в том, что они находятся в разных логических сетях.

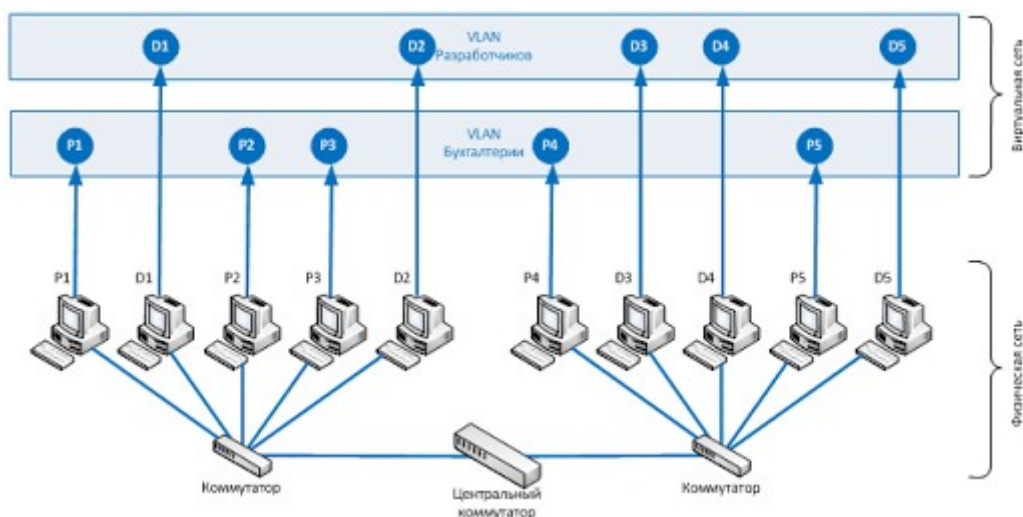


Рисунок 5-31. VLAN'ы существуют на более высоком уровне, чем физическая сеть, и не ограничены ей

6.5. Шлюзы

Термин «*шлюз*» (gateway) обычно относится к программному обеспечению, запущенному на устройстве, которое объединяет две различных среды, часто работая в качестве транслятора для них, либо некоего ограничителя их взаимодействия. Обычно шлюз требуется в случае, когда объединяемые среды «говорят на разных языках», т.е. одна среда использует протокол, который другая среда не понимает. Шлюз может транслировать пакеты протокола IPX (Internetwork Packet Exchange) в IP-пакеты, принимать почту с почтового сервера одного типа и форматировать ее таким образом, чтобы почтовый сервер другого типа мог принимать и понимать ее, либо соединять и транслировать различные каналные технологии, например, объединить среды FDDI и Ethernet.

Шлюзы выполняют гораздо более сложные задачи, чем сетевые устройства типа маршрутизаторов и мостов. Однако некоторые называют маршрутизаторы шлюзами, если они соединяют две различных сети (например, Token Ring и Ethernet), поскольку маршрутизаторы также могут транслировать технологии канального уровня. Рисунок 5-32 показывает, как функционирует сервер NAS (Network Access Server - Сервер сетевого доступа) в качестве шлюза между телекоммуникационными и сетевыми соединениями.

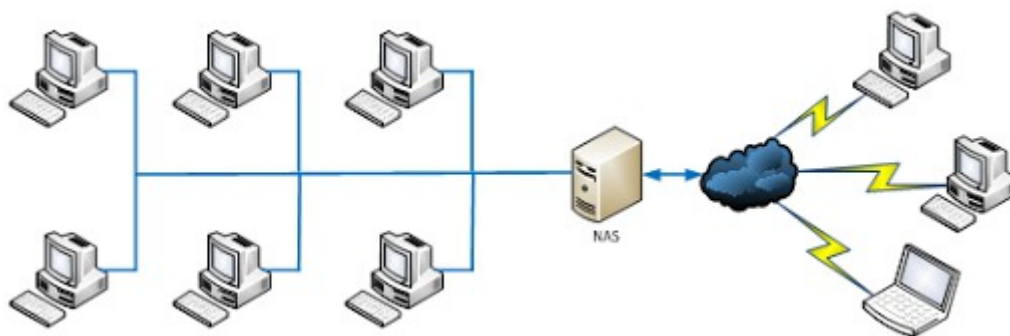


Рисунок 5-32. В сети может использоваться множество различных типов шлюзов. Один из примеров – NAS.

Если сеть подключена к магистральной, шлюз может транслировать различные технологии и форматы кадров между магистральной сетью и LAN. Если бы между магистралью FDDI и сетью Ethernet был установлен мост, компьютеры в LAN вероятно не смогли бы понять кадры протоколов FDDI. Для такой трансляции протоколов может понадобиться шлюз.

Популярным типом шлюзов являются шлюзы *электронной почты*. Поскольку многие производители систем электронной почты используют свой собственный синтаксис, формат

сообщений и способы их передачи, почтовые шлюзы требуются для преобразования сообщений между различным программным обеспечением почтовых серверов.

Предположим, что Дэвид, использующий Sendmail в своей корпоративной сети, пишет электронное письмо Дэну, в корпоративной сети которого используется Microsoft Exchange. Почтовый шлюз конвертирует это сообщение в стандарт, который понимают все почтовые серверы (обычно в X.400), и отправляет его на почтовый сервер Дэна.

Другим примером шлюза является голосовой и медиа-шлюз. Часто он используется для объединения сетей передачи голоса с сетями передачи данных. Это повышает эффективность, поскольку одна и та же среда может использоваться и для передачи голоса, и для передачи данных. Однако передача голоса осуществляется с помощью потоковых технологий, тогда как данные обычно передаются в виде пакетов. Таким образом, объединенная общая среда может взаимодействовать с двумя различными типами сетей: PSTN телефонных компаний и маршрутизаторами, работающими с пакетными данными в Интернете. Это означает, что шлюз должен разделять голосовую информацию и данные, и преобразовывать их в форму, которую может понять каждая из сетей.

В таблице 5-7 перечислены устройства, рассмотренные нами ранее в разделе «Сетевые устройства», и указаны их важнейшие характеристики.

Устройство	Уровень OSI	Функциональность
Повторитель	Физический	Усиливает сигнал и расширяет сеть
Мост	Канальный	Пересылает пакеты и фильтрует их на основе MAC-адресов; пересылает широковещательный, но не коллизионный трафик
Маршрутизатор	Сетевой	Разделяет и соединяет сети LAN, создавая объединенные сети; фильтрует трафик на основе IP-адресов
Коммутатор	Канальный	Обеспечивает частные виртуальные соединения между взаимодействующими устройствами; позволяет создавать VLAN'ы; уменьшает количество коллизий; противодействует сетевому sniffingu
Шлюз	Прикладной	Соединяет различные типы сетей; выполняет трансляцию протоколов и форматов

Таблица 5-7. Различия между сетевыми устройствами

6.6. Офисные автоматические телефонные станции

Телефонные компании используют технологии коммутации для передачи телефонных звонков вызываемым абонентам. В центральном офисе телефонной компании установлены коммутаторы, которые соединяют города и целые области посредством оптоволоконных колец. Так, например, если Петр звонит из своего дома, звонок сначала попадает в центральный офис местной телефонной компании, которая обслуживает Петра, а затем соответствующим образом переключает его на локальный или междугородний вызов.

Офисная автоматическая телефонная станция (PBX – Private Branch Exchange, Офисная АТС) – это частный телефонный коммутатор, который находится в собственности компании. Аналогичный коммутатор выполняет такие же задачи в центральном офисе телефонной компании. Офисная АТС имеет выделенное соединение с центральным офисом местной телефонной компании, где размещены более интеллектуальные коммутаторы.

Офисная АТС может выступать в качестве интерфейса между некоторыми типами устройств и предоставлять ряд телефонных услуг. В выделенной линии, подключенной к центральному офису телефонной компании, данные мультиплексируются (уплотняются). Рисунок 5-33 показывает, как данные из различных источников могут быть помещены в один канал Офисной АТС и отправлены на коммутаторы телефонной компании.



Рисунок 5-33. Офисная АТС объединяет различные типы данных в одном канале

Офисная АТС использует цифровые коммутационные устройства, которые могут управлять аналоговыми и цифровыми сигналами. Старые Офисные АТС могли работать только с аналоговыми устройствами, но в настоящее время большинство Офисных АТС являются цифровыми. Переход на цифровые системы и сигналы уменьшил количество Офисных АТС, а также количество уязвимостей безопасности телефонных систем, которые существовали ранее. Однако это не значит, что мошенничества с Офисными АТС сегодня не существует. Например, многие компании используют модемы (или другие методы доступа), позволяющие производителям звонить на них и выполнять функции поддержки и сопровождения Офисных АТС. Эти модемы часто являются незащищенным черным ходом в сеть компании. Такие модемы следует включать только при возникновении проблемы, для решения которой необходимо предоставить производителю удаленный доступ. Во все остальное время они должны быть заблокированы.

Кроме того, на многих Офисных АТС производителями устанавливается пароль администратора по умолчанию, который обязательно должен быть сменен, иначе *фрикеры* (телефонные хакеры), знающие устанавливаемые по умолчанию пароли, могут получить доступ к этим Офисным АТС. Если фрикер получит доступ к Офисной АТС, он может вызвать хаос, перемаршрутизовав звонки, или настроить систему таким образом, чтобы он и его друзья могли бесплатно (за счет компании) пользоваться междугородней связью. Такое мошенничество случается гораздо чаще, чем сообщают компании, потому что многие из них не анализируют детально свои телефонные счета.

Офисные АТС также уязвимы к брутфорс-атакам (полный перебор паролей) и другим типам атак, в которых фрикеры используют скрипты и словари, чтобы угадать необходимые реквизиты для получения доступа к системе. В некоторых случаях, фрикеры могут прослушивать и изменять голосовые сообщения людей.

6.7. Межсетевые экраны

Межсетевые экраны (firewall – фаерволлы) используются для ограничения доступа в одну сеть из другой. Большинство компаний используют межсетевые экраны для ограничения доступа из Интернет в свою сеть. Также межсетевые экраны можно использовать для ограничения доступа из одного внутреннего сетевого сегмента в другой. Например, если сетевому администратору нужно быть уверенным, что обычные сотрудники не имеют доступа в сеть Исследований и Разработки, он может установить межсетевой экран между этой сетью и всеми остальными сетями, настроив его на разрешение только одного типа трафика, который он считает приемлемым.

Межсетевой экран поддерживает и реализует политику сетевой безопасности компании. Организационная политика безопасности дает высокоуровневые инструкции по приемлемым и неприемлемым действиям, относящимся к безопасности. Межсетевой экран реализует более детальную и конкретную политику безопасности, которая определяет, каким службам и по каким портам разрешен доступ, а какие IP-адреса и диапазоны должны быть заблокированы. Межсетевой экран называют «бутылочным горлышком» сети, поскольку через него должны проходить все коммуникации для проверки и ограничения трафика.

Межсетевой экран может быть встроен в маршрутизатор или являться специализированным аппаратным устройством. Он выполняет мониторинг пакетов, которые приходят и исходят из защищаемой им сети. Он отфильтровывает пакеты, которые не соответствуют требованиям политики безопасности. В зависимости от настроек и политики безопасности межсетевой экран может уничтожать такие пакеты, переупаковывать их или перенаправлять. Фильтрация пакетов осуществляется на основе адресов их отправителя и получателя, номеров портов (сервисов), типов пакетов, типов протоколов, информации в заголовках, битов последовательности и т.д. Сетевые администраторы могут использовать специальные функции и параметры для идентификации и ограничения доступа.

Часто компании устанавливают межсетевой экран для создания **демилитаризованной зоны** (DMZ – Demilitarized Zone), которая является сетевым сегментом, размещенным между защищенными и незащищенными сетями. С помощью DMZ реализуется буферная зона между опасным Интернетом и ценными активами внутренней сети, которые компания пытается защитить. Как показано на Рисунке 5-34, для создания DMZ обычно устанавливаются два межсетевых экрана. В DMZ обычно размещаются веб, почтовые и DNS-серверы, которые должны быть надлежащим образом защищены, поскольку они являются первой линией обороны от атак. Часто в DMZ размещают сенсоры IDS, которые выявляют нежелательное и подозрительное поведение.

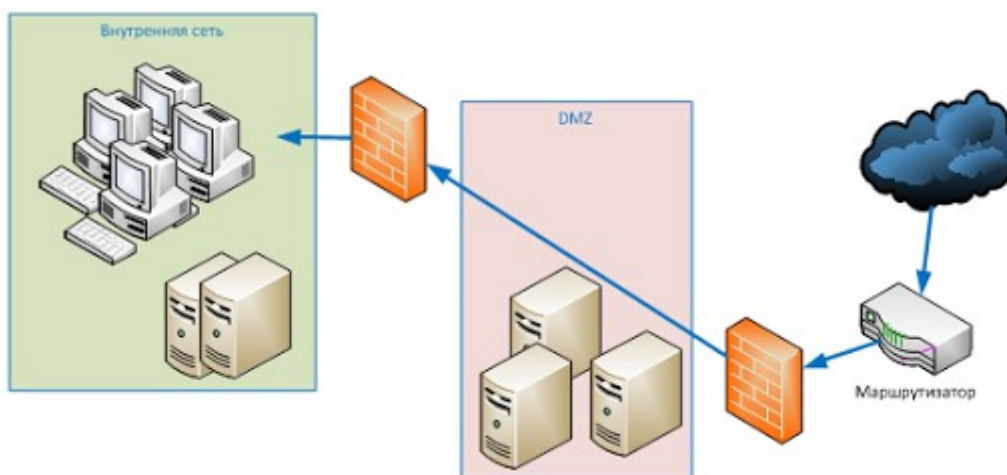


Рисунок 5-34. Для создания DMZ обычно используется не менее двух межсетевых экранов (или интерфейсов одного межсетевого экрана)

Существует множество различных типов межсетевых экранов, поскольку каждая сеть может иметь уникальные требования и цели безопасности. Межсетевые экраны развиваются и эволюционируют, повышается их эффективность и функциональность. Далее мы поговорим о следующих типах межсетевых экранов:

- С фильтрацией пакетов (packet filtering)
- С контролем состояния (stateful)
- Прокси (proxy)
- С динамической фильтрацией пакетов (dynamic packet filtering)
- Прокси ядра (kernel proxy)

Затем мы разберем три основные архитектуры межсетевых экранов:

- Экранированный узел (screened host)
- Двойная привязка (dual-home)
- Экранированная подсеть (screened subnet)

Межсетевые экраны с фильтрацией пакетов

Фильтрация пакетов (packet filtering) – это метод обеспечения безопасности, позволяющий контролировать, какие данные могут войти в сеть и/или выйти из нее. Пакетная фильтрация использует списки контроля доступа (ACL), настроенные на устройстве. ACL – это строки текста, называемые *правилами*, которые устройство применяет к каждому полученному им пакету. Эти правила четко определяют, какие пакеты могут быть разрешены, а какие должны быть отклонены. К примеру, в первой строке ACL может быть указано, что любые пакеты, приходящие с IP-адреса из диапазона 172.168.0.0, должны быть отклонены; во второй строке может быть запрещен доступ в сеть для пакетов Telnet; в третьей строке может быть запрещен любой трафик на порт 443; в четвертой строке может быть указано разрешение трафика на 80-й порт, который должен при этом маршрутизироваться на определенный адрес, принадлежащий веб-серверу. При получении каждого пакета, устройство сравнивает информацию из заголовка пакета с каждой строкой в ACL. При этом, основываясь на приведенном выше примере правил, если пакет использует протокол Telnet или запрашивает соединение с портом 443, он отклоняется. Если информация в заголовке пакета указывает, что он хочет взаимодействовать через порт 80 по протоколу HTTP через TCP, пакет принимается и перенаправляется на веб-сервер.

Такая фильтрация основана на информации сетевого уровня, что означает, что это устройство не может очень глубоко анализировать сам пакет. Оно может принимать решения только на основе ограниченной информации заголовка. Большинство маршрутизаторов используют ACL, работая в качестве межсетевого экрана, и используют таблицу маршрутизации для принятия решений о маршрутизации пакетов, но они не могут предоставить того уровня защиты, который обеспечивают другие межсетевые экраны, глубже анализирующие пакеты. Поскольку при фильтрации пакетов анализируется только информация заголовка, такие устройства не зависят от конкретного приложения, также как и многие прокси. Межсетевые экраны с фильтрацией пакетов не отслеживают состояние соединения, в отличие от межсетевых экранов с контролем состояния, которые рассмотрены далее. Фильтрация пакетов – это метод, используемый первым поколением межсетевых экранов. Другие методы были разработаны позднее и использованы в следующих поколениях межсетевых экранов.

Ниже приведены некоторые слабости межсетевых экранов с фильтрацией пакетов:

- Они не могут предотвратить атаки, которые используют функции и уязвимости, относящиеся к конкретным приложениям (на прикладном уровне).
- Функциональность журналирования в межсетевых экранах с фильтрацией пакетов ограничена.
- Большинство межсетевых экранов с фильтрацией пакетов не поддерживают современные схемы аутентификации пользователей.
- Многие межсетевые экраны с фильтрацией пакетов не могут выявить сетевые пакеты, в которых информация 3-го уровня модели OSI была изменена (является ложной) (spoofed).
- Используя небольшое количество переменных при принятии решения о доступе, межсетевые экраны с фильтрацией пакетов подвержены проблемам безопасности, вызванным неправильными настройками.

Преимущества использования межсетевых экранов с фильтрацией пакетов заключаются в их масштабируемости, независимости от приложений и их высокой производительности, т.к. они не выполняют интенсивной обработки пакетов.

Межсетевые экраны с контролем состояния

При использовании фильтрации пакетов, каждый пришедший пакет маршрутизатор проверяет по своему ACL, чтобы решить, должен ли пакет быть принят или отклонен. Если пакет разрешен, он передается на узел получателя или другой маршрутизатор, после чего маршрутизатор забывает про этот пакет. В этом заключается отличие от фильтрации с контролем состояния, при которой маршрутизатор «помнит» и отслеживает проходящие пакеты до тех пор, пока не будет закрыто соединение, в рамках которого они передавались.

Межсетевой экран, реализующий фильтрацию пакетов *с контролем состояния* (stateful-inspection firewall), похож на соседа с длинным носом, который все слышит, следит – кто и что делает, постоянно встает в разговоры. Он знает кто, что и кому сказал. Это может сильно раздражать, но если дом вдруг ограбят, помощь этого соседа будет просто неоценима. Он все знает и сможет рассказать полиции, что произошло. Аналогичным образом работает межсетевой экран с контролем состояния. Он также отслеживает, что компьютеры говорят друг другу, и хранит всю эту информацию в *таблице состояний* (state table), которая похожа на лист с записями чужих разговоров.

Межсетевые экраны с контролем состояния также принимают решения о разрешении или отвержении пакетов, но их функциональность находится на шаг дальше. Например, обычное устройство с фильтрацией пакетов может блокировать любые UDP-пакеты, запрашивающие сервис на 25-м порту, а устройство с фильтрацией пакетов и контролем состояния можно настроить таким образом, чтобы оно разрешало UDP-пакеты только в том случае, если они являются ответами на направленные ранее исходящие запросы. Например, Майкл отправляет запрос на компьютер в другой сети. Этот запрос записывается в таблицу состояний меж сетевого экрана, который «запоминает», что компьютер Майкла сделал запрос и что ему должен прийти ответный пакет, который нужно переслать Майклу. Когда запрошенный Майклом компьютер в Интернете отвечает на его запрос, эти пакеты сравниваются с записями в таблице состояний меж сетевого экрана. Поскольку в таблице состояний есть информация о предыдущем запросе Майкла, межсетевой экран принимает эти пакеты и пропускает их дальше – Майклу. С другой стороны, если Майкл не делал никаких запросов, а пакеты все же пришли из Интернет, межсетевой экран не найдет в своей таблице состояний информации о предыдущем запросе, и тогда он будет проверять эти пакеты по своей ACL, чтобы решить, следует ли пропускать эти пакеты в сеть.

Обычная фильтрация пакетов сравнивает входящие пакеты с правилами, указанными в ACL меж сетевого экрана. Когда пакеты принимает межсетевой экран с контролем состояния, он сначала анализирует свою таблицу состояний, чтобы проверить, было ли уже установлено соответствующее соединение и были ли запрошены эти данные. Если он не находит уже установленного соединения или запроса, пакет проверяется по ACL меж сетевого экрана. Если в соответствии с ACL такой тип трафика разрешен, пакету разрешается доступ в сеть. В противном случае пакет уничтожается.

Большинство межсетевых экранов с контролем состояния работают на сетевом и транспортном уровнях. Это зависит от конкретного продукта, но часто при установлении соединения, межсетевой экран исследует все уровни пакета (все заголовки, информацию и окончания). Если инициирующие соединение пакеты прошли такое глубокое исследование, межсетевой экран в дальнейшем просто анализирует сетевой и транспортный заголовки, чтобы сохранить информацию о сеансе. Такое уменьшение глубины проверки увеличивает производительность.

Хотя контроль состояния является большим шагом вперед в безопасности, он добавляет дополнительные сложности, т.к. устройство должно теперь хранить динамическую таблицу состояний и «помнить» соединения. К сожалению, межсетевые экраны с контролем состояния подвержены многим типам DoS-атак. Многие типы атак против таких межсетевых экранов направлены на переполнение (flooding) таблицы состояний ложной информацией. Таблица состояний – это такой же конечный ресурс, как пространство жесткого диска

системы, объем свободной памяти и процессорное время. Если таблица состояний полностью заполняется ложной информацией, устройство может зависнуть или перезагрузиться. Кроме того, если такой межсетевой экран перезагрузится по какой-либо причине, это приведет к потере им информации об установленных соединениях, из-за чего он будет отклонять легитимные пакеты.

Характеристики межсетевого экрана с контролем состояния. Ниже представлены некоторые важные характеристики межсетевого экрана с контролем состояния:

- Отслеживает каждый коммуникационный канал, записывая информацию в таблицу состояний
- Обеспечивает высокий уровень безопасности и не наносит большого ущерба производительности (по сравнению с прокси прикладного уровня)
- Является масштабируемым и прозрачным для пользователя
- Предоставляет данные для отслеживания протоколов без установления соединений, таких как UDP и ICMP
- "Запоминает" и актуализирует состояние и контекст данных в пакетах
- Считается межсетевым экраном третьего поколения

Прокси

Прокси (проху) – это посредник. Он перехватывает и исследует сообщения перед тем, как они будут доставлены их получателям. Представьте, что вам нужно передать коробку и сообщение президенту США. Вы не можете просто подойти к президенту и передать все ему лично. Вам нужно будет пройти через посредника в лице Секретной службы, которая примет от вас коробку и сообщение, проверит, что внутри коробки нет ничего опасного, а уже затем передаст ее вместе с вашим сообщением президенту. То же самое делает прокси – он принимает входящее в сеть или исходящее из сети сообщение, проверяет его на наличие вредоносной информации и, если он решает, что с пакетом все в порядке, пропускает пакет на компьютер-получатель. Прокси – это второе поколение межсетевых экранов.

Прокси стоят между доверенной и недоверенной сетями, они устанавливают соединения от лица источника. Если пользователь в сети Интернет посылает запрос на отправку данных компьютеру во внутренней защищенной сети, прокси первым получает этот запрос и анализирует его на наличие подозрительной информации. При этом запрос не отправляется автоматически компьютеру-получателю, вместо этого прокси сам принимает запрос от лица компьютера, который он защищает. Если прокси решает, что пакет безопасен, он отправляет его компьютеру-получателю. Когда компьютер-получатель отвечает, его ответ идет обратно к прокси, который переупаковывает пакет, чтобы включить в него в качестве адреса отправителя собственный адрес, а не адрес компьютера во внутренней сети. Очень важно, что прокси разрывает коммуникационный канал – прямое соединение между внешними и внутренними компьютерами отсутствует.

Межсетевой экран такого типа создает копию каждого принятого пакета и передает дальше именно копию. Создавая копию, он переупаковывает пакет, чтобы скрыть его настоящего отправителя. Если атакующий просканирует сеть компании, он получит только информацию, которая была перехвачена и переупакована прокси. Возвращаемые пакеты будут иметь только IP-адрес прокси и очень мало информации, нужной атакующему. Таким образом, с помощью такого межсетевого экрана внутренняя сеть защищается и скрывается.

Прокси – это единственная машина, которая взаимодействует с внешним миром, обеспечивая отсутствие прямого доступа к внутренним компьютерам. Соответственно, прокси является единственным узлом, которому нужен внешний IP-адрес. Остальные компьютеры во внутренней сети могут использовать внутренние адреса (немаршрутизируемые в сеть Интернет), поскольку все равно нет внешних компьютеров, которые могут увидеть их адреса.

Многие межсетевые экраны являются системами, *подключенными к нескольким сетям* (multihomed), т.е. они имеют более одной сетевой карты. Это позволяет компании создать несколько независимых DMZ. Один интерфейс подключается к недоверенной сети (обычно это Интернет), другой интерфейс – к доверенной (внутренней сети компании), а другие интерфейсы могут использоваться для создания сегментов с различными DMZ. В одной из таких DMZ могут быть размещены веб-серверы компании, в других – сервер внешней электронной почты, DNS-сервер и т. д.

Существует два типа прокси: прикладного уровня (application-level) и сетевого уровня (circuit-level). Они описаны далее.

«За» и «против» прокси межсетевых экранов

«За»:

- Анализирует информацию в пакете на всех уровнях вплоть до прикладного уровня
- Обеспечивает лучшую безопасность, чем фильтрация пакетов
- Разрывает соединение между доверенными и недоверенными системами

«Против»:

- Некоторые прокси поддерживают только ограниченное число приложений
- Снижается скорость передачи трафика
- Программные прокси могут иметь проблемы с масштабируемостью и производительностью
- Разрывает клиент-серверную модель, что хорошо для безопасности, но плохо для функциональности

Прокси прикладного и сетевого уровней. Прокси можно описать в виде посредника между недоверенными внешними узлами и доверенными внутренними узлами. Однако возникает ряд вопросов, когда мы рассматриваем два различных типа прокси. **Прокси прикладного уровня** (application-level проху) анализируют пакет вплоть до прикладного уровня и принимают решение о доступе на основе всего содержимого пакета. Они понимают различные службы и протоколы, а также используемые ими команды. Прокси прикладного уровня может, например, различать команды FTP GET и FTP PUT и принимать решение о доступе, учитывая такой уровень детализации. При этом межсетевые экраны с фильтрацией пакетов могут разрешить или запретить команды FTP только целиком, а не на уровне конкретных команд протокола FTP.

Прокси прикладного уровня работает только для одной службы или протокола. А поскольку компьютер может иметь множество различных типов служб и протоколов (FTP, NTP, SMTP, Telnet и т.д.), требуется по одному прокси на каждый из них. Это не означает, что требуется по одному прокси на каждую службу, скорее это будет один выделенный модуль межсетевого экрана, понимающий работу определенного протокола и знающий, как правильно осуществлять в нем фильтрацию подозрительных данных.

Прокси сетевого уровня (circuit-level проху) создает соединение между клиентским компьютером и сервером, обеспечивая защиту на сеансовом уровне. Он не понимает и не заботится о более высокоуровневых вопросах, которые учитывает прокси прикладного уровня. Он знает только адреса отправителя и получателя, и принимает решение о доступе на основании этой информации.

Реализация функций прокси прикладного уровня может быть очень сложной задачей. Такой прокси должен полностью понимать работу определенных протоколов, знать, какие команды в этих протоколах являются легитимными. Он должен контролировать множество аспектов в процессе передачи данных. В качестве аналогии рассмотрим пункт проведения детального досмотра в аэропорту, для работы которого необходимо задействовать множество сотрудников, которые будут проводить беседы с пассажирами, перед тем, как разрешить им

пройти в аэропорт и сесть в самолет. Такие сотрудники обучены задавать определенные вопросы и выявлять подозрительные ответы и действия людей, они имеют набор специальных навыков и уполномочены задерживать пассажиров, показавшихся им подозрительными. Теперь представьте, что аэропорт этот – международный, и каждый из этих сотрудников говорит на одном из иностранных языков (ведь пассажиры могут быть из различных уголков мира). Сотрудник, который говорит на немецком языке, не может понять и выявить подозрительные ответы человека из Италии, поскольку он его не понимает. То же самое относится и к прокси прикладного уровня. Каждый такой прокси – это часть программного обеспечения, спроектированная для понимания определенного протокола и выявления подозрительных данных при их передаче посредством этого протокола.

ПРИМЕЧАНИЕ. Любой тип прокси разбирает получаемые пакеты на отдельные части и анализирует каждую из таких частей на предмет подозрительной активности.

Если прокси прикладного уровня не понимает определенный протокол или сервис, он не может защитить коммуникации, выполняющиеся посредством этого протокола или сервиса. С этой точки зрения, прокси сетевого уровня обычно более полезен, поскольку он не учитывает настолько сложные вопросы. Преимущество прокси сетевого уровня заключается в том, что он может работать с более широким спектром протоколов и служб, чем прокси прикладного уровня, однако его огромным недостатком является невозможность выполнения столь детального контроля, который обеспечивает прокси прикладного уровня. Жизнь полна компромиссов.

Работа прокси сетевого уровня похожа на фильтрацию пакетов, при которой решения о доступе принимаются на основе адреса, порта и типа протокола. Он анализирует данные, содержащиеся в заголовке пакета, но не данные прикладного уровня. Он не знает, является ли содержимое пакета безопасным или нет.

С другой стороны прокси прикладного уровня зависит от конкретных протоколов и служб. Не менее одного прокси необходимо использовать для каждого протокола, т.к. один прокси не может правильно интерпретировать команды всех встретившихся ему протоколов. Прокси сетевого уровня работает на более низких уровнях модели OSI и не требует использования отдельного прокси для каждого протокола, поскольку он не анализирует информацию настолько детально.

SOCKS является примером прокси-шлюза сетевого уровня, который предоставляет безопасный канал между двумя компьютерами. Когда SOCKS-клиент посылает запрос на доступ к компьютеру в Интернете, этот запрос в действительности направляется на SOCKS-прокси сети, как показано на рисунке 5-35, который анализирует пакеты на предмет наличия вредоносной информации и проверяет соблюдение правил политики, чтобы решить, следует ли разрешать это соединение. Если пакет не содержит вредоносной информации и соответствующий тип соединений разрешен, SOCKS-прокси отправляет сообщение компьютеру-получателю в Интернете. Когда компьютер в Интернете отвечает, он также посылает свои пакеты SOCKS-прокси, который снова анализирует данные и пропускает пакеты на клиентский компьютер.

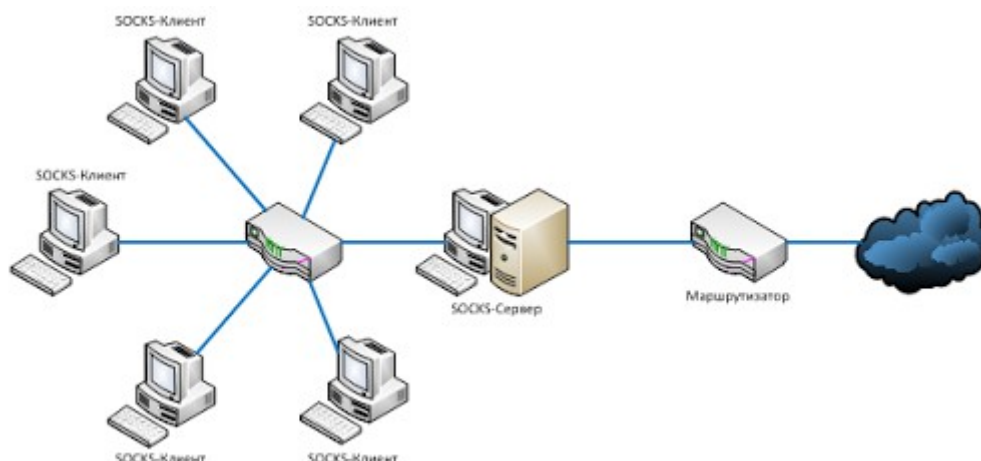


Рисунок 5-35. SOCKS-сервер обычно размещается за маршрутизатором, на каждом SOCKS-клиенте должно быть установлено программное обеспечение SOCKS

SOCKS-прокси может экранировать, фильтровать, выполнять аудит, журналировать и контролировать проходящие данные, входящие или исходящие из защищаемой сети. SOCKS-прокси достаточно популярны, поэтому многие приложения и протоколы уже настроены на работу с SOCKS, что снижает количество настроек в административных частях приложений, а различные межсетевые экраны имеют встроенное программное обеспечение SOCKS, для обеспечения защиты сетевого уровня.

Сравнение характеристик прокси межсетевых экранов прикладного и сетевого уровней.

Характеристики прокси прикладного уровня:

- Требуется наличие отдельного прокси для каждой разрешенной службы
- Обеспечивает более сложный контроль, чем прокси сетевого уровня
- Требуется большего объема обработки для каждого пакета, поэтому медленнее, чем прокси сетевого уровня

Характеристики прокси сетевого уровня:

- Не требуется отдельный прокси для каждой службы
- Не обеспечивает детального управления доступом, в отличие от прокси прикладного уровня
- Обеспечивает безопасность для широкого диапазона протоколов

Поскольку SOCKS – это прокси сетевого уровня, он не обеспечивает детального контроля, ориентированного на конкретные протоколы. Среди продуктов SOCKS серверное программное обеспечение SOCKS, которое может работать на серверах Unix, клиентские библиотеки SOCKS, поддерживающие SOCKS версии отдельных приложений и протоколов, SOCKS-оболочки для утилит, таких как traceroute и ping.

Преимущества использования прокси прикладного уровня заключаются в следующем:

- Имеются большие возможности для журналирования, что связано со способностью межсетевого экрана анализировать весь сетевой пакет, а не только сетевые адреса и номера портов.
- Аутентификация пользователей производится в соответствии с имеющейся корпоративной инфраструктурой. Прокси-шлюзы прикладного уровня способны напрямую аутентифицировать пользователей, в отличие от межсетевых экранов с фильтрацией пакетов и межсетевых экранов с контролем состояния и фильтрацией пакетов, которые могут выполнять аутентификацию только систем.
- Прокси-шлюзы прикладного уровня – это не простые устройства третьего уровня. Они могут контролировать атаки спуфинга и другие сложные атаки.

Характеристики SOCKS-прокси.Ниже приведены некоторые важные характеристики SOCKS-прокси:

- Это прокси сетевого уровня
- Он требует использования поддерживающих SOCKS клиентов, на которых установлено программное обеспечение SOCKS
- Он может быть ресурсоемким
- Он предоставляет функции аутентификации и шифрования аналогично другим VPN-протоколам, но он не является традиционным VPN-протоколом

Некоторые из недостатков использования прокси прикладного уровня приведены ниже:

- Обычно не подходят при использовании широких полос пропускания или приложений реального времени.
- Имеют ограничения в части поддержки новых сетевых приложений и протоколов.
- Большинство производителей прокси-шлюзов прикладного уровня предоставляют универсальные прокси-агенты для поддержки неопределенных сетевых протоколов или приложений.
- Использование универсальных агентов негативно влияет на преимущества архитектуры прокси-шлюзов прикладного уровня, поскольку они просто позволяют «туннелировать» трафик через межсетевой экран.

Динамическая фильтрация пакетов

Когда системе во внутренней сети нужно установить взаимодействие с системой вне доверенной сети, она должна выбрать исходный порт, чтобы получающая система знала, как правильно ей ответить. Получающей системе нужен IP-адрес и номер порта, чтобы ее ответ смог найти дорогу к компьютеру отправителя. Порты до 1023 называются *стандартными (общеизвестными) портами* (well-known ports) и зарезервированы для серверных служб. Для установления соединения с другой системой, отправитель должен выбрать динамический порт с номером выше 1023. Затем межсетевой экран с динамической фильтрацией пакетов создает список контроля доступа (ACL), который позволяет внешней системе взаимодействовать с внутренней системой через выбранный порт. Если ваш межсетевой экран с фильтрацией пакетов не поддерживает такую возможность, это может стать уязвимостью при использовании портов с номерами выше 1023, т.к. клиентская сторона выбирает эти порты динамически, и межсетевой экран не будет знать точно на каком порту разрешенный, а на каком запрещенный трафик.

ПРИМЕЧАНИЕ. Стандартный порт для HTTP – 80-й. Это означает, что на сервере есть служба, которая прослушивает 80-й порт на предмет HTTP-трафика. HTTP (как и большинство других протоколов) работают с использованием клиент-серверной модели. На серверной стороне используются стандартные порты (например, FTP использует порты 20 и 21; SMTP – 25), поэтому все знают, как подключиться к нужным службам. На клиентской стороне не должны использоваться стандартные порты для собственных нужд, вместо них должны выбираться случайные порты с номерами, выше 1023.

Внутренняя система может выбрать исходный порт, например, 11111 для передачи своих сообщений внешней системе. Кадр отправляется на межсетевой экран с фильтрацией пакетов, который создает ACL, как показано на Рисунке 5-36, содержащий указание на то, что должен быть разрешен ответ от компьютера-получателя на IP-адрес и порт 11111 внутренней системы. После отправки ответа системой получателя, межсетевой экран примет его и передаст дальше – внутренней системе. Такие ACL являются динамическими по своей природе, поэтому после завершения соединения (получения пакетов FIN или RST) ACL удаляется из списка. В протоколах без установления соединения, таких как UDP, ACL удаляется по окончании времени таймута.

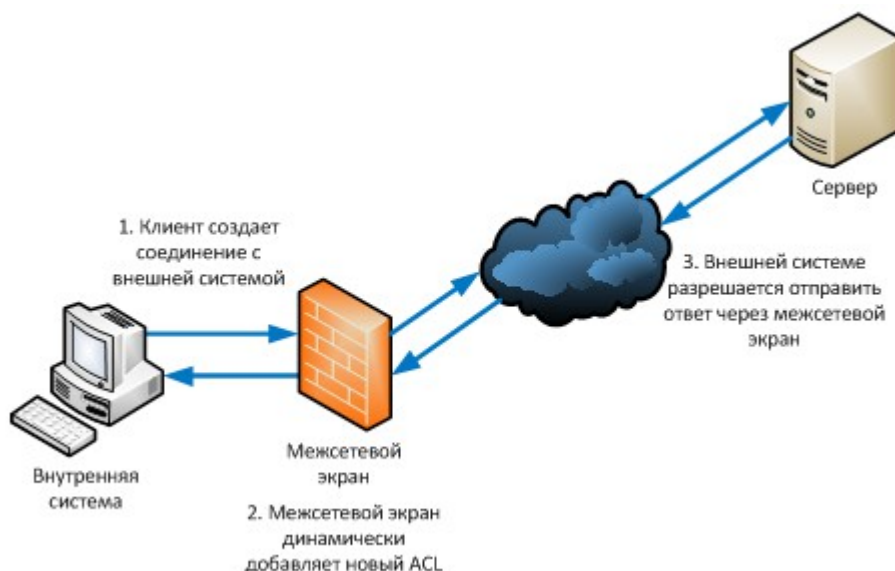


Рисунок 5-36. При динамической фильтрации пакетов ACL добавляется при создании соединения

Преимущество межсетевых экранов с динамической фильтрацией пакетов, являющихся межсетевыми экранами четвертого поколения, заключается в том, что они дают возможность разрешить любой тип исходящего трафика, а для входящего трафика разрешить только тот трафик, который является ответом на направленный ранее запрос.

Прокси уровня ядра

Прокси уровня ядра (kernel proxy firewall) называют межсетевыми экранами пятого поколения. Они отличаются от всех описанных ранее технологий межсетевых экранов, т.к. они создают динамические, соответствующие требованиям заказчика стеки TCP/IP, когда необходимо оценивать пакеты.

Когда пакет поступает на прокси уровня ядра, создается новый виртуальный сетевой стек, который состоит из прокси только тех протоколов, которые необходимы для надлежащей проверки этого конкретного пакета. Например, если это пакет FTP, то только FTP-прокси загружается в стек. Пакет тщательно исследуется на каждом уровне стека – анализируются данные заголовков канального, сетевого, транспортного, сеансового и прикладного уровней. Если на любом из этих уровней межсетевому экрану что-то покажется небезопасным, пакет уничтожается.

Прокси уровня ядра быстрее прокси прикладного уровня, поскольку все проверки и обработка происходят в ядре, а не на высоком уровне программного обеспечения операционной системы. Они остаются системами, основанными на прокси, разрывая соединения между внутренними и внешними системами. Они играют роль посредника и могут выполнять функции NAT, изменяя исходный адрес, как это делают раммотранные ранее прокси межсетевые экраны.

В таблице 5-8 перечислены важные понятия и характеристики рассмотренных ранее типов межсетевых экранов. Хотя различные реальные межсетевые экраны могут предоставлять смесь этих сервисов и работать на различных уровнях модели OSI, важно понимать основные определения и функции этих типов межсетевых экранов.

Тип межсетевого экрана	Уровень OSI	Характеристики
Пакетная фильтрация	Сетевой	Анализирует адреса отправителя и получателя, номера портов и запрашиваемые службы. Маршрутизаторы используют ACL, определяющие допустимые варианты доступа к сети.
Прокси прикладного уровня	Прикладной	Глубоко анализирует пакет и принимает детализированные решения о доступе. Требуется по одному прокси на каждый сервис.
Прокси сетевого уровня	Сетевой	Анализирует только информацию заголовка пакета. Обеспечивает защиту для более широкого спектра протоколов и служб, чем прокси прикладного уровня, но не обеспечивает настолько же детального уровня контроля.
С контролем состояния	Сетевой, транспортный	Учитывает состояние и контекст пакетов. Хранит в таблице состояний информацию о каждом соединении.
Прокси уровня ядра	Прикладной	Быстрее, поскольку обработка осуществляется в ядре. Для каждого пакета создается один сетевой стек.

Таблица 5-8. Различия между сетевыми межсетевыми экранами

Ниже приведены некоторые лучшие практики, применимые к любому типу межсетевых экранов:

- Блокируйте перенаправление ICMP-трафика
- ACL следует делать проще и прямолинейнее
- Запрещайте маршрутизацию от источника
- Закрывайте ненужные порты с потенциально опасными службами
- Блокируйте неиспользуемые интерфейсы
- Блокируйте направленные широковещательные IP-рассылки
- Блокируйте входящие пакеты с внутренними адресами (это может быть признаком их подделки (спуфинга))
- Блокируйте трафик многоадресных рассылок, если они не требуются
- Включите журналирование

Устройства. Межсетевой экран может быть реализован в виде программного обеспечения, установленного на обычный компьютер и использующего обычную операционную систему, а может быть выделенным аппаратным устройством (appliance), имеющим собственную операционную систему. Второй вариант обычно более безопасен, т.к. производитель использует урезанную версию операционной системы (обычно Linux или BSD Unix). Полнофункциональная операционная система не требуется для межсетевого экрана. Повышенная сложность полнофункциональной операционной системы открывает двери для уязвимостей. Если хакер сможет воспользоваться уязвимостью и отключить межсетевой экран компании, компания будет подвержена большой опасности.

Анализ данных. Данные – это «новое золото» для большинства компаний, которое сегодня необходимо защищать для обеспечения безопасности, соблюдения требований законодательства, регуляторов, получения конкурентных преимуществ. В связи с этим были реализованы дополнительные функции и специальные продукты, которые выполняют анализ данных. Этот анализ производится на различных уровнях и в различных типах протоколов. Некоторые продукты могут анализировать только содержимое пакетов протокола SMTP, тогда как другие могут анализировать содержимое пакетов SMTP, FTP, HTTP и многих других. Содержимое пакетов контролируется на предмет подозрительных действий (вирусы, мобильный код, ActiveX и т.д.), либо на предмет наличия критичных данных, которые не должны покидать сеть компании (коммерческая тайна, конфиденциальные данные, номера банковских карт и т.п.).

Архитектура межсетевых экранов

Межсетевые экраны могут быть размещены во множестве различных мест в сети для реализации различных потребностей. Они могут защищать внутреннюю сеть от воздействий из внешней сети, являясь узким местом для всего трафика. Межсетевой экран может использоваться для сегментирования сети, реализации управления доступом между двумя и

более подсетями. Межсетевой экран может также использоваться для создания DMZ между внутренней сетью и внешней сетью.

Межсетевой экран следует устанавливать на узлах-бастионах, на которых заблокированы все сервисы, кроме необходимых для выполнения ими своих функций.

Узел-бастион. *Узел-бастион* (bastion host) – это просто другое название укрепленных (hardened) систем. Узел-бастион обычно открыт для внешних атак, поскольку он устанавливается на передовой линии сетевой безопасности, именно он является системой, которая видна из Интернета. Соответственно, это узел-бастион должен быть максимально защищен – на нем не должны быть запущены ненужные сервисы, неиспользуемые подсистемы должны быть заблокированы, для уязвимостей должны быть установлены соответствующие патчи, ненужные учетные записи должны быть заблокированы, а все ненужные для выполнения его функций порты должны быть закрыты. Узел-бастион не связан с программным обеспечением межсетевого экрана и его работой. Это просто система, которая надежно защищена. Любая размещенная в DMZ система должна быть установлена на узле-бастионе, поскольку она находится в непосредственной близости от Интернета и злоумышленников, которые могут нанести ей вред. Если программное обеспечение межсетевого экрана установлено не на защищенную операционную систему или узел-бастион, сам такой межсетевой экран является уязвимым.

Межсетевой экран с двойной привязкой. *Двойная привязка* (dual-homed) означает, что устройство имеет два интерфейса: один смотрит во внешнюю сеть, а другой – во внутреннюю. Если программное обеспечение межсетевого экрана установлено на устройство с двойной привязкой (обычно так и бывает), в его операционной системе должна быть отключена пересылка и маршрутизация сетевых пакетов, исходя из соображений безопасности. Если эти функции не отключены, к компьютеру не будут применены необходимые правила ACL и другие ограничения, применение которых требуется от межсетевого экрана. Когда пакет из недоверенной сети приходит на подключенную к ней (внешнюю) сетевую карту межсетевого экрана с двойной привязкой, в настройках операционной системы которого не отключена пересылка трафика, его операционная система просто перешлет пакет на сетевую карту, подключенную к внутренней доверенной сети, без проведения анализа пакета программным обеспечением межсетевого экрана.

Многие сетевые устройства сегодня имеют *множественную привязку* (multihomed), т.е. они имеют несколько сетевых карт, использующихся для подключения различных сетей. Именно на таких устройствах обычно устанавливается программное обеспечение межсетевого экрана, при необходимости обеспечения безопасности и управления передачей трафика между различными сетями. Использование обычной архитектуры межсетевого экрана с множественной привязкой позволяет компании иметь несколько DMZ. В одной DMZ могут быть размещены устройства, совместно используемые компаниями в экстрасети, в другой DMZ могут быть размещены серверы DNS и почтовые серверы компании, в третьей – веб-серверы компании и т.д. Отдельные DMZ используются по двум причинам: для управления различными видами трафика (например, чтобы обеспечить, что HTTP-трафик идет только на веб-сервера, а DNS-запросы – только к DNS-серверам), а также для обеспечения гарантии того, что если одна система в одной DMZ скомпрометирована атакующим, то системы в других DMZ ему по-прежнему недоступны.

Экранированный узел. *Экранированный узел* (screened host) – это межсетевой экран, который взаимодействует напрямую с маршрутизатором периметра и внутренней сетью. На Рисунке 5-37 показан такой тип архитектуры.

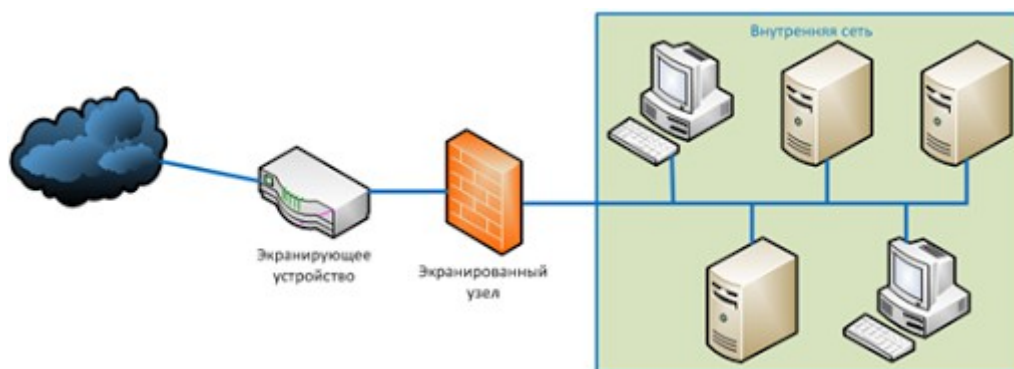


Рисунок 5-37. Экранированный узел – это межсетевой экран, экранированный маршрутизатором

Полученный из Интернет трафик сначала проходит фильтрацию пакетов на внешнем маршрутизаторе. Прошедший через маршрутизатор трафик отправляется на экранированный узел – межсетевой экран, который применяет дополнительные правила к трафику и уничтожает запрещенные пакеты. Затем трафик передается на узел получателя во внутренней сети. Экранированный узел (межсетевой экран) – это устройство, которое принимает трафик напрямую от маршрутизатора. Никакой внешний трафик не может пройти напрямую от маршрутизатора во внутреннюю сеть, минуя межсетевой экран.

Если межсетевой экран является прикладной системой, защита на сетевом уровне обеспечивается маршрутизатором, а на прикладном уровне – прокси. Такое распределение обеспечивает высокую степень безопасности, т.к. атакующему для достижения успеха потребуется скомпрометировать две системы.

Что означает в данном контексте слово «экранирование» (screening)? Как показано на Рисунке 5-37, маршрутизатор – это экранирующее устройство, а межсетевой экран – экранированный узел. Это просто означает, что существует уровень, который анализирует трафик и удаляет из него кучу «мусора» до направления этого трафика на межсетевой экран. Экранированный узел отличается от экранированной подсети, которая описана далее.

Экранированная подсеть. Архитектура *экранированной подсети* (screened-subnet) добавляет еще один уровень безопасности в архитектуру экранированного узла. Внешний межсетевой экран экранирует данные, входящие в сеть DMZ. Однако вместо перенаправления трафика непосредственно во внутреннюю сеть, трафик передается на внутренний межсетевой экран, который также выполняет фильтрацию трафика. Использование двух этих физических межсетевых экранов создает DMZ.

Если используется среда с экранированным узлом, то атакующему, успешно проникшему через межсетевой экран, ничего не мешает получить полный доступ во внутреннюю сеть. В среде с экранированной подсетью атакующий должен пройти через еще один маршрутизатор (или межсетевой экран) для получения доступа во внутреннюю сеть. В таком многоуровневом подходе к безопасности, реализуется больше уровней для обеспечения лучшей защиты. На Рисунке 5-38 показан простой пример экранированной подсети.



Рисунок 5-38. В экранированной подсети два межсетевых экрана используются для создания DMZ

На Рисунке 5-38 показан простейший пример. В реальности обычно используются значительно более сложные сети и DMZ. На Рисунках 5-39 и 5-40 показаны некоторые другие возможные архитектуры экранированных подсетей и их конфигураций.

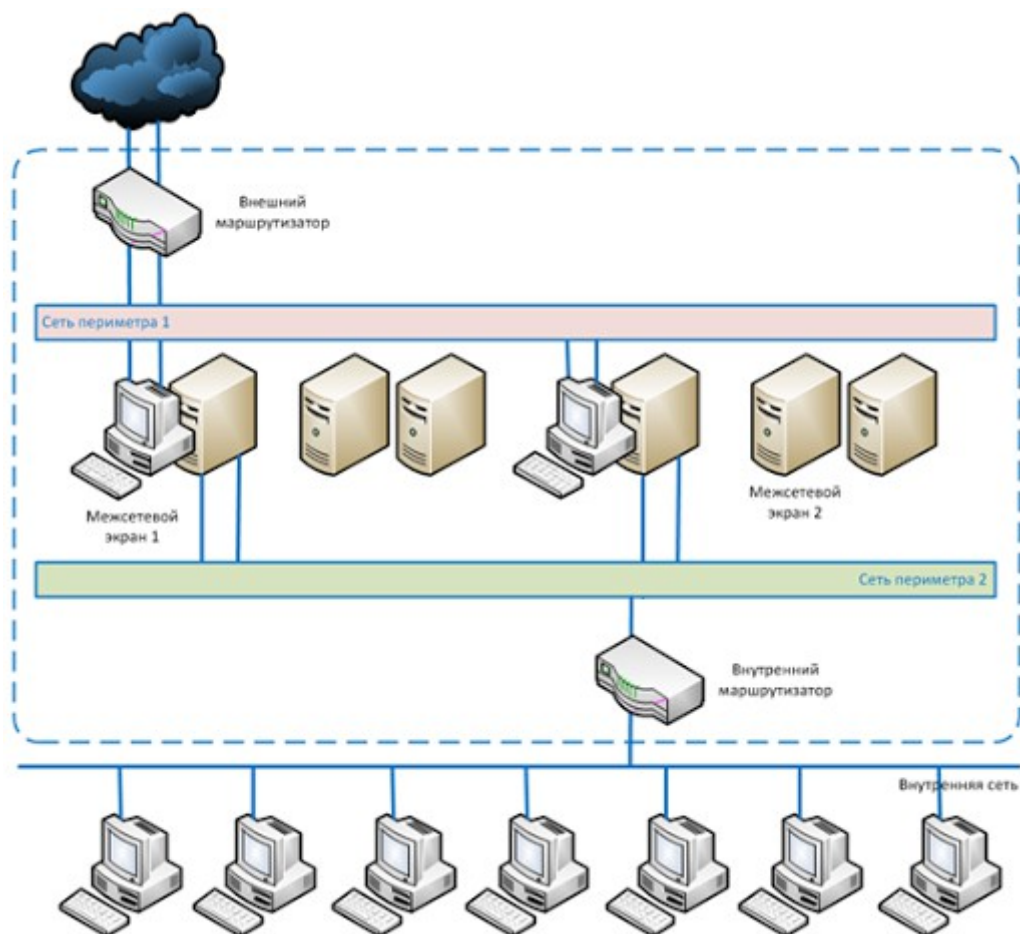


Рисунок 5-39. Экранированная подсеть может включать в себя несколько различных сетей и нескольких различных межсетевых экранов, каждый из которых фильтрует трафик по своим правилам, закрывая определенные уязвимости

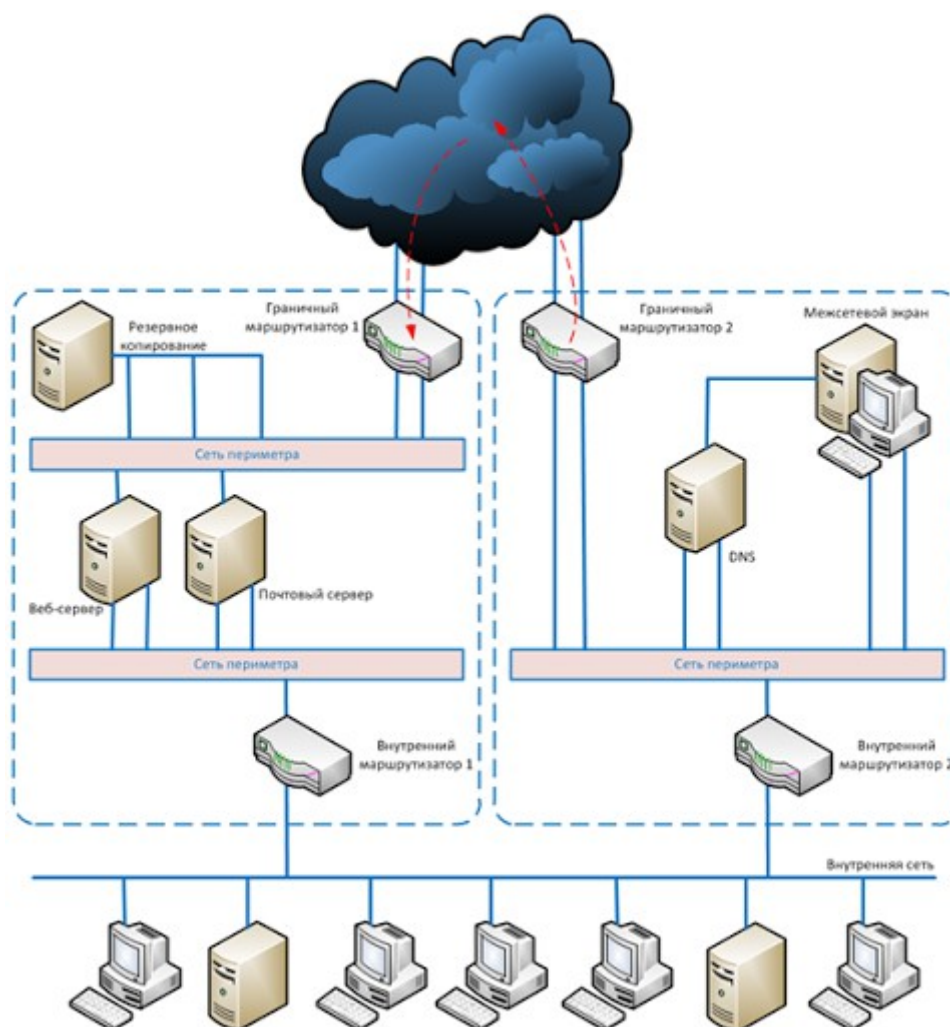


Рисунок 5-40. Некоторые архитектуры имеют разделенные экранированные подсети с различными типами серверов в каждой из них

Вариант с экранированной подсетью обеспечивает лучшую защиту, чем отдельный межсетевой экран или межсетевой экран на экранированном узле, поскольку в этом варианте три устройства работают совместно, и все они должны быть скомпрометированы атакующим перед тем, как он получит доступ в сеть. Такая архитектура также создает DMZ между двумя маршрутизаторами, которая работает как небольшая изолированная сеть между доверенной внутренней сетью и недоверенной внешней сетью. Обычно внутренние пользователи имеют ограниченный доступ к серверам в этой области. Веб-серверы, почтовые серверы и другие публичные серверы часто размещаются в DMZ. Такое решение обеспечивает наивысшую безопасность, однако его настройка и сопровождение могут быть очень сложными и трудоемкими, например, когда нужно добавить новые сервисы, должны быть перенастроены все три системы, а не одна.

Организация и настройка DMZ, сети периметра, экранированных узлов и подсетей определяются политикой безопасности компании. В политике следует ясно описывать требуемый уровень безопасности, указывать сервисы, которые должны быть доступны внутренним и внешним пользователям.

ПРИМЕЧАНИЕ. Иногда архитектуру с экранированным узлом называют однозвенной (single-tiered) конфигурацией, а с экранированной подсетью – двухзвенной (two-tiered) конфигурацией. Если с помощью трех межсетевых экранов создаются две отдельных DMZ, это называют трехзвенной (three-tiered) конфигурацией.

«Обязанности» межсетевых экранов

«По умолчанию» любой межсетевой экран должен блокировать любые пакеты, кроме

разрешенных явно. Соответственно, если в правилах межсетевого экрана не указано, какие пакеты являются разрешенными, он должен блокировать все принимаемые пакеты. Любой входящий в сеть пакет, в котором в качестве адреса отправителя указан адрес узла внутренней сети, должен блокироваться. **Маскарадинг** (masquerading), или **спуфинг** (spoofing), является популярной атакой, при которой атакующий изменяет заголовок пакета, указывая в нем в качестве адреса отправителя адрес внутреннего узла, который он хочет атаковать. Такой пакет в любом случае следует считать поддельным и нелегитимным, поскольку нет причин, по которым пришедший из сети Интернет пакет имел бы в качестве адреса отправителя адрес, принадлежащий внутренней сети. Поэтому межсетевой экран должен блокировать такой пакет. То же самое относится и к исходящему трафику. Не следует разрешать исходящий из сети трафик, в пакетах которого в адресе отправителя указан адрес, не принадлежащий внутренней сети. Если межсетевой экран выявляет такие пакеты, это означает, что кто-то или что-то во внутренней сети занимается подделкой сетевых пакетов (спуфингом). С большой вероятностью, это может быть следствием работы зомби-агентов, использующихся для проведения DoS- (DDoS-) атак.

Если безопасность имеет наивысший приоритет для компании, ее межсетевые экраны должны производить пересборку фрагментированных пакетов до их отправки получателю. В некоторых типах атак, хакеры изменяют пакеты и делают их похожими на какие-либо другие пакеты, которыми они не являются. Когда фрагментированный пакет приходит на межсетевой экран, он видит только часть картины и должен сделать правильное предположение о том, является ли это частью опасного пакета или нет. Поскольку этот фрагмент включает только часть полного пакета, межсетевой экран принимает решение, не имея в своем распоряжении всех фактов. А когда все фрагменты пакета будут доставлены на компьютер получателя, в результате их сборки может быть получен вредоносный пакет, который нанесет большой ущерб. В среде, которая требует высокого уровня безопасности, межсетевой экран должен принимать каждый фрагмент, собирать из них полный пакет, и только после этого принимать решение на основе информации всего пакета. Однако при этом возникает существенный недостаток, вызванный тем, что межсетевой экран, пересобирающий фрагменты пакета перед принятием решения об отправке его на компьютер получателя, становится причиной дополнительных накладных расходов и задержек в передаче трафика. Задачей офицера безопасности и компании является принятие решения о целесообразности применения такой конфигурации и допустимости дополнительных задержек при передаче трафика.

Многие компании устанавливают запрет на прием пакетов, содержащих информацию маршрутизации от источника (source routing), о чем мы уже говорили ранее. Наличие информации *маршрутизации от источника* означает, что пакет решает, как его передать получателю, самостоятельно, без помощи маршрутизаторов между компьютерами отправителя и получателя. Маршрутизация от источника обеспечивает перемещение пакета через сеть по заранее определенному пути. При этом компьютер отправителя должен знать топологию сети и правильный маршрут для пакета. Это упрощает задачу маршрутизаторов и других устройств между отправителем и получателем, поскольку им не нужно думать, как маршрутизировать этот пакет. Однако это ведет к увеличению рисков нарушения безопасности. Когда маршрутизатор получает пакет с информацией маршрутизации от источника, он понимает, что пакет сам знает, что с ним нужно делать, и пропускает его дальше. При этом не все фильтры могут применяться к таким пакетам, а для сетевых администраторов предпочтительнее, чтобы пакеты маршрутизировались по установленным ими маршрутам, а не по маршрутам, диктуемым самими пакетами. Чтобы исключить неправильную маршрутизацию, многие межсетевые экраны настраивают на проверку наличия в пакетах информации маршрутизации от источника и отклонение пакетов, в которых она есть.

К сожалению, установка компанией межсетевого экрана может создать ложное чувство

безопасности. Межсетевые экраны – это только часть головоломки, а для обеспечения безопасности нужно множество частей.

Характеристики архитектуры межсетевого экрана. Важно понимать следующие характеристики различных типов архитектуры межсетевых экранов:

С двойной привязкой:

- Один компьютер с отдельными сетевыми картами, подключенными к каждой сети.
- Используется для отделения внутренней доверенной сети от внешней недоверенной сети.
- На компьютере должны быть отключены функции пересылки и маршрутизации пакетов, чтобы две сети были реально разделены.

Экранированный узел:

- Маршрутизатор фильтрует (экранирует) трафик перед его отправкой на межсетевой экран.

Экранированная подсеть:

- Внешний маршрутизатор фильтрует (экранирует) трафик перед его отправкой в подсеть. Трафик, направленный во внутреннюю сеть, проходит через два межсетевых экрана.

В следующем списке приведены некоторые недостатки межсетевых экранов:

- В большинстве случаев необходимо использовать распределенный подход для контроля всех сетевых точек доступа, что нельзя реализовать при помощи одного только межсетевого экрана.
- Межсетевой экран является потенциальным «бутылочным горлышком» для потока трафика.
- Межсетевой экран может заблокировать нужные сервисы, которые могут быть необходимы пользователям (хотя, конечно, это недостаток для пользователей, а для специалистов по безопасности – преимущество).
- Большинство межсетевых экранов не обеспечивает защиту от вирусов, которые загружаются пользователями или приходят в сообщениях электронной почты. Межсетевые экраны не имеют механизмов выявления вирусов.
- Граничные межсетевые экраны обеспечивают низкий уровень защиты от внутренних атакующих.
- Межсетевые экраны не обеспечивают защиты от несанкционированно установленных модемов, работающих в режиме ожидания входящих вызовов.
- Межсетевые экраны не защищают от несанкционированно установленных беспроводных точек доступа.

Роль межсетевого экрана становится все сложнее и сложнее, она получает больше функций и обязанностей. Эти сложности работают против сетевого администратора и специалиста по безопасности, т.к. от них требуется правильное понимание и надлежащее внедрение дополнительной функциональности межсетевых экранов. Без понимания различных типов имеющихся на рынке межсетевых экранов, а также архитектур, количество «дыр» в безопасности увеличивается, открывая дорогу злоумышленникам.

6.8. Хост-приманка

Хост-приманка (honeypot) – это компьютер, который размещается обычно в экранированной подсети или DMZ и пытается привлечь внимание атакующих к себе и, соответственно, отвлечь их от реально использующихся в работе компьютеров. Чтобы сделать хост-приманку привлекательным для атакующих, администраторы могут разрешить на них наиболее часто атакуемые службы и порты. Однако администратор должен позаботиться, чтобы эта система была полностью изолированной, чтобы атакующий,

скомпрометировав ее, не мог получить доступ к другим компьютерам в сети. На некоторых хостах-приманках администраторы создают эмуляцию отдельных служб, т.е. реальная служба на них не запущена, но работа программного обеспечения организована так, что для атакующего это выглядит как работающая и доступная служба. В Домене 08 мы еще раз коснемся хостов-приманок в контексте понимания разницы между *заманиванием* (enticement) и *провокацией* (entrapment). Легальные хосты-приманки могут привлечь внимание атакующих, чтобы они попытались взломать именно этот компьютер, но они не провоцируют атакующего на это. Приведенный в Домене 08 пример говорит о том, что баннер, размещенный на веб-сервере – приманке, который говорит, что на нем можно скачать бесплатно музыку в формате MP3, – это провокация. На легальном хосте-приманке просто запущены уязвимые службы, открыты порты, доступны для просмотра банеры служб – этого достаточно, чтобы злоумышленник попытался атаковать такую систему в первую очередь, но она никого не провоцирует на незаконные действия.

Сетевые администраторы не хотят допустить атакующих до своих систем, для этого некоторые из них устанавливают хосты-приманки, в качестве «подсадных уток». Другие администраторы хотят сами напасть на тех, кто атаковал их. Они включают на хостах-приманках детальное журналирование событий и проводят детальные расследования любых попыток несанкционированного доступа для судебного преследования атакующих.

6.9. Разделение и изоляция сетей

Очень важно отделять сети и подсети друг от друга. Обычно это означает необходимость установки маршрутизаторов, которые не пропускают информацию широковещательных и коллизионных доменов, используют различные диапазоны адресов для различных сегментов. Большинство сетей в наше время используют технологию Ethernet, постоянно выполняющую широковещательную рассылку информации, которая может быть полезна внешним и внутренним атакующим. Поскольку данные свободно распространяются в такой среде, вам следует убедиться, что части сети, в которых хранится критичная информация, надлежащим образом отделены в отдельные сегменты от других частей сети.

Архитектура сети должна быть хорошо продумана от начала до конца, полностью документирована и тщательно протестирована. К примеру, сетевой администратор может решить ограничить прямой доступ к мейнфрейму, на котором хранится критичная информация. Соответствующий трафик следует направить через один канал и фильтровать его маршрутизатором или межсетевым экраном. Также, администратор может решить, что не все пользователи сети должны иметь доступ к административной подсети, в которой размещены консоли управления для всех маршрутизаторов, систем IDS, серверов, на которых хранятся журналы регистрации событий. Компьютеры из административной подсети должны иметь возможность взаимодействия с остальными частями сети, но нет никаких оснований, чтобы обычные пользователи имели возможность свободного доступа в административную подсеть. Доступ в эту подсеть следует изолировать с помощью списков контроля доступа, реализуемых с помощью окружающих маршрутизаторов (или межсетевых экранов) и правильной сегментации.

Документирование архитектуры сети является хорошей идеей, но то, что она отражена на бумаге, не делает ее правильной. Если на схеме сети указано, что с помощью межсетевого экрана исключено взаимодействие подсети А с подсетью В, это должно быть тщательно протестировано, должна быть проведена попытка атаки (тест на проникновение), чтобы понять, так ли это на самом деле.

7. Сетевые сервисы и Протоколы

В начале этого Домена мы касались протоколов, технологий, топологий и устройств, которые могут использоваться в среде LAN. Однако эти сервисы используются не только в среде LAN, они также применяются в инфраструктуре MAN и WAN. Описание LAN и WAN

приводится в отдельных разделах для обеспечения ясного понимания разницы между этими сетевыми концепциями. Как было указано ранее, сеть создается для обеспечения возможности взаимодействия компьютеров друг с другом, обеспечения централизованного администрирования и совместного использования ресурсов. Ресурсами обычно являются сетевые службы. В следующих разделах описаны наиболее популярные службы среды LAN.

7.1. Сетевая операционная система

Сетевая операционная система (NOS – Network operating system, Сетевая ОС) – это специализированное программное обеспечение, созданное для управления доступом к сетевым ресурсам и предоставления необходимых сервисов, позволяющих компьютерам взаимодействовать с окружающей сетью. Сетевая ОС отличается от однопользовательской операционной системы. Сетевая ОС работает в рамках клиент/серверной модели, в которой ресурсы, файлы и приложения централизованы и все пользователи используют их непосредственно на сервере (и не имеют собственных копий этих ресурсов на каждой рабочей станции). Сетевая ОС управляет соединениями и связями между системами и компонентами.

Также, сетевая ОС имеет встроенные механизмы аутентификации, необходимые для работы в сетевом окружении и выполнения функций аудита. Однопользовательские операционные системы не обеспечивают строгой аутентификации. Любой, кто работал с Windows 95 или 98 знает, что достаточно было нажать кнопку ОК или Отмена, когда операционная система запрашивала имя и пароль, даже если вы не вводили эту информацию. Такие операционные системы не требуют аутентификации. Однако любой, кто работал с Windows NT или Windows 2000 знает, что эти операционные системы не дадут вам доступа к рабочему столу пользователя, если вы не знаете правильное имя пользователя и/или пароль.

Однопользовательские операционные системы могут работать в среде точка-точка (разновидность рабочей группы) и обеспечивать возможность совместного использования ресурсов и файлов. Однако среда точка-точка не обеспечивает достаточного уровня безопасности, не предоставляет возможностей централизованного управления (в том числе управления доступом), что необходимо в большинстве сетей. Кроме того, однопользовательские операционные системы не имеют полноценных служб каталогов, аналогичных имеющимся в сетевых ОС. А это важная часть большинства современных сетей.

Ниже представлен небольшой список сервисов сетевых ОС, которых нет в однопользовательских системах:

- службы каталогов
- поддержка межсетевого взаимодействия, маршрутизации, WAN
- поддержка работы удаленных dial-up-пользователей
- функциональность кластеризации
- строгая аутентификация, авторизация, управление доступом и аудит
- файловые сервисы и сервисы печати, включая резервное копирование и репликацию
- инструменты управления и администрирования для удаленных клиентов
- функциональность распространения программного обеспечения, инвентаризации аппаратного и программного обеспечения
- возможности обеспечения отказоустойчивости

Если пользователю компьютера, использующему сетевую ОС, требуется доступ к ресурсу в сети, программное обеспечение сетевой ОС использует редирактор, направляющий

компьютер на запрашиваемый ресурс. Во многих случаях этот редиректор работает на более низком уровне, чем высокоуровневое приложение, которое обращается к нему. Обычно приложение даже не знает, что ресурс не хранится на локальном компьютере. Это решает множество проблем и не требует от разработчиков приложений делать дополнительную работу, разрабатывая процесс отслеживания самих сетевых ресурсов.

Ссылки по теме:

- Google listings for operating systems

7.2. Служба доменных имен

Представьте, насколько сложно было бы использовать Интернет, если бы нам нужно было помнить реальные IP-адреса сайтов. **Служба доменных имен** (DNS – Domain Name Service) – это метод преобразования имен узлов (hostname) в IP-адреса, что позволяет использовать имена вместо IP-адресов при запросе конкретных узлов в Интернете. Не так давно, когда Интернет насчитывал всего около 100 компьютеров (против миллионов компьютеров сейчас), использовался специальный список для хранения информации о связи имени каждого компьютера с его IP-адресом. Этот список хранился на FTP-сервере и все имели доступ к нему. Задача поддержки актуальности этого списка постепенно становилась все более сложной и утомительной, поэтому компьютерное сообщество решило автоматизировать этот процесс.

Была разработана иерархическая система для доменных имен и в 1992 году NSF (National Science Foundation - Национальный научный фонд) заключил контракт с NSI (Network Solutions, Inc.) на управление и поддержку доменных имен, а также процесса регистрации этих имен. NSI выполняла регистрацию имен и вела каталог преобразования имен узлов на серверах DNS. Также она поддерживала официальную базу данных Интернета, которая являлась корневой для DNS-серверов. Официальный корневой DNS-сервер содержал 13 файлов, по одному на каждый сервер домена высшего уровня.

До 1999 года IANA (Internet Assigned Numbers Authority - Администрация адресного пространства Интернет) поддерживала и координировала распределение IP-адресов. Крупные интернет-провайдеры регистрировали большие блоки IP-адресов и затем распределяли их между более мелкими интернет-провайдерами или индивидуальными пользователями. Однако после 1999 ICANN (Internet Corporation for Assigned Names and Numbers) пересмотрела обязанности по распределению блоков IP-адресов, управлению DNS, управлению системой корневого сервера. NSI продолжила поддерживать официальные корневые базы данных.

Прекрасно. Закончим экскурс в историю. Но как работают DNS и какое место они занимают в сети?

Когда пользователь вводит URL (Uniform Resource Locator - Унифицированный указатель ресурса) в адресной строке своего веб-браузера, этот URL состоит из слов или букв, которые пользователю не сложно запомнить (например, www.logicalsecurity.com). Однако эти слова понятны только пользователю, а компьютер работает с IP-адресами. Поэтому после ввода пользователем URL и нажатия Enter, компьютер в действительности обращается к DNS-серверу, чтобы преобразовать этот URL или имя узла в IP-адрес, понятный компьютеру. После того как URL или имя узла распознано и преобразовано в IP-адрес, компьютер знает как обратиться к веб-серверу, который содержит запрашиваемую веб-страницу.

Многие компании имеют собственные DNS-серверы для преобразования имен своих внутренних узлов. Также эти компании обычно используют DNS-серверы своего интернет-провайдера для преобразования имен узлов в сети Интернет. Внутренний DNS-сервер может использоваться для преобразования имен узлов во всей сети, обычно используется больше одного DNS-сервера для разделения между ними нагрузки и обеспечения

отказоустойчивости.

В DNS-серверах пространства имен DNS административно разделены на **зоны**. Например, одна зона может содержать все имена компьютеров департаментов маркетинга и бухгалтерии, другая зона может содержать имена всех компьютеров административного, исследовательского и юридического департаментов. DNS-сервер, который хранит файлы одной из таких зон, называют *ответственным* (authoritative) сервером имен для этой конкретной зоны. В зону может входить один или более доменов, при этом DNS-сервер, который хранит записи о соответствующих узлах, является ответственным сервером имен для этих доменов.

DNS-сервер содержит записи, которые связывают имена узлов с их IP-адресами. Эти записи называются записями ресурсов (resource record). Если компьютеру пользователя нужно преобразовать имя узла в IP-адрес, он смотрит свои настройки TCP/IP, чтобы найти в них адрес DNS-сервера. Затем компьютер отправляет запрос DNS-серверу, включая в него имя узла для преобразования. DNS-сервер просматривает свои записи ресурсов в поисках записи с нужным именем узла и, найдя ее, возвращает компьютеру IP-адрес этого узла.

Рекомендуется в каждой зоне размещать **первичный** и **вторичный** DNS-сервер. Первичный DNS-сервер содержит актуальные записи ресурсов для зоны, а вторичный DNS-сервер содержит копии этих записей. При этом пользователи могут использовать вторичный DNS-сервер для преобразования имен, что позволит снизить нагрузку на первичный сервер. Если первичный сервер по какой-либо причине отключается или выходит из строя, пользователи могут продолжать использовать вторичный сервер для преобразования имен. Наличие первичного и вторичного DNS-серверов обеспечивает отказоустойчивость и избыточность, гарантируя непрерывную работу пользователей, что бы ни случилось с одним из этих серверов.

Первичный и вторичный DNS-сервера синхронизируют информацию между собой посредством *передачи зоны* (zone transfer). После того, как на первичном DNS-сервере произойдут изменения, эти изменения реплицируются на вторичный сервер. Важно настроить DNS-сервер таким образом, чтобы он разрешал передачу зон только между конкретными серверами. Многие годы атакующие выполняли с помощью подставных DNS-серверов передачу зоны для получения очень важной информации о сети.

Несанкционированная передача зоны может произойти в том случае, если DNS-серверы неправильно настроены (или не настроены) для исключения такой атаки.

DNS в Интернет и домены

Сети в Интернет соединены в иерархическую структуру, там используются другие DNS-сервера, как показано на Рисунке 5-41. При выполнении маршрутизации, если маршрутизатор не знает правильный путь к получателю пакета, маршрутизатор передает этот пакет вышестоящему маршрутизатору. Вышестоящий маршрутизатор знает обо всех нижестоящих маршрутизаторах. Этот маршрутизатор имеет более широкий обзор маршрутизации в Интернете, у него хорошие шансы передать пакет по назначению. Это справедливо и для DNS-серверов. Если DNS-сервер не имеет записи о запрашиваемом ресурсе и не знает, какой DNS-сервер содержит такую запись, он передает запрос вышестоящему DNS-серверу.

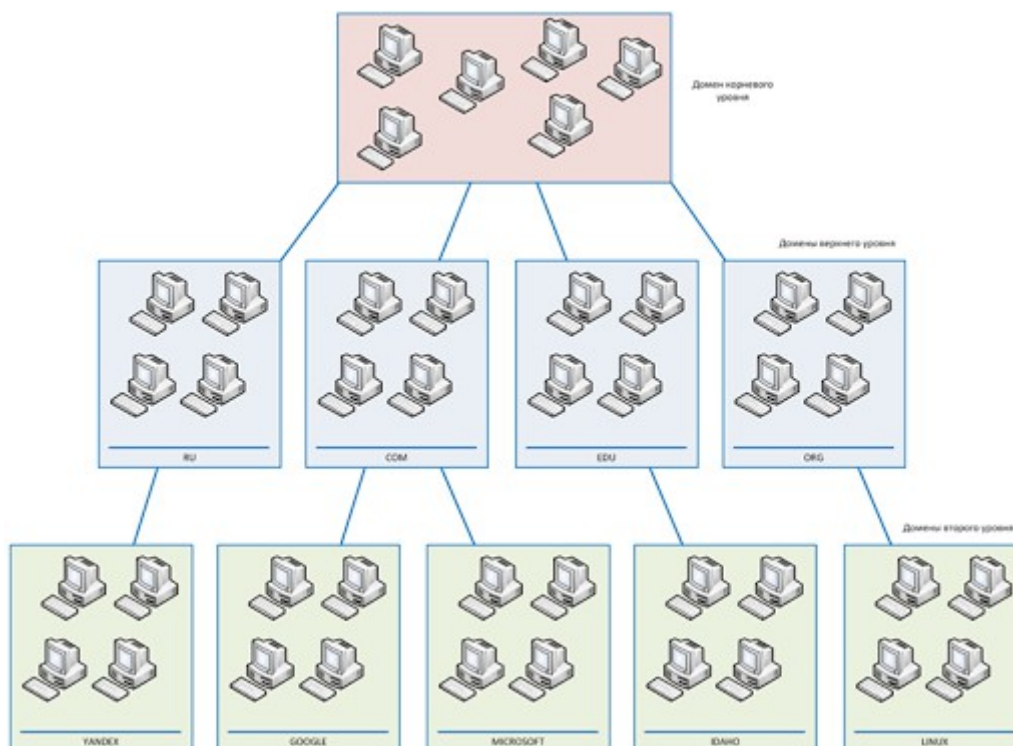


Рисунок 5-41. Иерархия имен DNS похожа на иерархию маршрутизации в Интернете

Схема именования в Интернете похожа на перевернутое дерево с корневыми серверами на вершине. Нижние ветви этого дерева делятся на высокоуровневые домены, под которыми находятся домены второго уровня. Наиболее популярные высокоуровневые домены перечислены ниже:

- **COM** - коммерческие
- **EDU** - образовательные
- **MIL** - американские военные организации
- **INT** - международные организации
- **GOV** - правительственные
- **ORG** - организации
- **NET** - сети

Но как работают вместе все эти DNS-серверы в Интернете? Например, если пользователь вводит URL веб-сайта, торгующего компьютерными книгами, его компьютер запрашивает корпоративный (первичный) DNS-сервер для преобразования адреса сайта в его IP-адрес. Поскольку запрашиваемый веб-сайт вероятно находится вне корпоративной сети компании, этот DNS-сервер не знает его адрес. Но он не просто отвергает запрос пользователя, а пересылает его другому DNS-серверу в Интернете. Запрос на преобразование имени веб-сайта проходит через различные DNS-серверы пока не дойдет до того, который знает IP-адрес нужного пользователю сайта. Эта информация отправляется обратно на компьютер пользователя, который, используя полученный адрес, пытается установить соединение с веб-сайтом, давая возможность пользователю купить компьютерную книгу.

DNS-серверы и преобразование имен узлов чрезвычайно важны при использовании корпоративных сетей и Интернета. Без них пользователи вынуждены были бы помнить и набирать IP-адреса для каждого веб-сайта и отдельной системы, вместо имени. Это привело бы к хаосу.

Угрозы DNS

Как было сказано ранее, ни один из DNS-серверов не знает адреса всех узлов, запрос на преобразование имен которых он может получить. Когда DNS-сервер (сервер А) получает запрос на преобразование имени узла в IP-адрес, сервер просматривает свои записи ресурсов в поисках информации, необходимой для ответа на запрос. Если у сервера нет соответствующей записи, он пересылает запрос на другой DNS-сервер (сервер В), который просматривает свои записи ресурсов и, в случае нахождения необходимой информации, отправляет ее обратно серверу А. Сервер А кэширует в своей памяти полученную информацию о связи между IP-адресом и именем узла (на случай, если другой клиент направит такой же запрос) и пересылает ее запросившему клиенту.

Запомним это и рассмотрим следующий сценарий. Например, Энди хочет сделать так, чтобы пользователи сети Интернет при попытке посетить веб-сайт компании конкурента попадали на веб-сайт компании Энди. Для этого Энди устанавливает средства для перехвата исходящих с сервера А запросов на другие DNS-серверы для преобразования имени сайта компании конкурента в IP-адрес. Когда Энди перехватит запрос сервера А на сервер В для получения IP-адреса сайта компании конкурента, Энди по-быстрому отправит серверу А в ответ на его запрос IP-адрес своего сайта. При этом, программное обеспечение сервера А принимает первый поступивший ему ответ, кэширует полученную информацию и пересылает ее запрашивающему клиенту. Теперь, когда клиент попытается зайти на веб-сайт конкурента компании Энди, он вместо этого попадет на веб-сайт компании Энди. То же самое произойдет с любым пользователем, который воспользуется сервером А для преобразования имени сайта компании конкурента в IP-адрес, т.к. на сервере А будет кэширована поддельная информация.

ПРИМЕЧАНИЕ. Посмотрите еще раз Домен 02, чтобы вспомнить, как происходит атака фарминга DNS.

Вышеуказанная уязвимость является следствием того, что сервер А при получении ответа на свой запрос не проводит аутентификацию его отправителя. Минимизация угроз DNS требует применения множества мер, самой важной из которых является использование строгого механизма аутентификации, такого как **DNSSEC** (DNS Security, являющегося частью многих современных реализаций программного обеспечения DNS-серверов). Если на сервере А включена функция DNSSEC, этот сервер будет проверять электронную цифровую подпись в ответном сообщении перед принятием полученной информации, чтобы убедиться, что она исходит от авторизованного DNS-сервера. Это звучит достаточно просто, однако для полноценного развертывания DNSSEC все DNS-серверы в Интернете должны стать частью PKI, чтобы иметь возможность проверять электронные цифровые подписи (электронные цифровые подписи и PKI будут рассмотрены в Домене 06).

Несмотря на то, что DNSSEC требует гораздо больше ресурсов, чем обычный DNS, все больше и больше организаций начинают применять DNSSEC. Например, американское правительство обеспечило использование DNSSEC для всех своих доменов верхнего уровня (.gov, .mil и т.п.). Такие страны, как Бразилия, Швеция и Болгария уже внедрили DNSSEC для своих доменов верхнего уровня. Кроме того, ICANN заключила соглашение с VeriSign на внедрение DNSSEC для всех своих доменов верхнего уровня (.com, .net, .org и т.п.) к 2011 году.

ПРИМЕЧАНИЕ. Компаниям следует разделять свои DNS-серверы на внутренние и внешние. DNS-сервер в DMZ должен обрабатывать внешние запросы на преобразование имен, а внутренний DNS-сервер – только внутренние запросы. Это создает дополнительный уровень защиты, обеспечивая «невидимость» из сети Интернет внутреннего DNS-сервера.

Теперь давайте обсудим другую сложность в защите DNS – изменение файла *hosts*. Этот метод часто используется вредоносными программами. Файл *hosts* используется операционной системой для хранения информации о связях между IP-адресами и именами узлов. Это простой текстовый файл, который хранится в папке *%system root %\system32\drivers\etc* в системах Windows или в */etc/hosts* в Unix/Linux. Он содержит просто

список IP-адресов и соответствующих им имен узлов.

В большинстве операционных систем настройки сделаны таким образом, что компьютер обращается к файлу `hosts` перед тем, как сформировать запрос на DNS-сервер. Операционные системы отдают предпочтение файлу `hosts`, поскольку обычно он находится под прямым контролем системного администратора.

Как уже было сказано ранее, в первое время существования Интернета, перед появлением концепции DNS, подобные файлы были основным источником информации для определения сетевых адресов узлов по их именам. С постепенным ростом количества подключенных к Интернету узлов, поддержка таких файлов стала почти невозможной, что и привело к созданию Службы доменных имен (DNS).

Файлы `hosts` часто становятся целью для атак вредоносного программного обеспечения, изменение файла `hosts` используется для упрощения процесса распространения этого вредоносного программного обеспечения по системам, подключенным к внутренней сети. Получив контроль над файлом `hosts` и внося в него изменения, вредоносное программное обеспечение может, например, перенаправить трафик с нужных пользователям ресурсов на веб-сайты, на которых размещено вредоносное содержимое. Очень часто вредоносные программы вносят изменения в файл `hosts` с целью блокировки посещения пользователями сайтов антивирусных компаний, а также блокировки загрузки антивирусными программами обновлений вирусных сигнатур с сайтов обновлений. Обычно это делается путем «привязки» блокируемых адресов к виртуальному сетевому интерфейсу (loopback address) 127.0.0.1. Самым эффективным способом предотвращения внесения несанкционированных изменений в файл `hosts` является установка для него атрибута «только чтение» (read-only).

Атакующим не всегда нужно оказывать техническое воздействие на атакуемые системы. Они могут также воспользоваться некоторыми очень простыми методиками, которые имеют высокую эффективность для маршрутизации запросов пользователей к некорректным системам. Самым часто используемым способом является *скрытие URL*. Документы HTML и сообщения электронной почты позволяют пользователям прикреплять или вставлять *гиперссылки* к любому тексту (например, такие ссылки, как «Нажмите здесь» или «Далее» есть на большинстве веб-страниц). Атакующий может воспользоваться этим способом, чтобы обмануть ничего не подозревающего пользователя и заставить его нажать на сформированную злоумышленником ссылку.

Например, рассмотрим следующую ситуацию. Атакующий указал не вызывающий подозрений текст `www.good.site`, но установил для этого текста ссылку на другой сайт – `www.bad.site`. У людей вызывает интерес сайт `www.good.site` и они нажимают на эту ссылку, не зная, что в действительности они переходят на `www.bad.site`. Кроме того, атакующие часто используют похожие символы или коды символов, чтобы минимизировать подозрения пользователя.

Теперь взглянем на некоторые правовые аспекты регистрации доменов. Хотя они не представляют прямого риска безопасности вашим серверам DNS или вашей ИТ-инфраструктуре, их игнорирование может поставить под угрозу владение вами вашим доменным именем в сети Интернет и ваше присутствие в Интернет. Осведомленность о проблемах, связанных с *захватом доменов* (domain grabbing) и *кибер-сквоттингом* (cyber squatting) позволят вам лучше спланировать ваше присутствие в сети Интернет и избежать подобных проблем.

ICANN продвигает модель управления, которая следует политике «первым пришел – первым обслужен» при регистрации доменных имен, не обращая при этом внимания на торговые марки. Это ведет к необходимости защиты привлекательных и известных доменов от *кибер-сквоттеров* – людей, которые регистрируют на себя домены с известными или авторитетными именами, надеясь продать их позже реальным компаниям, которым они

необходимы для организации своего присутствия в сети Интернет.

Другой тактикой, применяемой кибер-сквоттерами, является отслеживание и регистрация на себя доменов, на которые закончился предыдущий период регистрации. При этом они рассчитывают, что предыдущие владельцы этих доменов забыли вовремя перерегистрировать их и будут вынуждены обратиться к кибер-сквоттеру для выкупа своего доменного имени. Также кибер-сквоттеры находят и регистрируют домены, которые в дальнейшем могут понадобиться реальным компаниям для проведения ребрендеринга.

Для защиты компании от этих угроз нужно позаботиться о том, чтобы регистрировать необходимое доменное имя сразу, как только компания решила запустить новый бренд или новую торговую марку. Также хорошей мерой является регистрация важных доменов на длительный период (5-10 лет) вместо ежегодной перерегистрации. Это снижает вероятность того, что домен успеют «увести» кибер-сквоттеры. Другой эффективной мерой является одновременная регистрация похожих имен. Например, если вы владеете доменом `logicalsecurity.com`, неплохой идеей будет зарегистрировать заодно и домены `logical-security.com` и `logicalsecurity.net` – это предотвратит получение их кем-то другим в своих целях.

7.3. NIS

NIS (Network Information System - Информационная система сети) похожа на телефонную книгу («Желтые страницы»), она позволяет определить местонахождение сетевых ресурсов. Используя сервер NIS, пользователи и приложения могут находить и использовать файлы и программы в любом месте в сети. Обычно NIS используется для «отслеживания» парольных файлов, алиасов электронной почты и списков узлов.

ПРИМЕЧАНИЕ. Список узлов (host table) – это файл, в котором хранится информация о связях между именами узлов и их IP-адресами. Он используется подобно DNS, но это просто файл, который компьютер может использовать для определения связи между именем узла и соответствующим ему IP-адресом – это не технология или продукт. NIS обычно сравнивают с DNS, поскольку оба они предоставляют необходимым компьютерам механизмы для определения IP-адресов систем.

В среде Unix системы используют определенные конфигурационные файлы (пароли, сетевые настройки, учетные записи пользователей), обычно работу сети проще организовать, если все системы используют идентичные конфигурационные файлы. NIS позволяет реализовать централизованное хранение и поддержку всех этих конфигурационных файлов, а не хранить и поддерживать их отдельно на каждом компьютере. При этом, когда компьютер загружается, он не смотрит свои собственные конфигурационные файлы, он сразу обращается к серверу NIS, который предоставляет ему файлы с описанием групп, паролями, списком узлов, служб и номеров их портов, а также информацию о ресурсах домена.

NIS работает с использованием архитектуры клиент/сервер. На рабочих станциях запускается клиентский процесс (`yplibd`), который находит серверный процесс NIS (`yplibd`) посредством отправки широковещательного запроса. Это происходит при загрузке рабочей станции, либо когда приложению нужно найти определенную информацию, например, об адресе сетевого сервера печати. Это похоже на работу службы DHCP. Вместо того чтобы использовать локальные файлы или настройки (IP-адрес компьютера, сетевые настройки), компьютер в процессе загрузки запрашивает DHCP-сервер. DHCP-сервер отправляет компьютеру необходимые ему для работы в сети настройки, которые этот компьютер применяет. Это позволяет обеспечить идентичность настроек всех систем, что необходимо для эффективной передачи данных по сети.

Как и любой другой протокол или технология, разработанный в 70-х – 80-х годах, NIS был разработан только с учетом функциональности, но не безопасности. Например, при его использовании файлы с зашифрованными паролями доступны любому запрашивающему компьютеру. Хакеры могут использовать эту возможность для перехвата парольного файла и

проведения брутфорс-атаки для получения возможности несанкционированного доступа. Вся информация передается сервером NIS в открытом виде, что позволяет атакующим перехватывать ее, а перед отправкой критичных файлов сервер NIS не проводит аутентификацию запрашивающей их системы.

В другой атаке используется маскарading для подмены сервера NIS. Например, Валери хотела бы получить доступ к компьютеру Марка, но она не имеет необходимой для этого учетной записи. Поэтому Валери сначала отправляет широковещательный запрос серверу NIS для получения существующих конфигурационных файлов, как это делает любой другой компьютер в сети. Затем, она изменяет эти файлы, добавляя в них свою учетную запись и пароль. После этого она прослушивает сеть, ожидая широковещательный запрос серверу NIS от компьютера Марка. Как только компьютер Марка делает такой запрос, Валери отправляет ему измененные файлы. Система Марка применяет эти файлы, которые создают учетную запись для Валери, и теперь она может получить доступ к компьютеру Марка, когда захочет.

Таким образом, эти недостатки могут дать существенные преимущества атакующему, поэтому возникла потребность в обновлении NIS. Умные люди исправили многие из этих проблем и добавили плюс в название, NIS+.

NIS+ имеет повышенную производительность и усиленную безопасность. Пространство имен NIS является плоским, оно прекрасно работает в маленьких сетях, где не требуется большая масштабируемость. NIS+ использует иерархическое пространство имен, аналогичное DNS. Использование такого типа структуры пространства имен позволяет адаптировать этот сервис к реальной сети компании и легко наращивать его при необходимости. NIS требует ручного обновления (с основного сервера NIS на все подчиненные серверы NIS), которое может занимать по несколько часов в ежедневно, поскольку оно реализуется посредством пакетной обработки. NIS+ позволяет автоматизировать обновления, и производить их инкрементально. NIS хранит данные в таблицах с двумя столбцами, тогда как NIS+ хранит информацию в 16 предварительно созданных таблицах. Таблица 5-9 резюмирует различия между NIS и NIS+.

Характеристика	NIS	NIS+
Структура имен	Плоская	Иерархическая
Хранение данных	Хранятся в таблицах с двумя столбцами	Хранятся в таблицах с множеством столбцов
Безопасность	Нет безопасности	Аутентификация, авторизация и шифрование
Источник данных	Единственный вариант	Клиент выбирает: NIS, NIS+, DNS или локальные файлы
Обновления	Обновления посредством пакетной обработки	Инкрементальные обновления распространяются мгновенно

Таблица 5-9. Различия между NIS и NIS+

В действительности, для обеспечения безопасности разработчики просто добавили S-RPC (Secure Remote Procedure Call). Протокол RPC рассматривается в Домене 09, но сейчас нам просто нужно знать, что он позволяет взаимодействовать клиентским и серверным системам в рамках модели клиент/сервер. Если клиенту NIS нужно взаимодействовать с сервером NIS или наоборот, их взаимодействие происходит посредством протокола RPC на сеансовом уровне. Использование S-RPC дало NIS+ функциональность аутентификации, авторизации и шифрования. Например, когда пользователю или компьютеру нужен доступ к серверу NIS+, производится проверка его идентификационных данных и пароль S-RPC. Затем пользователь помещается в класс (owner, group, world, nobody), который определяет, что пользователь может и что не может делать с объектами NIS+. Все коммуникации между клиентом NIS+ и сервером – зашифрованы, что исключает возможность их перехвата.

NIS+ обратно совместим с NIS, что создает уязвимость, которой могут воспользоваться хакеры. Если на системе Валери установлено программное обеспечение клиента NIS, а сервер NIS+ настроен на обратную совместимость, клиент NIS может получить файлы без

предварительной аутентификации и авторизации. Таким образом, Валери может получить парольный файл и приступить к его взлому. Эта уязвимость обычно остается доступной, если сетевой администратор не понимает разницы между этими двумя версиями сервиса и не думает о последствиях применения таких настроек.

Кроме того, сам NIS+ может быть также атакован. Если атакующий взломал Unix-систему, на которой установлено программное обеспечение сервера NIS+, он может изменить настройки, добавить себя в качестве пользователя во все системы, получить доступ к парольным файлам.

NIS+ может быть настроен для работы в одном из трех режимов безопасности:

- **Уровень 0.** Средства безопасности отключены. Любая система или пользователь имеют полный доступ к объектам NIS+. Такой режим следует использовать только для работы в тестовом режиме.
- **Уровень 1.** Обеспечивает низкий уровень безопасности, не требует аутентификации. Этот режим также следует использовать только в процессе тестирования.
- **Уровень 2.** Режим по умолчанию, применяется аутентификация и авторизация. Если кто-то делает запрос без учетных данных, он помещается в класс nobody, который запрещает любые действия.

7.4. Службы каталогов

Служба каталогов (directory service) хранит иерархическую базу данных пользователей, компьютеров, принтеров, ресурсов и атрибутов каждого из них. Этот каталог используется в основном для поиска информации, позволяя пользователям легко находить нужные им ресурсы и других пользователей для получения доступа. Большинство баз данных служб каталогов построено на основе модели X.500, для доступа к ним используется LDAP (Lightweight Directory Access Protocol), который мы рассмотрим далее в этом Домене.

Службы каталогов часто сравнивают с телефонными книгами, которые вы просматриваете в поисках нужной вам контактной информации. Хотя службы каталогов обычно содержат гораздо больше информации, чем просто телефонные номера. DNS в действительности является разновидностью службы каталогов.

Сам каталог использует классы и подклассы объектов для создания репозитория каталога, обычно представляющего из себя базу данных. Администратор может централизованно применять политики к этим объектам. В этих объектах может содержаться информация о пользователях, их местонахождении, информация о периферийных устройствах, ресурсах, профилях, сетевых сервисах и т.п. С помощью этих объектов, администратор может разработать политики управления доступом, политики безопасности и аудита, в которых будет определено, кто и как может использовать объекты, какие действия должны журналироваться. Также, могут быть разработаны и применены политики для управления шириной полосы пропускания канала, фильтрацией межсетевого экрана, VPN-доступом, QoS.

Департамент ИТ создает и поддерживает множество различных каталогов. Эти каталоги могут быть основаны на потребностях бизнеса или безопасности, к ним могут применяться различные политики безопасности, управления доступом, а также различные профили. Если используется более одного каталога, им требуется способ взаимодействия друг с другом – это осуществляется посредством *мета-каталогов* (meta-directories). *Метаданные* (metadata) – это данные о данных. Мета-каталоги содержат высокоуровневую информацию о самом каталоге, что позволяет пользователю одного каталога быстро находить объект, проводя его поиск по всем каталогам одновременно.

Каждый каталог следует определенной *схеме* (schema), как обычная база данных. Схема

обеспечивает структуру репозитория каталога и определяет, как должны быть представлены объекты и их взаимоотношения. Каждый производитель службы каталога имеет базовую схему, которая позволяет администраторам определять их собственные объекты и соответствующие им атрибуты. Однако, как и в продуктах других типов, могут возникнуть проблемы взаимодействия между схемами различных производителей, что усложнит организацию взаимодействия между ними. Если компания покупает другую компанию и возникает необходимость в объединении их сетей и слиянии их служб каталогов – это может оказаться очень сложным проектом.

Службы каталогов предлагают широкие возможности пользователям, администраторам и сетям в целом. Они позволяют администраторам поддерживать и управлять всеми ресурсами и пользователями сети. База данных каталога работает как место хранения почти всей важной информации сети и позволяет пользователям легко и быстро находить нужные им сервисы или ресурсы. Двумя примерами служб каталогов являются Microsoft Active Directory (AD) и Novell Directory Service (NDS). Хотя обе они основаны на модели X.500, достаточно сложно организовать взаимодействие между ними.

Пользователям в сети доступно множество сервисов, что и было основной причиной ее создания. В этом разделе мы рассматриваем NOS, DNS, NIS и службы каталогов, но в действительности сети предоставляют гораздо больше сервисов – например, многие сети предоставляют сервисы печати, которые позволяют нескольким пользователям совместно использовать локальные или удаленные принтеры. Администраторам предоставляются сервисы для централизованного управления сетью, которые дают им возможность обзора сети в целом из одного приложения с графическим интерфейсом, позволяющего добавлять и удалять пользователей, решать возникающие в сети проблемы, проводить аудит действий пользователей и событий в сети, добавлять и удалять сервисы, управлять доступом удаленных пользователей и т.д. Некоторые сети предоставляют терминальные службы, позволяющие пользователям использовать маломощные рабочие станции, которые просто отображают удаленный рабочий стол, а операционная система сервера выполняет всю необходимую работу.

Сети интересны и очень востребованы в современном компьютерном мире. Однако сети достаточно сложны, что является причиной многих ошибок, дыр в безопасности, уязвимостей – всего того, на что рассчитывают атакующие. Чем лучше вы понимаете организацию работы сетей и их компонентов, тем более эффективные механизмы безопасности вы можете внедрить и обеспечить более высокий уровень защиты сети.

7.5. LDAP

LDAP (Lightweight Directory Access Protocol – Упрощенный протокол доступа к каталогу) – это клиент-серверный протокол, используемый для доступа к сетевым каталогам, таким как Microsoft AD или NDS. Эти каталоги следуют стандарту X.500. Первой версией этого протокола был протокол DAP (Directory Access Protocol), созданный для реализации клиентской части службы каталога X.500. Его идея заключалась в предоставлении интерфейса для каждого сервиса, предоставляемого каталогом стандарта X.500. Однако стандарт X.500 был слишком сложным для полноценной реализации, соответственно и протокол DAP был чрезвычайно сложен и ресурсоемок. Поэтому современные службы каталогов используют только часть стандарта X.500, а мы используем урезанную (упрощенную) версию DAP.

Спецификация LDAP работает с каталогами, организованными в виде базы данных с иерархической древовидной структурой. Дерево имеет листья (сущности) с уникальными именами (DN). Эти имена следуют иерархии и описывают место сущности в дереве. Сущностями могут быть сетевые ресурсы, компьютеры, люди, беспроводные устройства и т.д. Каждый элемент имеет атрибут и значение. Атрибуты похожи на столбцы в реляционной базе данных, они содержат информацию об элементе. Например, если элемент является

принтером, атрибутами могут быть его IP-адрес, сетевое имя, MAC-адрес и описание. Значения – это просто данные, которые заполняют соответствующие поля. Таким образом, атрибуты содержат поля и когда эти поля заполняются данными, они являются значениями атрибутов. Каждая компания определяет свою собственную структуру каталога, атрибуты и их значения, которые лучше всего подходят потребностям этой компании.

ПРИМЕЧАНИЕ. Новейшая версия LDAP, версия 3, имеет исчерпывающую встроенную модель безопасности, которая поддерживает стандарты безопасности интернета, такие как TLS (Transport Layer Security).

Ссылки по теме:

- IETF LDAP (v3) Revision Charter
- RFC 2256 – A Summary of the X.500(96) User Schema for Use with LDAPv3

7.6. Трансляция сетевых адресов

Если компьютерам нужно взаимодействовать друг с другом, они должны использовать одинаковый тип схемы адресации, которую понимает и может использовать каждый из них. Интернет использует схему с IP-адресами и любые компьютеры или сети для взаимодействия должны применять эту схему, иначе они будут находиться в «виртуальной комнате» и смогут взаимодействовать только сами с собой.

Однако IP-адресов становится недостаточно (до полного перехода на IPv6) и они дорожают. Поэтому умные люди придумали **NAT** (Network Address Translation - Трансляция сетевых адресов), которая позволяет сетям не следовать схеме адресации Интернета, но при этом иметь возможность взаимодействия через Интернет.

Для использования во внутренних сетях LAN были зарезервированы частные IP-адреса, описанные в RFC 1918. Эти адреса могут использоваться внутри компании, но они не могут использоваться в Интернет, поскольку они не могут маршрутизироваться. NAT позволяет компании использовать эти частные адреса, имея при этом возможность для прозрачного взаимодействия с компьютерами в Интернете.

Приведенный ниже список показывает диапазоны частных IP-адресов:

- **10.0.0.0 – 10.255.255.255** Сеть Класса А
- **172.16.0.0 – 172.31.255.255** 16 сетей Класса В
- **192.168.0.0 – 192.168.255.255** 256 сетей Класса С

NAT – это шлюз, расположенный между сетью и Интернетом (или другой сетью), который выполняет прозрачную маршрутизацию и трансляцию имен. Поскольку количество свободных IP-адресов быстро сокращалось, в 1999 году был разработан протокол IPv6, который должен был полностью решить проблему с количеством адресов. NAT позволяет решить проблему недостатка количества адресов только частично, позволив большему количеству компаний стать частью сети Интернет. Однако на сегодняшний день IPv6 принимается и внедряется достаточно медленными темпами, т.к. NAT временно решил проблему. Многие производители межсетевых экранов реализуют NAT в своих продуктах, что дополнительно позволяет получить существенные преимущества с точки зрения безопасности. Если атакующие хотят взломать сеть, они сначала должны узнать все о сети, ее топологию, сервисы и адреса. Однако, если в сети используется NAT, атакующим не просто узнать схему адресов компании и топологию ее сети, поскольку NAT работает как вышибала в ночном клубе, стоя на переднем крае обороны сети и скрывая реальную схему адресации IP.

NAT скрывает внутренние адреса, объединяя их на одном устройстве, в результате чего любые исходящие из сети кадры имеют в качестве адреса источника только адрес этого

устройства, а не фактический адрес компьютера, отправившего сообщение. Например, если сообщение отправлено внутренним компьютером с адресом 10.10.10.2, это сообщение останавливается на устройстве, на котором запущено программное обеспечение NAT. Это устройство имеет IP-адрес 1.2.3.4. NAT изменяет заголовок кадра, устанавливая вместо внутреннего адреса 10.10.10.2 адрес устройства NAT 1.2.3.4. Когда компьютер в Интернете отвечает на это сообщение, он отправляет ответ на адрес 1.2.3.4. Устройство NAT меняет адрес получателя в заголовке ответного сообщения на 10.10.10.2 и отправляет его по внутренней сети соответствующему пользователю.

Может применяться три основных типа реализации NAT:

- **Статическая трансляция** (static mapping). Программное обеспечение NAT имеет настроенный пул внешних IP-адресов. Каждый частный адрес статически связывается с определенным внешним адресом. При этом компьютер А всегда получает внешний адрес X, компьютер В всегда получает внешний адрес Y и т.д. Обычно такая реализация NAT используется для серверов, которым нужно постоянно иметь один и тот же внешний адрес.
- **Динамическая трансляция** (dynamic mapping). Программное обеспечение NAT также имеет пул внешних IP-адресов, но вместо статической привязки внешних адресов определенным внутренним адресам, динамическая трансляция работает по принципу «первый пришел, первым обслужен». Так, если Бобу нужно взаимодействовать через Интернет, его система делает запрос к серверу NAT. Сервер NAT привязывает первый внешний IP в своем списке к частному адресу Боба. При этом следует оценить, как много компьютеров обычно будет взаимодействовать с внешними сетями одновременно. Эта оценка позволит определить необходимое количество внешних адресов, которые нужно купить компании (вместо покупки по одному внешнему адресу на каждый внутренний компьютер).
- **Трансляция адресов портов** (PAT – Port address translation). Компания имеет и использует только один внешний IP-адрес для всех систем, которым нужно взаимодействовать с внешними сетями. Но как во внешней сети все компьютеры могут использовать один и тот же IP-адрес? Хороший вопрос. Например, устройство NAT имеет внешний IP-адрес 127.50.41.3. Когда компьютеру А требуется взаимодействовать с системой в Интернете, устройство NAT фиксирует частный адрес этого компьютера и исходящий номер порта (10.10.44.3, порт 44887). Устройство NAT меняет IP-адрес в заголовке пакета с адреса этого компьютера на 127.50.41.3 с исходящим портом 40000. Когда компьютеру В также требуется взаимодействовать с системой в Интернете, устройство NAT фиксирует его частный адрес и исходящий номер порта (10.10.44.14, порт 23398) и изменяет информацию в заголовке на адрес 127.50.41.3 с исходящим портом 40001. Когда внешняя система отвечает компьютеру А, пакет сначала идет на устройство NAT, которое видит номер порта 40000 и делает вывод, что пакет предназначен для компьютера А. Поэтому устройство NAT меняет информацию в заголовке на адрес 10.10.44.3 и порт 43887 и отправляет его компьютеру А для обработки. Компания может существенно сэкономить, используя PAT, поскольку нужно купить только несколько внешних IP-адресов, которые будут использоваться всеми системами внутренней сети.

Большинство реализаций NAT *контролируют состояние* (stateful), т.е. они отслеживают взаимодействие между внутренними и внешними узлами до окончания сеанса. Устройство NAT должно помнить внутренний IP-адрес и порт для обратной отправки сообщений. Эти характеристики контроля состояния похожи на межсетевые экраны с контролем состояния, но NAT не выполняют анализ входящих пакетов с целью выявления в них вредоносных компонентов. NAT является службой, обычно выполняющейся на маршрутизаторах или межсетевых экранах в рамках экранированной подсети компании.

Хотя NAT был разработан для временного и быстрого решения проблемы уменьшения числа свободных IP-адресов, в действительности он снизил важность этой проблемы на неопределенное время. Многие компании организовали работу по схемам с частными адресами, снизив свою потребность во внешних IP-адресах. NAT очень полезен, и производители внедряют эту технологию в свои продукты, но из-за нее снизились темпы внедрения IPv6.

Ссылки по теме:

- RFC 1631 – The IP Network Address Translator (NAT)
- “NAT Basics,” HomeNetHelp

8. Интрасети и Экстрасети

Функциональность, возможности и популярности веб-технологий росли последнее время взрывными темпами. Компании создавали свои внутренние веб-сайты для централизации бизнес-информации, такой как номера телефонов сотрудников, нормативные документы, информация о событиях, новости, внутренние инструкции по работе и т.п. Многие компании внедрили веб-терминалы, которые сотрудники используют для выполнения своих повседневных задач, работы с централизованными базами данных, совершения операций, сотрудничества в рамках проектов, использования глобальных календарей, видеоконференций и досок объявлений, работы с техническими и маркетинговыми данными.

Веб-клиенты отличаются от рабочих станций, которые могут входить в сеть и имеют свой собственный «рабочий стол». Веб-клиенты ограничивают возможности получения пользователем доступа к файловой системе компьютера, ресурсам, пространству на жестком диске, доступа к серверным системам, а также выполнения других задач. Веб-клиенты могут быть настроены на предоставление графического интерфейса только с теми кнопками, полями и страницами, которые необходимы пользователям для выполнения своих задач. Это предоставляет всем пользователям стандартный универсальный интерфейс с одинаковыми возможностями.

Если компания в своей внутренней сети применяет интернет- и веб-технологии, она использует **интрасеть** (intranet - интранет) – “частную” сеть, в которой применяются Интернет-технологии, такие как TCP/IP. В этой компании есть веб-серверы и клиентские машины, используются веб-браузеры, применяется набор протоколов TCP/IP. Веб-страницы написаны на HTML или XML, а доступ к ним осуществляется через HTTP.

Использование веб-технологий дает множество плюсов. Они существуют на протяжении довольно длительного времени, они очень легко внедряются, не имеют серьезных проблем взаимодействия – пользователь просто щелкает по ссылке и сразу перемещается к запрашиваемому ресурсу. Веб-технологии не зависят от платформы, то есть, например, все веб-сайты и страницы могут храниться на Unix-сервере, а пользовательские рабочие станции под управлением Windows или MacOS могут использовать их, для чего им потребуется только веб-браузер.

Экстрасеть (extranet - экстранет) распространяется за пределы сети компании, позволяя двум или нескольким компаниям совместно использовать информацию, создавать общие ресурсы. Бизнес-партнеры обычно создают экстрасеть для обеспечения взаимодействия между своими компаниями. Экстрасеть позволяет бизнес-партнерам вместе работать над проектами, делиться маркетинговой информацией, общаться и сотрудничать в различных вопросах, отправлять почту, каталоги, информацию о предстоящих событиях и т.п. Торговые партнеры часто используют электронный обмен данными (EDI – Electronic Data Interchange), позволяющий использовать электронный документооборот, в рамках которого передаются и совместно используются приказы, счета, заказы и другие данные. EDI использует веб-технологии для предоставления простого доступа и простых методов взаимодействия.

Однако экстрасеть может стать уязвимостью или «дырой» в безопасности компании, если она неправильно реализована или не поддерживается должным образом. Необходимо правильно настроить межсетевые экраны для управления доступом и использованием коммуникационных каналов экстрасети. Экстрасеть лучше реализовать на основе выделенных каналов связи, что значительно затруднит злоумышленникам проникновение в сети компаний. Однако сегодня многие экстрасети создаются посредством сети Интернет, что требует применения правильно настроенных VPN-каналов и политик безопасности.

Сети с дополнительными услугами. Множество различных компаний используют EDI для организации внутренних коммуникаций и связи с другими компаниями. Очень распространено использование такой связи между компанией и ее поставщиками. Например, некоторые поставщики поставляют оборудование в различные компании, такие как Target, Wal-Mart и Kmart. Многие такие поставки осуществляются из Китая, после чего оборудование направляется на склад в Соединенных Штатах. Когда Wal-Mart нужно заказать оборудование, она направляет свой заказ через сеть EDI, которая основана на электронных формах, а не бумажных документах. Сеть с дополнительными услугами (VAN – Value-Added Network) – это инфраструктура EDI, разработанная и поддерживаемая сервис-бюро (service bureau). Wal-Mart отслеживает свое оборудование, находящееся у работников, сканируя штрих-коды отдельных устройств. Когда какого-либо оборудования остается мало, работник Wal-Mart направляет заказ на следующую поставку конкретного оборудования. Этот заказ направляется в специальный почтовый ящик в VAN, а затем пересылается поставщику, который поставляет этот тип оборудования для Wal-Mart. Поскольку Wal-Mart (как и некоторые другие магазины) работает с тысячами поставщиков, применение VAN упрощает процесс оформления заказов – вместо того, чтобы работник искал нужного поставщика и направлял ему заказ, все это происходит в фоновом режиме через автоматизированную сеть EDI, которая управляется компанией, реализующей VAN (называемой сервис-бюро) для использования другими компаниями.

EDI переходят от собственных структур VAN EDI на стандартизированные коммуникационные структуры, чтобы использовать больше возможностей для взаимодействия, решить проблемы совместимости и упростить обслуживание. Для этого используются XML, SOAP и веб-службы.

Ссылки по теме:

- Intranet Road Map (guide and tutorial)
- Intranet Journal

9. Городские вычислительные сети

Городская вычислительная сеть (MAN – Metropolitan area network) – это, как правило, магистраль, соединяющая сети LAN друг с другом, сети LAN и WAN между собой, Интернет, телекоммуникационные и кабельные сети. Большинство современных сетей MAN являются **Сетями синхронной оптической связи** (SONET – Synchronous Optical Network) или кольцами FDDI, представленными провайдерами телекоммуникационных услуг (технологии FDDI обсуждались ранее в этом Домене). Эти кольца охватывают большие площади, компании могут подключаться к кольцам через каналы T1 или T3. На Рисунке 5-42 показаны две компании, соединенные с помощью кольца SONET, а также устройства, которые обычно необходимы для организации такого вида коммуникаций. Это упрощенный пример MAN. В реальности множество компаний подключены к одному кольцу.

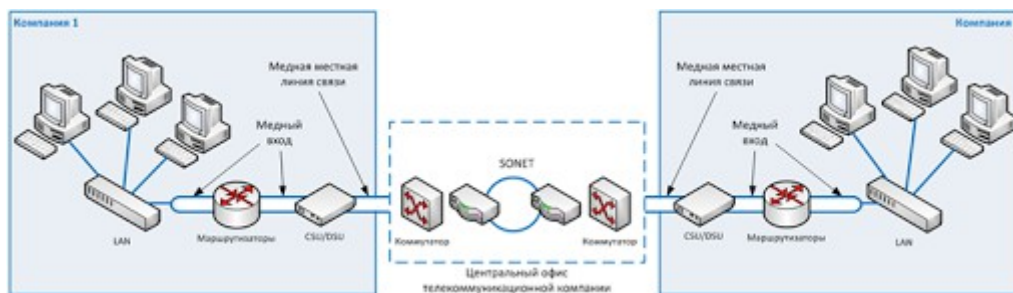


Рисунок 5-42. MAN покрывает большую площадь и позволяет компаниям подключаться друг к другу, к Интернету или к другим соединениям WAN

Фактически SONET является стандартом для телекоммуникаций посредством волоконно-оптических кабелей. Операторы и телефонные компании имеют множество сетей SONET в Северной Америке. При этом если они следуют стандартам SONET, их сети могут взаимодействовать друг с другом без существенных трудностей.

SONET является *самовосстанавливающейся системой* (self-healing), т.е. в случае разрыва канала передачи, он может задействовать дополнительное резервное кольцо для обеспечения непрерывности работы. Все каналы и кольца SONET обладают избыточностью. Избыточные каналы используются в случае, если что-то случается с основным кольцом.

Сети SONET могут передавать голос, видео и данные по оптическим сетям.

Низкоскоростные сети SONET часто подключают к более крупным и быстрым сетям SONET, как это показано на Рисунке 5-43. Это позволяет компаниям в разных городах и регионах взаимодействовать друг с другом.

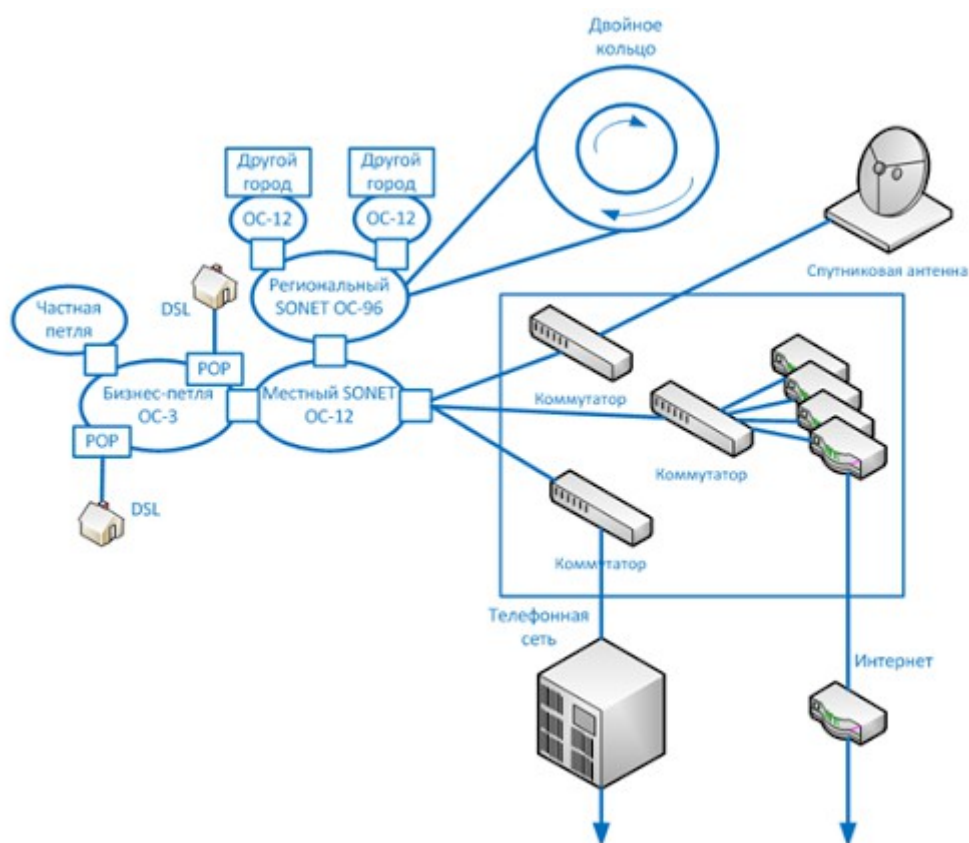


Рисунок 5-43. Небольшие кольца SONET подключаются к более крупным для создания отдельных сетей MAN

Ссылки по теме:

- RFC 2615 – PPP over SONET/SDH
- TechFest WAN links

10. Глобальные вычислительные сети

Технологии LAN предоставляют коммуникационные возможности в небольшой географической области, тогда как технологии *глобальных вычислительных сетей* (WAN – Wide area network) позволяют реализовать связь на больших географических расстояниях. Технологии LAN работают с тем, как компьютер помещает свои данные в сетевой кабель, с правилами и протоколами формирования и передачи этих данных, с процедурами обработки ошибок, порядком извлечения компьютером получателя данных из кабеля. Но если компьютеру в одной сети требуется взаимодействовать с другой сетью, находящейся в

другом конце страны или в другой стране, применяются технологии WAN.

Сеть должна иметь какие-либо соединения с другими сетями, обычно это реализуется с помощью маршрутизатора, имеющего подключение к маршрутизаторам провайдера услуг или телефонной компании. В WAN применяется несколько видов различных технологий, точно так же, как и в LAN. Этот раздел рассматривает многие из этих технологий WAN.

10.1. Эволюция телекоммуникаций

Телефонные системы существуют на протяжении уже около 100 лет, они начинались с использования аналоговых систем, основанных на медных проводах. На центральной коммутационной телефонной станции изначально отдельные телефоны соединяли вручную люди (операторы), затем было внедрено электронное коммутационное оборудование. После соединения двух телефонов, они получали сквозное соединение друг с другом. Для нескольких телефонных вызовов использовался один и тот же провод, что называется мультиплексированием. **Мультиплексирование** (multiplexing) – это метод объединения нескольких каналов данных в едином коммуникационном канале. При этом передача осуществляется достаточно быстро и эффективно, абоненты на разных концах даже не знают, что они разделяют линию со многими другими абонентами. Для них все выглядит так, как будто они имеют собственную выделенную линию.

В середине 1960-х годов, появились цифровые телефонные системы с магистральными линиями (trunk) T1, которые обеспечивали 24 канала голосовой связи всего на двух парах медных проводов. Это позволило достичь скорости передачи в 1,544 Мбит/с, что было не только быстрее, но и давало возможность мультиплексировать больше телефонных вызовов в одном проводе. Если выполняются вызовы между различными коммутационными телефонными станциями (местные телефонные звонки), они мультиплексируются по линиям T1. Если требуется совершить звонок на большее расстояние, то пришедшие по линии T1 вызовы мультиплексируются по линиям T3, которые могут передавать до 28 линий T1. Это показано на Рисунке 5-44.

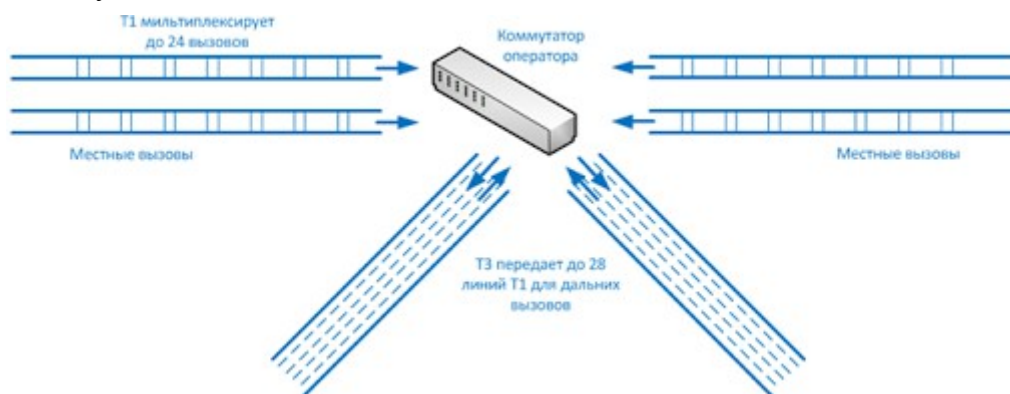


Рисунок 5-44. Местные вызовы мультиплексируются в линии T1, а дальние вызовы перемещаются с линий T1 и мультиплексируются в линиях T3

Протоколы LAN и WAN. Количество ошибок передачи в среде LAN ниже, чем в среде WAN, что следует учитывать при сравнении сложности этих сред. Трафик WAN может передаваться на сотни и тысячи километров, проходить через различные виды устройств, кабелей и протоколов. Из-за этого различия большинство протоколов WAN предварительно устанавливают соединение. Протоколы с предварительным установлением соединения обеспечивают надежную передачу данных, поскольку у них есть возможности для обнаружения и исправления ошибок.

Следующим пополнением телекоммуникационных технологий было оптическое волокно, которое позволило мультиплексировать еще больше вызовов в одной магистральной линии на еще больших расстояниях. Затем появились оптические технологии, такие как SONET, которые передавали оцифрованные голосовые сигналы в пакетах данных. SONET – это стандарт для передачи данных по волоконно-оптическим кабелям. Этот стандарт

устанавливает необходимые параметры для передачи цифровой информации посредством оптических систем. Телекоммуникационные операторы использовали эту технологию для мультиплексирования низкоскоростных оптических соединений в высокоскоростные соединения, подобно тому, как более низкоскоростные соединения LAN подключаются к высокоскоростным соединениям WAN в наше время. Рисунок 5-45 показывает пример соединенных вместе колец SONET.

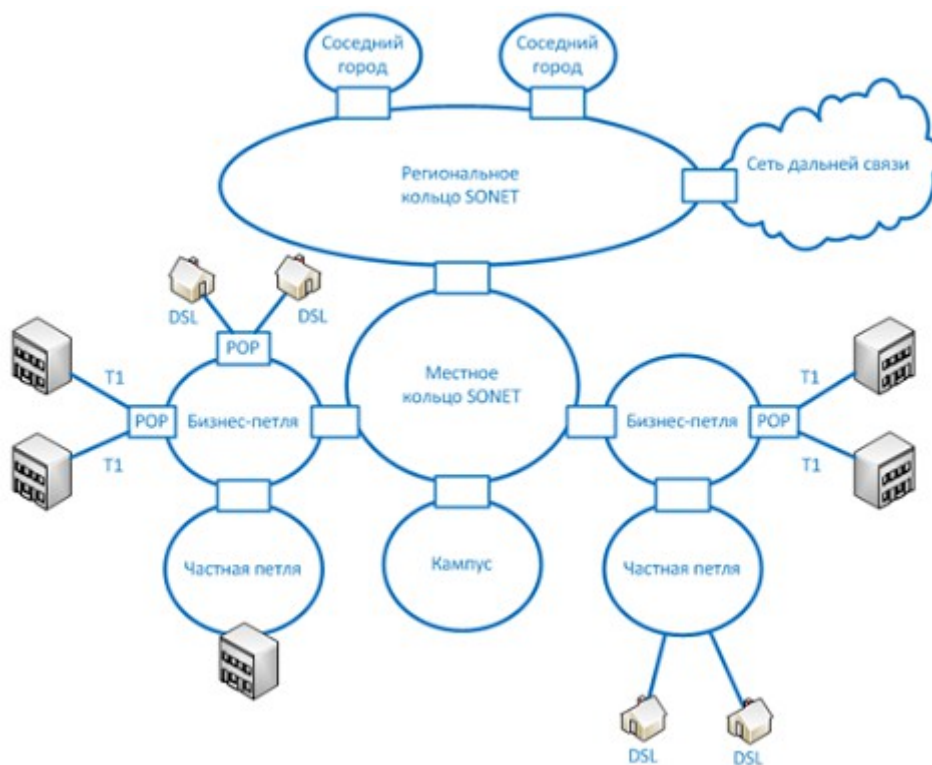


Рисунок 5-45. Технология SONET позволяет взаимодействовать множеству оптических коммуникационных петель

Рисунок 5-45 показывает, каким образом телекоммуникационные операторы могут предоставлять телефонную связь и доступ в Интернет для компаний и частных лиц на больших территориях. Стандарт SONET позволяет взаимодействовать всем операторам.

Следующим эволюционным шагом в истории телекоммуникаций была технология ATM (Asynchronous Transfer Mode - Асинхронный режим передачи). ATM инкапсулирует данные в фиксированные ячейки и может использоваться для передачи данных через сеть SONET. Для описания взаимодействия SONET и ATM, представьте, что SONET – это шоссе, по которому едут автомобили (пакеты ATM).

ATM – это высокоскоростная сетевая технология, которая используется операторами и провайдерами, а также телефонными компаниями в реализациях LAN и WAN. ATM использует фиксированный размер ячейки вместо кадров переменного размера, используемых в более ранних технологиях. Фиксированный размер ячеек обеспечивает более высокую производительность и сокращает накладные расходы на обработку ошибок (более подробно технологию ATM мы рассмотрим позже в разделе «ATM»).

Ниже указаны основные этапы истории телекоммуникаций:

- Медные линии передают аналоговые сигналы
- Линии T1 передают до 24 разговоров
- Линии T3 передают до 28 линий T1
- Используются сети SONET и оптоволокно

- Используется АТМ через SONET

SONET был разработан в США для достижения скорости передачи данных около 50 Мбит/с с целью поддержки потока данных от линий T1 (1,544 Мбит/с) и линий T3 (44,736 Мбит/с). Данные перемещаются через эти T-каналы к границе сети SONET в виде электрических напряжений. Затем напряжения должны быть преобразованы в свет для передачи по волоконно-оптическим линиям, известным как линии оптической передачи (ОС – Optical Carrier). Каждый кадр ОС-1 передается на скорости 51,84 Мбит/с с пропускной способностью 44,738 Мбит/с.

ПРИМЕЧАНИЕ. Оптические линии передачи могут обеспечить различные значения пропускной способности: ОС-1 = 51,84 Мбит/с, ОС-3 = 155,52 Мбит/с, ОС-12 = 622,08 Мбит/с и т.д.

В Европе используется иная инфраструктура, там выбрали использование SDH (Synchronous Digital Hierarchy), которая поддерживает линии E1 (2,048 Мбит/с) и E3 (34,368 Мбит/с). SONET является стандартом для Северной Америки, а SDH является стандартом для всего остального мира. SDH и SONET похожи, но несовместимы из-за имеющихся отличий. Для обеспечения взаимодействия SONET и SDH между ними должен быть установлен шлюз, который будет осуществлять трансляцию сигналов.

В настоящее время разрабатываются многие другие технологии, направленные на увеличение объема данных, который может быть эффективно передан в единицу времени.

10.2. Выделенные линии

Выделенные линии (dedicated link) также называют арендованными линиями (leased line) или соединениями «точка-точка» (point-to-point link). Это отдельные линии связи, которые предварительно созданы для организации WAN-коммуникаций между двумя пунктами (абонентами). Они являются *выделенными* (dedicated), что означает, что только абоненты на обоих концах выделенной линии могут взаимодействовать между собой. Выделенная линия никогда не может совместно использоваться кем-либо другим, кроме абонентов на двух ее концах. Раньше это был основной способ взаимодействия для компаний, потому что не было так много вариантов как сейчас. Создание выделенной линии является хорошим вариантом для соединения двух пунктов (например, двух компаний-партнеров), которые будут часто взаимодействовать друг с другом и которые требуют высокоскоростной передачи данных и определенной полосы пропускания. Однако выделенные линии стоят дорого по сравнению с другими возможными технологиями, которые позволяют нескольким компаниям совместно использовать одну и ту же пропускную способность, а также совместно нести расходы за связь. Конечно, это не означает, что выделенные линии не используются – они обязательно используются, просто нужно учитывать, что имеются и многие другие варианты, в том числе X.25, Frame Relay и технологии АТМ.

Цифровые телекоммуникационные системы

Цифровые телекоммуникационные системы (T-carriers, T-каналы) являются выделенными линиями, которые могут передавать голосовую информацию и данные по магистральным линиям. Они были разработаны АТ&Т и впервые реализованы в начале 1960-х годов для передачи голоса с использованием импульсно-кодовой модуляции (PCM – Pulse-code modulation). Изначально это использовалось для передачи голоса в цифровом виде по выделенной, двухточечной линии связи с большой пропускной способностью. Наиболее часто используемые T-каналы – это линии T1, которые обеспечивают скорость до 1,544 Мбит/с и линии T3, которые обеспечивают скорость до 45 Мбит/с, о чем говорилось ранее. Оба варианта являются цифровыми, они мультиплексируют несколько отдельных каналов в одном высокоскоростном канале.

Эти линии выполняют мультиплексирование посредством TDM (Time-division multiplexing – Мультиплексирование с разделением по времени). Что это означает? Рассмотрим линию T1, которая может мультиплексировать до 24 каналов. Предположим, компания имеет офисную АТС, подключенную к линии T1, которая, в свою очередь, подключена к коммутационной станции телефонной компании. При этом по линии T1 могут одновременно передаваться на коммутационную станцию телефонной компании 24 вызова. Если бы эта компания не использовала линию T1, ей потребовалось бы 24 отдельных провода витой пары для одновременного выполнения того же количества вызовов.

Как показано на Рисунке 5-46, данные помещаются в эти 24 канала и передаются. Каждый канал получает до восьми бит для вставки в созданный таймслот (time slot). Двадцать четыре таких восьми-битных таймслота составляют кадр T1. Кажется, что это не большой объем информации, однако следует учесть, что в секунду создается 8000 кадров. Поскольку все это происходит довольно быстро, принимающая сторона не замечает задержек и не знает, что это соединение используется не только ей, но и 23 другими устройствами.

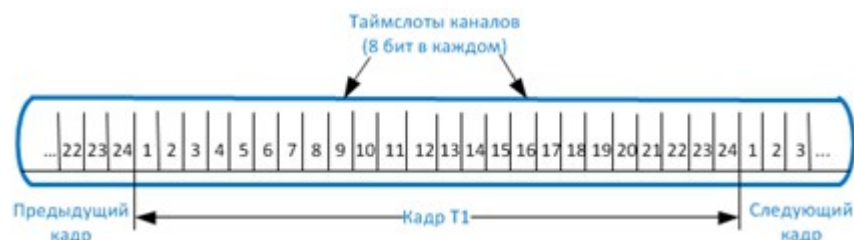


Рисунок 5-46. С помощью мультиплексирования несколько телефонных вызовов или сеансов передачи данных одновременно передаются по одному проводу

Изначально компаниями-операторами использовались линии T1 и T3, но постепенно они были заменены в основном оптическими линиями. Теперь линии T1 и T3 передают данные внутри этих мощных и супербыстрых оптических линий. Линии T1 и T3 сдаются в аренду компаниям и провайдерам, которым требуется высокая пропускная способность. Иногда, каналы T1 совместно используются несколькими компаниями, каждой из которых в отдельности не требуется полная пропускная способность в 1,544 Мбит/с. Это называется *частичным T-каналом* (fractional T-line). Различные каналы передачи и соответствующие им характеристики указаны в таблице 5-10.

Канал	Кол-во T1	Кол-во голосовых каналов	Скорость (Мбит/с)
Частичный	1/24	1	0,064
T1	1	24	1,544
T2	4	96	6,312
T3	28	672	44,736
T4	168	4032	274,760

Таблица 5-10. Резюме по иерархии цифровых телекоммуникационных каналов

Как было сказано ранее, выделенные линии имеют свои недостатки. Они стоят дорого и не являются гибкими. Если компания переезжает на другое место, линия T1 не может последовать за ней. Выделенная линия стоит дорого, поскольку компании приходится платить за выделенное соединение с большой пропускной способностью, даже если эта компания не использует всю полосу пропускания. Не многим компаниям требуется такая пропускная способность 24 часа в сутки. Большинство компаний отправляет данные время от времени, но не постоянно.

Стоимость выделенной линии определяется расстоянием между абонентами. Поэтому линия T1 между двумя зданиями, находящимися на расстоянии в 2 километра, гораздо дешевле аналогичной линии длиной в 50 километров.

Повышенное мультиплексирование. Существуют другие виды функциональности

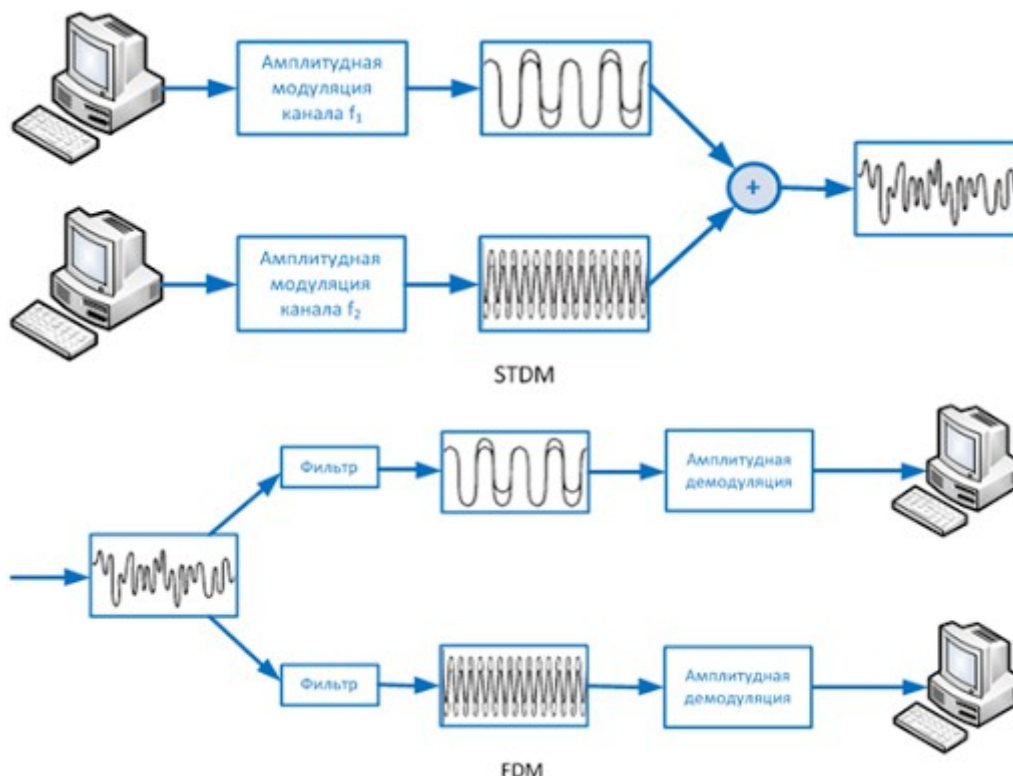
мультиплексирования, о которых вам следует знать:

Statistical time-division multiplexing (STDM)

- Одновременно передается несколько типов данных через один коммуникационный кабель или линию (например, T1 или T3), как показано ниже.
- STDM анализирует статистические данные, относящиеся к типичной рабочей нагрузке каждого устройства (принтер, факс, компьютер), и в режиме реального времени определяется, как много времени должно быть выделено каждому устройству для передачи данных.

Frequency-division multiplexing

- Для передачи данных используется имеющийся беспроводный спектр.
- Каждая частота в спектре используется в качестве канала для передачи данных.



Ссылки по теме:

- "All You Wanted to Know About T1 But Were Afraid to Ask," by Bob Wachtel, Data Comm for Business, Inc.

10.3. Технологии WAN

Технологии WAN применяются, чтобы позволить сетям взаимодействовать на больших расстояниях. Сегодня компаниям доступно несколько разновидностей технологий WAN. При выборе наиболее подходящей технологии WAN компании, как правило, нужно проанализировать информацию о функциональных возможностях, пропускной способности, требованиях соглашения об уровне обслуживания, необходимом оборудовании, стоимости и дополнительных возможностях поставщиков услуг. В следующих разделах мы рассмотрим некоторые из доступных на сегодняшний день технологий WAN.

CSU/DSU

CSU/DSU (Channel Service Unit/Data Service Unit – Устройство обслуживания канала/Устройство обслуживания данных) требуется в случае использования цифрового оборудования для подключения LAN к WAN. Это подключение может осуществляться по линиям T1 или T3, как показано на Рисунке 5-47. Поскольку сигналы и кадры могут

различаться в оборудовании LAN и в оборудовании WAN, провайдером услуг требуется использовать CSU/DSU.

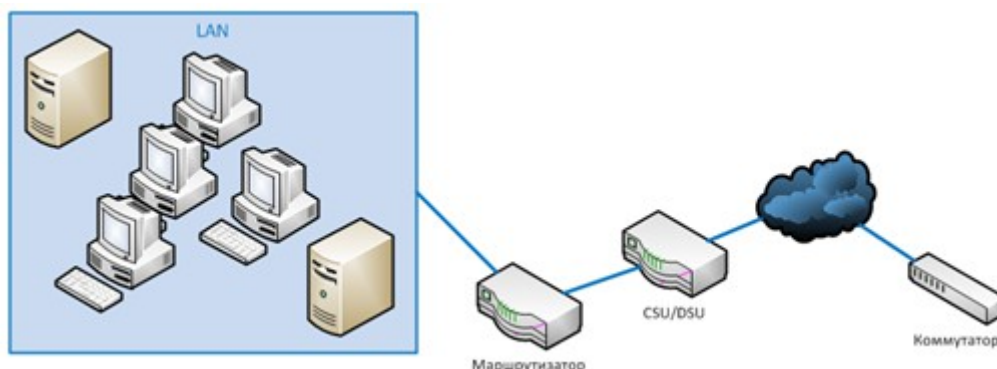


Рисунок 5-47. Устройство CSU/DSU требуется для взаимодействия цифрового оборудования с телекоммуникационными линиями

Устройство DSU преобразует цифровые сигналы от маршрутизаторов, мостов и мультиплексоров в сигналы, которые могут передаваться по цифровым линиям телефонной компании. Устройство DSU обеспечивает правильный уровень напряжения и отсутствие потерь информации при преобразовании сигналов. CSU подключает сеть напрямую к линии телефонной компании. CSU/DSU не обязательно является отдельным устройством, оно может быть реализовано в виде части сетевого устройства.

CSU/DSU предоставляет цифровой интерфейс для DTE (Data Terminal Equipment - Оконечное оборудование данных), например, терминалов, мультиплексоров или маршрутизаторов, а также интерфейс к устройствам DCE (Data Circuit-Terminating Equipment - Оконечное оборудование линии связи), например, коммутатору оператора. CSU/DSU работает в основном в качестве транслятора и, в отдельных случаях, как стабилизатор.

Коммутация

Выделенные каналы имеют единственный маршрут передачи, поэтому не возникает сложностей при определении способа отправки пакетов различным получателям. Между двумя сетями существует лишь две точки. Однако это становится намного сложнее, когда речь идет о тысячах связанных друг с другом сетей, возникает необходимость в коммутации (switching).

Могут использоваться два основных типа коммутации: коммутация каналов (circuit switching) и коммутация пакетов (packet switching). При **коммутации каналов** устанавливается виртуальное соединение, которое работает как выделенная линия между двумя системами. ISDN и телефонная связь являются примерами коммутации каналов, показанной в нижней половине Рисунка 5-48.

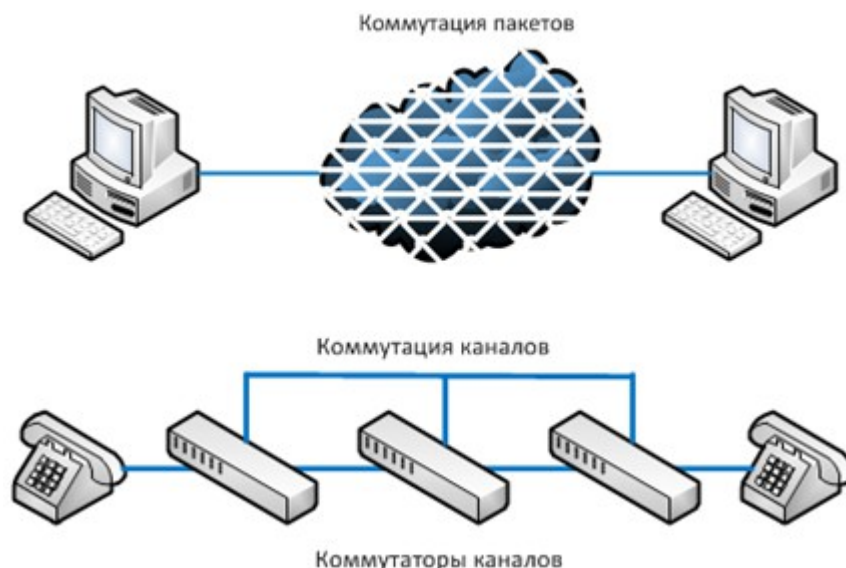


Рисунок 5-48. Коммутация каналов дает один коммуникационный маршрут, тогда как коммутация пакетов дает множество различных маршрутов

Когда система отправителя создает соединение с системой получателя, создается коммуникационный канал. Если две системы являются локальными по отношению друг к другу, требуется меньше устройств для создания этого канала. Чем дальше друг от друга находятся две эти системы, тем больше устройств должны быть вовлечены в создание канала и соединение двух систем.

В качестве примера системы с коммутацией каналов можно привести работу обычного телефона. Когда один человек звонит другому, устанавливается выделенное виртуальное коммуникационное соединение такого же типа. После того как соединение установлено, устройства поддерживают этот коммуникационный канал, чтобы он не переместился динамически на другие устройства. Канал остается настроенным на первоначальные устройства до момента окончания звонка или отключения связи.

Коммутация пакетов не создает выделенного виртуального соединения, пакеты одного соединения могут пройти через целый ряд различных отдельных устройств (см. верхнюю часть Рисунка 5-48), вместо того, чтобы следовать друг за другом через одно и то же устройство. В качестве примеров технологии коммутации пакетов, существующих в Интернете, можно указать X.25 и Frame Relay. Инфраструктура, которая поддерживает эти методы, создается маршрутизаторами и коммутаторами различных типов. Они предоставляют несколько путей к одному и тому же получателю, что обеспечивает высокую степень избыточности.

В сети с коммутацией пакетов данные разбиваются на пакеты, содержащие номер в последовательности для проверки кадра. Эти пакеты проходят через различные устройства, их пути могут быть динамически изменены маршрутизатором или коммутатором, определяющими наилучший маршрут для каждого конкретного пакета. После получения пакетов компьютером получателя, все пакеты пересобираются в соответствии с номерами последовательности, указанными в кадрах, и затем интерпретируются.

ПРИМЕЧАНИЕ. Коммутация пакетов основана на STDM, в которой анализируются статистические данные о различных возможных маршрутах для принятия решения о наилучшем маршруте для пакета.

Поскольку путь, по которому будет направлен пакет в среде с коммутацией пакетов, не установлен жестко, задержка может быть переменной в отличие от технологии с коммутацией каналов. Это не вызывает проблем, т.к. в сетях с коммутацией пакетов, как правило, передаются данные, а не голос. Для передачи голосовой информации больше подходят сети с коммутацией каналов, т.к. передача голосовой информации критична к

задержкам. Голосовые звонки, как правило, обеспечивают устойчивый поток информации, тогда как соединения, передающие данные, хаотичны по своему характеру. Когда вы говорите по телефону, сохраняется определенный ритм. Вы и ваш друг не говорите очень быстро, а затем останавливаетесь на несколько минут, создавая полную тишину. Однако, именно таким образом, как правило, работают соединения, передающие данные. Большие объемы данных передаются от одного конца до другого за один раз, а затем тишина продолжается до тех пор, пока не настанет время для передачи следующей порции данных.

ПРИМЕЧАНИЕ. Голос через IP (VoIP – Voice over IP) передает голосовые данные через среды с коммутацией пакетов. Эта технология описывается в разделе «Многофункциональные технологии доступа», далее в этом Домене.

Коммутация каналов vs. Коммутация пакетов. Ниже представлено краткое резюме по различиям между технологиями коммутации каналов и пакетов.

Коммутация каналов:

- Виртуальные каналы с предварительным установлением соединения
- Трафик передается предсказуемым и постоянным образом
- Фиксированные задержки
- Обычно применяется для передачи голосовых данных

Коммутация пакетов:

- Пакеты до одного и того же получателя могут передаваться по различным маршрутам, выбираемым динамически
- Трафик обычно хаотичен по своему характеру
- Переменная величина задержек
- Обычно применяется для передачи данных (не голосовых)

Frame Relay

Долгое время многие компании для взаимодействия с другими компаниями использовали выделенные линии. Компании А и В имели выделенный канал между собой, обеспечивающий определенную пропускную способность 24 часов в сутки, этот канал не использовался кем-либо другим. Это было прекрасно, поскольку лишь две компании могли использовать канал, поэтому определенный уровень пропускной способности был доступен всегда. Однако это было дорого, к тому же большинство компаний не использовало постоянно пропускную способность в полной мере. Фактически, компании тратили много денег на сервис, который они не использовали постоянно. В настоящее время вместо использования выделенных линий компании часто выбирают сети Frame Relay.

Frame Relay – это WAN-протокол, работающий на канальном уровне. Этот протокол использует технологию коммутации пакетов, позволяя нескольким компаниям и сетям разделять между собой общую среду WAN. Тогда как стоимость прямых соединений точка-точка зависит от расстояния между конечными точками, стоимость Frame Relay основана на величине используемой полосы пропускания. Поскольку несколько компаний и сетей используют одну и ту же среду и оборудование (маршрутизаторы и коммутаторы), затраты каждой компании в отдельности могут быть значительно сокращены по сравнению с выделенными линиями.

Если компания знает, что для ежедневной работы ей требуется полоса пропускания X, она заплатит определенную плату, чтобы постоянно иметь в своем распоряжении такую полосу пропускания. Если другая компания знает, что ей не нужна большая пропускная способность, она может платить более низкую плату, однако при этом у нее не будет гарантий большой пропускной способности. Эта вторая компания в любом случае будет иметь более высокую пропускную способность, чем она заказывала. Пропускная способность будет минимальна (равна той, которую она заказывала), когда канал занят, но

она будет увеличиваться по мере снижения загруженности канала. Компании, которые платят больше для того, чтобы гарантированно получить большую полосу пропускания, получают больший уровень **CIR** (Committed information rate – Гарантированная полоса пропускания).

В соединениях Frame Relay используются два основных типа оборудования: DTE (Data Terminal Equipment – Терминальное оборудование) и DCE (Data Circuit-Terminating Equipment – Оконечное оборудование передачи данных). Обычно DTE – это принадлежащее клиенту устройство (например, маршрутизатор или коммутатор), которое обеспечивает связь между собственной сетью компании и сетью Frame Relay. DCE – это устройство поставщика услуг или телекоммуникационной компании, которое осуществляет передачу данных и коммутацию в облаке Frame Relay. DTE является «трамплином» компании в сеть Frame Relay, а DCE выполняет работу в самом облаке Frame Relay.

Облако Frame Relay – это набор DCE, которые обеспечивают коммутацию и передачу данных. Такие услуги предоставляют многие провайдеры, некоторые из них используют для этого оборудование других провайдеров – все это может привести к сложностям, поскольку пакет может пройти огромным количеством различных маршрутов. Этот набор был назван *облаком* (cloud), чтобы отделить его от других типов сетей, а также из-за того, что обычно после попадания пакета в облако, пользователи не знают о маршруте, по которому он будет направлен. При этом кадры передаются через постоянные или коммутируемые виртуальные каналы, определенные в рамках DCE, либо через коммутаторы оператора.

Frame Relay – это сервис связи «любой-с-любым» (any-to-any), совместно используемый множеством пользователей. Как было отмечено ранее, это выгодно, т.к. при этом затраты значительно ниже, чем в случае использования выделенных линий. Поскольку сервис Frame Relay является общим, если один абонент не использует его пропускную способность, она становится доступной другим абонентам. С другой стороны, при увеличении объема трафика, свободная (доступная другим) полоса пропускания уменьшается. Поэтому абонентам, которые хотят на постоянной основе иметь в своем распоряжении определенную полосу пропускания, приходится платить больше тех, которым это не требуется.

Рисунок 5-49 показывает пять площадок (site), соединенных посредством выделенных линий, в сравнении с пятью площадками, соединенными через облако Frame Relay. Первый вариант требует большого количества выделенных линий, которые являются дорогостоящими и не гибкими. Второй вариант дешевле и при этом он дает компании гораздо больше гибкости.

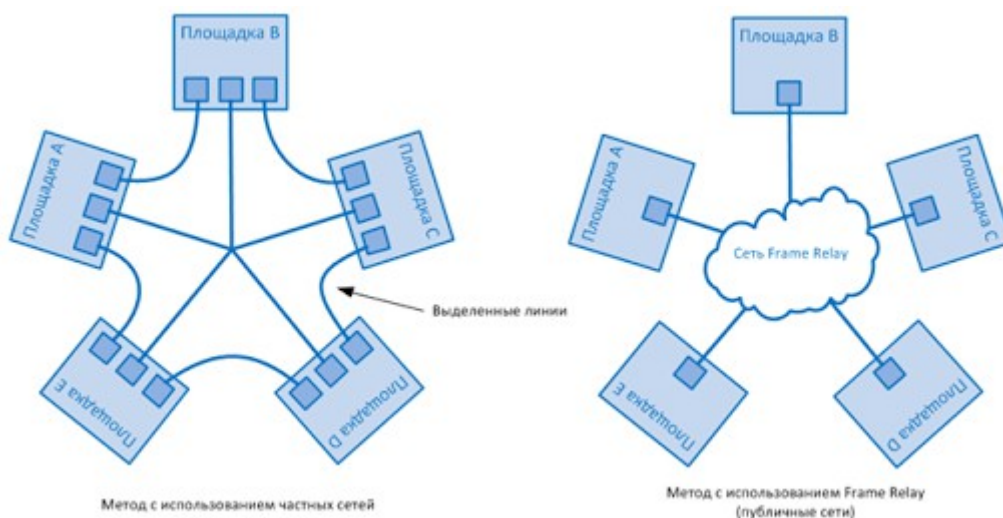


Рисунок 5-49. Соединения с использованием частной сети требуют множества дорогих выделенных линий; Frame Relay позволяет пользователям совместно использовать публичные сети

Виртуальные каналы

Frame Relay (и X.25) передает кадры через **виртуальные каналы** (virtual circuit). Эти каналы могут быть *постоянными* (permanent) (т.е. запрограммированными заранее) или *коммутируемыми* (switched) (т.е. быстро создаваемыми по мере необходимости и отключающимися, когда они больше не нужны). Работа **постоянного виртуального канала** (PVC – Permanent Virtual Circuit) похожа на работу выделенной персонально клиенту линии с согласованной доступной полосой пропускания. Когда клиент заключает договор на получение определенной скорости, PVC программируется персонально для этого клиента, чтобы обеспечить постоянное наличие для него определенной величины полосы пропускания.

В отличие от PVC, **коммутируемые виртуальные каналы** (SVC – Switched Virtual Circuit) требуют выполнения шагов, похожих на процедуры подключения по Dial-Up. Разница заключается в том, что для PVC создаются постоянные пути, а для SVC они должны создаваться каждый раз при возникновении необходимости. Это похоже на выполнение телефонного звонка через публичную телефонную сеть. При выполнении процедуры установки соединения запрашивается требуемая пропускная способность, создается соединение с компьютером получателя, который должен принять вызов, определяется путь, программируется пересылка информации на каждый коммутатор по всему пути SVC. SVC используются для проведения телеконференций, создания временных подключений к удаленным площадкам, выполнения репликации данных, а также для выполнения голосовых звонков. Когда соединение станет ненужным, оно разрывается и коммутаторы «забывают», что оно когда-то существовало.

PVC обеспечивает гарантированный уровень пропускной способности, однако он не имеет той гибкости, которой обладает SVC. Если клиенту нужно временное подключение по PVC, ему нужно связаться с провайдером, а затем дождаться, когда провайдер создаст его, что может занять несколько часов.

X.25

X.25 – это старый WAN-протокол, который определяет, как устройства и сети создают и поддерживают соединения. Как и Frame Relay, X.25 является технологией коммутации, которую используют коммутаторы оператора для предоставления соединений множеству различных сетей. Он также обеспечивает соединения типа «любой-с-любым», т.е. множество пользователей используют один и тот же сервис одновременно. Стоимость услуг для абонентов определяется на основе величины используемой ими полосы пропускания, в отличие от выделенных линий, в которых взимается постоянная плата.

Данные делятся по 128 байт и инкапсулируются в кадры HDLC (High-level Data Link Control). Затем кадры адресуются и пересылаются через коммутаторы оператора. Это похоже на Frame Relay, однако Frame Relay по сравнению с X.25 является гораздо более продвинутой и эффективной, поскольку протокол X.25 был разработан и выпущен в 1970-х годах. В то время большинство подключаемых к сети устройств являлись простыми терминалами и мейнфреймами, сети не обладали встроенной функциональностью и отказоустойчивостью, а Интернет в целом не был таким стабильным и устойчивым к ошибкам, как в наше время. X.25 был предназначен для компенсации этих недостатков и обеспечения нескольких уровней выявления и исправления ошибок, обеспечения отказоустойчивости. Это сделало протокол очень «толстым», что было необходимо в то время, однако сегодня это замедляет передачу данных и обеспечивает более низкую производительность, чем Frame Relay или ATM.

ATM

Асинхронный режим передачи (ATM – Asynchronous Transfer Mode) является еще одной технологией коммутации, но он вместо коммутации пакетов использует метод коммутации ячеек (cell-switching). ATM – это высокоскоростная сетевая технология, используемая для

LAN, MAN, WAN и соединений провайдеров услуг. Как и Frame Relay, это технология коммутации с предварительным установлением соединения, она создает и использует фиксированный канал. IP является примером технологии, не использующей предварительное установление соединения. В наборе протоколов TCP/IP, IP не производит предварительного установления соединения, а TCP – производит. Поэтому сегменты IP можно быстро и легко маршрутизировать и коммутировать, не беспокоясь о том, дошли ли фактически данные до получателя – это работа TCP. TCP работает на обеих сторонах (отправителя и получателя), гарантируя доставку данных получателю – в случае возникновения проблемы, не позволившей доставить данные получателю, эти данные отправляются повторно. При использовании ATM или Frame Relay, устройства между отправителем и получателем должны лучше понимать передаваемые данные и пункт их назначения, в отличие от использования протоколов, не устанавливающих соединения.

ATM – это технология коммутации ячеек, а не пакетов. Данные сегментированы в ячейках фиксированного размера (по 53 байта), вместо пакетов переменного размера. Это обеспечивает более эффективное и быстрое использование коммуникационных маршрутов. ATM устанавливает виртуальные каналы, которые работают подобно выделенным маршрутам между отправителем и получателем. Эти виртуальные каналы могут гарантировать определенную полосу пропускания и обеспечивать QoS (Quality of Service - Качество обслуживания). По этим причинам ATM является хорошим вариантом для передачи звука и видео.

Технология ATM используется операторами и провайдерами услуг, она является частью ключевых технологий Интернет. В то же время технология ATM может использоваться и отдельными компаниями в магистральных линиях и каналах связи с сетями провайдеров услуг.

Для подключения к сетям общего пользования компании традиционно используют выделенные линии (Т-линии). Однако, компании переходят к внедрению ATM-коммутаторов в своей сети для связи с инфраструктурой оператора. При этом используется расчет стоимости услуг на основе выделенной пропускной способности, что может быть значительно ниже платы за постоянный канал связи. Некоторые компании заменили свои магистрали Fast Ethernet и FDDI на ATM. Для использования ATM в качестве магистрали компании, устанавливают ATM-коммутаторы, которые принимают кадры Ethernet (или любых других используемых канальных технологий) и переводят их в 53-байтовые ячейки ATM.

Качество обслуживания

Качество обслуживания (QoS – Quality of Service) – это возможность, позволяющая протоколу проводить различие между разными классами сообщений и назначать уровни приоритета. Некоторые приложения, такие, как видеоконференцсвязь, чувствительны ко времени – т.е. задержки при передаче данных приведут к неприемлемой работе приложения. Технология QoS позволяет администратору устанавливать уровень приоритета для чувствительного ко времени трафика. Затем протокол обеспечивает для этого типа трафика определенную или минимально приемлемую скорость передачи.

QoS позволяет провайдеру услуг гарантировать уровень сервиса своим клиентам. Изначально QoS появился в ATM, а затем был интегрирован в другие технологии и протоколы, обеспечивающие перемещение данных из одного места в другое. Клиентам доступны четыре различных типа сервисов QoS в ATM (они перечислены ниже). Каждый сервис применяется к определенному типу передаваемых данных.

- **Постоянная скорость** (CBR – Constant Bit Rate). Канал с предварительным установлением соединения, посредством которого передаются чувствительные ко времени данные, такие как голосовая или видеосвязь. Клиенты при установлении

соединения указывают необходимую им полосу пропускания.

- **Переменная скорость** (VBR – Variable Bit Rate). Канал с предварительным установлением соединения, который лучше всего использовать для нечувствительных к задержкам приложений, поскольку поток данных является неравномерным. Клиенты при установлении соединения указывают необходимую им пиковую и установившуюся скорость передачи данных.
- **Неопределенная скорость** (UBR – Unspecified Bit Rate). Канал без предварительного установления соединения, который не обещает конкретных скоростей передачи данных. Клиент не может управлять скоростью потока трафика (и не нуждается в этом).
- **Доступная скорость** (ABR – Available Bit Rate). Канал с предварительным установлением соединения, позволяющий регулировать скорость. Клиенты получают полосу пропускания, которая остается после достижения гарантированной скорости.

АТМ поддерживает различные интерфейсы для обеспечения такой гибкости и использования этих возможностей. Устройство DSU применяется для подключения LAN к общедоступной АТМ-сети. Такая архитектура используется для случая, когда сеть компании подключена к сети АТМ посредством маршрутизатора или моста. Устройство DSU необходимо для обеспечения преобразования цифрового сигнала в сигнал, который понимает сеть АТМ.

АТМ создает фиксированный канал для передачи данных. Фиксированные каналы предварительно программируются в коммутаторах, установленных на маршруте, по которому передаются данные.

АТМ был первым протоколом, в котором был реализован настоящий QoS, но потребности компьютерного сообщества возросли и возникла необходимость отправлять чувствительные ко времени данные через множество различных типов сетей. Поэтому разработчики интегрировали QoS и в другие технологии.

QoS имеет три основных уровня (класса):

- **Низкоприоритетный сетевой трафик** (Best-effort service). Не гарантирован уровень пропускной способности и задержек, не гарантирована доставка. Трафик, для которого установлена классификация по приоритету, направляется перед трафиком, для которого установлен класс «низкоприоритетный сетевой трафик». Для большей части трафика, передаваемого по сети Интернет, устанавливается именно этот класс.
- **Дифференцированный сервис** (Differentiated service). По сравнению с низкоприоритетным трафиком, трафик, для которого установлен этот класс, имеет большую полосу пропускания, меньшие задержки и меньшее количество потерянных кадров.
- **Гарантированный сервис** (Guaranteed service). Обеспечивает передачу определенных данных на гарантированной скорости. Для чувствительного ко времени трафика (голос и видео) устанавливается этот класс.

Администраторы могут установить классификацию приоритетов (или использовать специализированный продукт для управления политикой) для различных типов трафика, который передается с помощью соответствующих протоколов и устройств.

SMDS

SMDS (Switched Multimegabit Data Service) – это высокоскоростная технология коммутации пакетов, которая применяется для обеспечения возможности клиентам расширить их сети LAN через сети WAN и MAN. Например, если компания имеет офис в одном штате и ей необходимо взаимодействовать с офисом в другом штате, можно обеспечить взаимодействие

двух ее сетей LAN через уже существующие сети общего пользования с помощью протокола SMDS. Этот протокол не производит предварительное установление соединения и может обеспечить требуемую полосу пропускания.

Хотя некоторые компании продолжают использовать эту технологию, она была в значительной степени заменена Frame Relay. Провайдеры услуг, как правило, только поддерживают сети, использующие эту технологию, для уже существующих клиентов, новым клиентам использование таких сетей не предлагается.

SDLC

Протокол **SDLC** (Synchronous Data Link Control) предназначен для сетей, которые используют выделенные, арендованные линии с постоянными физическими соединениями. Он используется в основном для связи с узлами IBM в рамках SNA (Systems Network Architecture - Системная сетевая архитектура). SDLC разработан IBM в 1970-е годы, он является бит-ориентированным синхронным протоколом, который используется внутри других коммуникационных протоколов, таких как HDLC, LAP (Link Access Procedure) и LAPB (Link Access Procedure-Balanced).

SDLC был разработан для обеспечения возможности взаимодействия мейнфреймов на большом расстоянии друг от друга. В средах, использующих SDLC, как правило, устанавливается первичная система, которая управляет коммуникациями вторичных станций. SDLC реализует технологию последовательного (polling) доступа к среде, позволяющую вторичным станциям взаимодействовать в сети. Рисунок 5-50 показывает, первичные и вторичные станции в сети SDLC.



Рисунок 5-50. SDLC используется в основном в среде мейнфреймов в сети SNA

HDLC

Протокол **HDLC** (High-level Data Link Control) также является бит-ориентированным протоколом канального уровня, который используется для передачи данных по синхронным линиям. HDLC является расширением SDLC, который используется в основном в среде SNA. HDLC обеспечивает высокую пропускную способность, потому что он поддерживает полнодуплексную передачу, а также использует двухточечные и многоточечные соединения.

Как и в случае SDLC, работа HDLC организована с использованием первичных станций, которые соединены с вторичными станциями для обеспечения передачи данных. Производители используют собственные параметры в своих версиях реализации HDLC, что привело к проблемам совместимости между реализациями HDLC различных производителей.

HSSI

HSSI (High-Speed Serial Interface) – это интерфейс, который используется для соединения мультимплексов и маршрутизаторов в высокоскоростных сервисах связи, таких как ATM и Frame Relay. Он поддерживает скорость до 52 Мбит/с, как в WAN-соединениях T3. Как правило, он реализуется в маршрутизаторах и мультиплексирующих устройствах для обеспечения последовательных интерфейсов в WAN.

Эти интерфейсы определяют электрические и физические параметры для использования DTE/DCE устройствами, поэтому HSSI работает на физическом уровне. Данный интерфейс был разработан Cisco и T3plus Networking.

Ссылки по теме:

- Internetworking Technology Handbook, Chapter 12, “High-Speed Serial Interface,” Cisco Systems, Inc.
- “High-Speed Serial Interface (HSSI) Design Specification,” by John T. Chapman (Cisco Systems, Inc.) and Mitri Halabi (T3plus Networking, Inc.)

Многофункциональные технологии доступа

Многофункциональные технологии доступа (Multiservice access technology) объединяют несколько категорий коммуникаций (данные, голос и видео) в одном коммуникационном канале. Это обеспечивает более высокую производительность, сокращает эксплуатационные расходы, повышает гибкость, обеспечивает интеграцию и упрощает управление для администраторов. Обычные телефонные системы, основанные на коммутируемых каналах и ориентированные на передачу голоса, называются **PSTN** (Public-switched telephone network - Телефонная сеть общего пользования). PSTN использует коммутацию каналов вместо коммутации пакетов. Когда совершается телефонный вызов, он передается на интерфейс PSTN, которым является телефон пользователя. Этот PSTN подключен к местной линии связи (петле) телефонной компании с помощью медных проводов. Когда сигналы этого вызова достигнут центрального офиса телефонной компании (конец местной линии связи), они перейдут в коммутируемые линии телефонной компании. С момента установления соединения между абонентами, в течение всего сеанса взаимодействия, данные будут проходить через одни и те же коммутаторы.

При выполнении телефонного вызова устанавливается соединение, передаваемые в рамках этого соединения сигналы должны контролироваться, а после завершения сеанса связи соединение должно быть разорвано. Это осуществляется посредством протокола SS7 (Signaling System 7). Если применяется **VoIP** (Voice over IP), он использует **SIP** (Session Initiation Protocol - Протокол установления сеанса), с помощью которого производится установление и разрыв сеансов связи аналогично тому, как это делает SS7 для не-IP телефонных вызовов. SIP является протоколом прикладного уровня, который может работать через TCP или UDP. SIP дает основу для реализации более сложных возможностей телефонных сетей, аналогичным тем, которые реализуются SS7, например, включение телефонного звонка, набор телефонного номера, воспроизведение сигналов «занято» и т. п.

PSTN заменяется сетями, ориентированными на передачу пакетных данных, которые могут содержать голос, видео и иные данные. Новые сети VoIP используют различные коммутаторы, протоколы и каналы связи. VoIP должен пройти через сложный переходный этап, в течение которого необходимо обеспечить взаимодействие старых систем и инфраструктуры с новыми системами до момента вывода из эксплуатации старых систем.

В технологиях VoIP применяется высококачественное сжатие, а идентификационными номерами (телефонными номерами) являются IP-адреса. Эта технология позволяет преодолеть некоторые барьеры, присутствующие сегодня в PSTN. Интерфейсные устройства (телефоны) имеют встроенные функции и логику, что усложняет реализацию различных видов сервисов, которые может поддерживать вся сеть в целом. При использовании VoIP, интерфейсом для доступа к сети может быть компьютер, сервер, офисная АТС или другое устройство, на котором работает телефонное приложение. Это предоставляет больше гибкости в процессе добавления новых сервисов, обеспечивает больше возможностей для управления взаимодействием устройств.

Поскольку это технология коммутации пакетов, возможны задержки при передаче данных.

Это проявляется в виде задержек при разговоре, незначительной рассинхронизации. Возникновение таких недостатков в процессе телефонного разговора с использованием VoIP может означать, что пакеты задерживаются в очереди отправки на устройстве отправителя или в процессе передачи. Это называется *колебаниями задержки* (jittering). В настоящее время разработаны протоколы, помогающие минимизировать эти проблемы и обеспечить более непрерывную работу телефонной связи.

ПРИМЕЧАНИЕ. Для передачи данных чувствительных к времени приложений, например, голосовая или видеосвязь, следует использовать изохронную (isochronous) сеть. Изохронная сеть использует протоколы и устройства, которые необходимы для обеспечения гарантированной непрерывной полосы пропускания.

Для работы VoIP необходимы четыре основных компонента: устройство IP-телефонии, менеджер обработки вызовов, система голосовой почты и голосовой шлюз. *Устройство IP-телефонии* (IP telephony device) – это просто телефон, на котором выполняется необходимое программное обеспечение, позволяющее ему работать в качестве сетевого устройства. В традиционных телефонных системах используется «умная сеть» и «простой телефон». При использовании VoIP, телефон должен быть «умным», на нем должно быть установлено необходимое программное обеспечение, позволяющее ему принимать аналоговые сигналы, оцифровывать их, разбивать их на пакеты, создавать необходимые заголовки и окончания пакетов, чтобы они смогли достичь своих получателей. *Система голосовой почты* (voicemail system) является местом для хранения сообщений, она предоставляет пользователю функции для поиска в каталоге и переадресации вызовов. *Голосовой шлюз* (voice gateway) выполняет маршрутизацию пакетов и обеспечивает доступ к унаследованным голосовым системам и резервное копирование передаваемой голосовой информации.

Когда пользователь совершает телефонный звонок, его «умный телефон», отправляет сообщение *менеджеру обработки вызовов*, указывая на необходимость выполнения вызова. Когда получатель телефонного вызова снимает трубку, это сообщает менеджеру обработки вызовов о том, что вызов был принят. Менеджер обработки вызовов уведомляет телефоны звонящего и принимающего звонок, что канал активен и голосовые данные отправляются туда и обратно через сеть.

Передавать голосовые данные с помощью пакетов сложнее, чем обычные данные. Это связано с тем, что данные, как правило, отправляются с интервалами, а голосовые данные должны передаваться в виде постоянного потока. Задержка при передаче данных не так заметна, как задержка при передаче голоса. Технология VoIP и поддерживающие ее протоколы имеют преимущество, обеспечивающее передачу голосовых данных с повышенной пропускной способностью, что снижает влияние таких проблем, как переменный характер задержек, циклические задержки, а также потери пакетов.

Если компания использует VoIP, она оплачивает и поддерживает только одну сеть, а не две отдельных выделенных сети (одну – для передачи данных, другую – для передачи голоса). Это экономит деньги и снижает нагрузку на администраторов, однако при этом возникают некоторые проблемы безопасности, которые следует понимать и учитывать.

ПРИМЕЧАНИЕ. Шлюз среды (Media Gateway) – это транслирующее устройство между отдельными телекоммуникационными сетями. Шлюз среды VoIP выполняет преобразование между голосом в TDM (Time Division Multiplexing) и VoIP.

Законный перехват. Для многих компаний обязательно выполнять законный перехват передаваемых голосовых данных. Это помогает правоохранительным органам отслеживать и контролировать подозрительную и криминальную деятельность. Законный перехват информации в сетях VoIP не так просто организовать, как в традиционных сетях PSTN. Существуют различные решения для выполнения этой задачи, но для качественной реализации необходимо изначально спроектировать сеть для выполнения эффективного мониторинга передаваемой голосовой информации. Сама система законного перехвата информации должна быть незаметна для пользователей, должна иметь удобный интерфейс для управления и эффективные механизмы безопасности.

ПРИМЕЧАНИЕ. Голосовые данные могут передаваться через IP, Frame Relay и ATM протоколы.

Шлюзы H.323

Рекомендации ITU-T (Международный союз электросвязи) охватывают широкий спектр мультимедийных услуг связи. H.323 также входит в эти рекомендации, он является стандартом для передачи видео и аудио пакетов данных в режиме реального времени. H.323 предназначен для передачи данных в условиях, когда множество пользователей могут быть вовлечены в обмен данными. Среда H.323 похожа на терминальную среду, терминалами в ней могут быть телефоны или компьютеры с программным обеспечением телефонии, шлюзы, которые соединяют эту среду с PSTN, многоточечные устройства управления и "привратники" (gatekeeper), управляющие вызовами и функциональностью.

Как и шлюзы других типов, шлюзы H.323 соединяют различные виды систем и устройств, предоставляя им необходимые функции трансляции. Терминалы H.323 подключены к этим шлюзам, которые, в свою очередь, могут быть подключены к PSTN. Эти шлюзы транслируют протоколы между телефонной сетью, основанной на каналах, и сетью VoIP, основанной на пакетах. Также шлюзы транслируют трафик, ориентированный на каналы в ориентированный на пакеты трафик и наоборот.

В настоящее время следует использовать шлюзы, которые позволят новой технологии взаимодействовать со старыми, однако в скором времени старые сети PSTN могут стать историей и все коммуникации смогут работать с пакетами вместо каналов.

Терминология. Термины «IP-телефония» и VoIP взаимозаменяемы:

- VoIP широко используется для предоставления современных сервисов: Caller ID, QoS, голосовая почта и т.д.
- Термин «IP телефония» является общим термином для всех приложений реального времени, работающих по IP, в т.ч. приложений для передачи голоса через сервисы мгновенных сообщений (IM – Instant Messaging) и видеоконференции.

Новые технологии предоставляют сервисы, которые способны на гораздо большее, чем просто передача голоса. Хотя мы рассматривали в основном VoIP, существуют и другие технологии, которые позволяют по одной сети передавать голос и данных, например, передача голоса через ATM (VoATM – Voice over ATM) и Frame Relay (VoFR – Voice over Frame Relay). ATM и Frame Relay являются протоколами с предварительным установлением соединения, а протокол IP – нет. Поэтому Frame Relay и ATM обеспечивают лучший QoS и меньший уровень колебаний задержки и латентности.

Чтобы взять лучшее из обоих миров, можно использовать комбинацию IP через ATM или Frame Relay. Это позволяет коммуникациям, ориентированным на пакеты, работать через сеть, ориентированную на соединения, предоставляя сквозное (end-to-end) соединение. IP находится на сетевом уровне и не зависит от среды – он может работать с различными протоколами и технологиями канального уровня.

Обычно у компании есть собственная офисная АТС, которая является коммутатором между компанией, PSTN и линиями T1 или T3, с помощью которых эта АТС подключена к центральному офису телефонной компании, где расположены коммутаторы, являющиеся дверью в PSTN. Если вместо доступа в PSTN через коммутаторы центрального офиса используются технологии WAN, данные передаются через облако Frame Relay или ATM. Пример такой конфигурации показан на Рисунке 5-51.

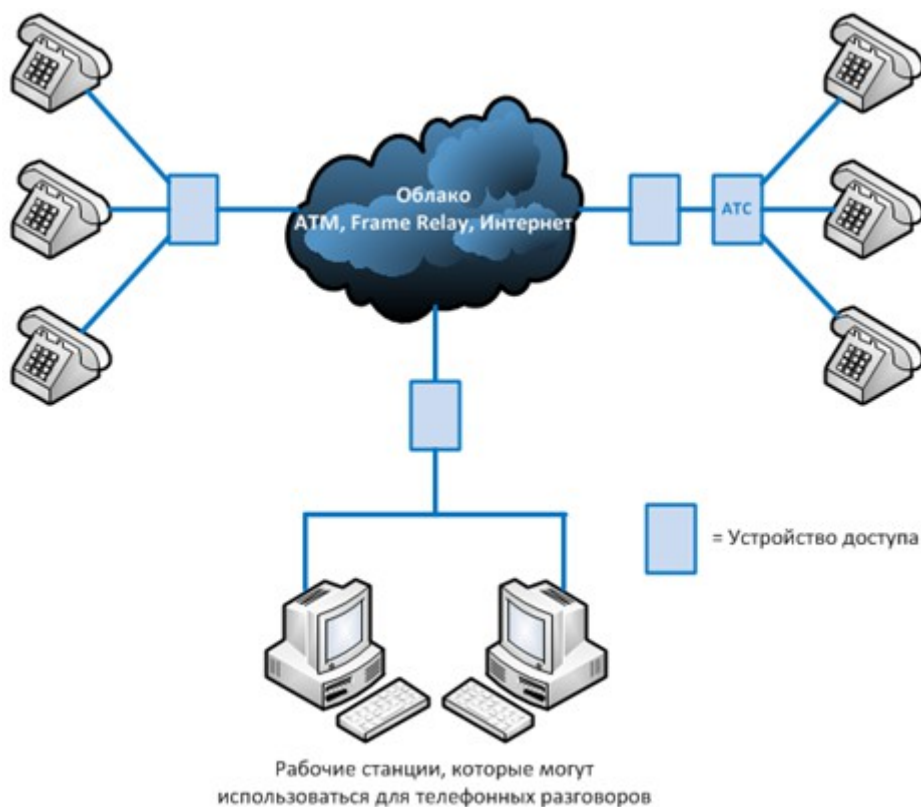


Рисунок 5-51. При выполнении обычных телефонных вызовов, телефон подключается к PSTN; при передаче голоса через технологии WAN – телефон подключается к WAN

Поскольку Frame Relay и ATM используют PVC и SVC, оба они имеют возможность использовать SVC для выполнения телефонных вызовов. Помните, что CIR используется в тех случаях, когда компании нужно быть уверенной, что она всегда будет иметь определенную доступную пропускную способность. Оплачивая эту гарантированную пропускную способность, компания платит за PVC, с целью программирования коммутаторов и маршрутизаторов для управления и поддержки соединений. С другой стороны, SVC создаются по требованию и носят временный характер. Они идеально подходят для осуществления телефонных вызовов или передачи видео в процессе видеоконференции.

Ссылки по теме:

- Voice over IP Quick Start Guide, Cisco Systems, Inc.
- IETF IP Telephony Charter

SIP

Как было сказано ранее, **SIP** (Session Initiation Protocol - Протокол установления сеанса) – это протокол обмена сигналами, широко используемый для сеансов коммуникаций VoIP. Он используется в таких приложениях, как видеоконференцсвязь, мультимедия, передача мгновенных сообщений, онлайн-игры. Это аналог протокола SS7, используемого в сетях PSTN, он поддерживает функциональность обычных телефонных систем.

SIP состоит из двух основных компонентов: UAC (User Agent Client) и UAS (User Agent Server). UAC – это приложение, которое создает запросы SIP для инициирования коммуникационного сеанса. UAC является обычным средством передачи сообщений и программным телефоном, который используется для выполнения телефонных звонков по VoIP. UAS – это сервер SIP, который обеспечивает выполнение всех операций по маршрутизации и передаче сигналов, необходимых для осуществления телефонных звонков по VoIP.

SIP использует трехсторонний процесс «рукопожатия» (three-way-handshake process) для инициализации сеанса. Чтобы лучше понять этот процесс, рассмотрим следующий пример. Два человека Билл и Джон пытаются пообщаться посредством VoIP-телефонов. Система Билла начинает с отправки пакета INVITE системе Джона. На данном этапе система Билла не знает, где находится Джон, поэтому пакет INVITE отправляется серверу SIP, который ищет адрес Джона на *сервере-регистраторе* SIP. Когда местоположение системы Джона определено, пакет INVITE пересылается Джону. В течение всего этого процесса сервер поддерживает соединение со звонящим (Биллом), отправляя ему пакет TRYING, который говорит ему о том, что процесс выполняется. Как только пакет INVITE достигнет системы Джона, программное обеспечение на ней включит телефонный звонок. Пока система Джона звонит и ожидает, когда Джон ответит на звонок, она отправляет системе Билла пакет RINGING, говоря ему, что пакет INVITE был получен системой Джона, и сейчас система Джона ожидает, когда Джон примет звонок. Как только Джон ответит на звонок, системе Билла (через сервер) будет отправлен пакет OK. Система Билла отвечает на этот пакет пакетом ACK для начала процедуры настройки соединения. Здесь важно отметить, что сам по себе SIP не используется для передачи голосового потока, поскольку это просто сигнальный протокол. Реальный голосовой поток передается с помощью специальных протоколов, таких как **RTP** (Real-time Transport Protocol). Когда Билл и Джон закончат разговор, система, разрывающая соединение, направляет другой системе сообщение BYE. Другая система отвечает ей пакетом OK, подтверждая, что сеанс завершен. Рассмотренный процесс «рукопожатия» показан на Рисунке 5-52.

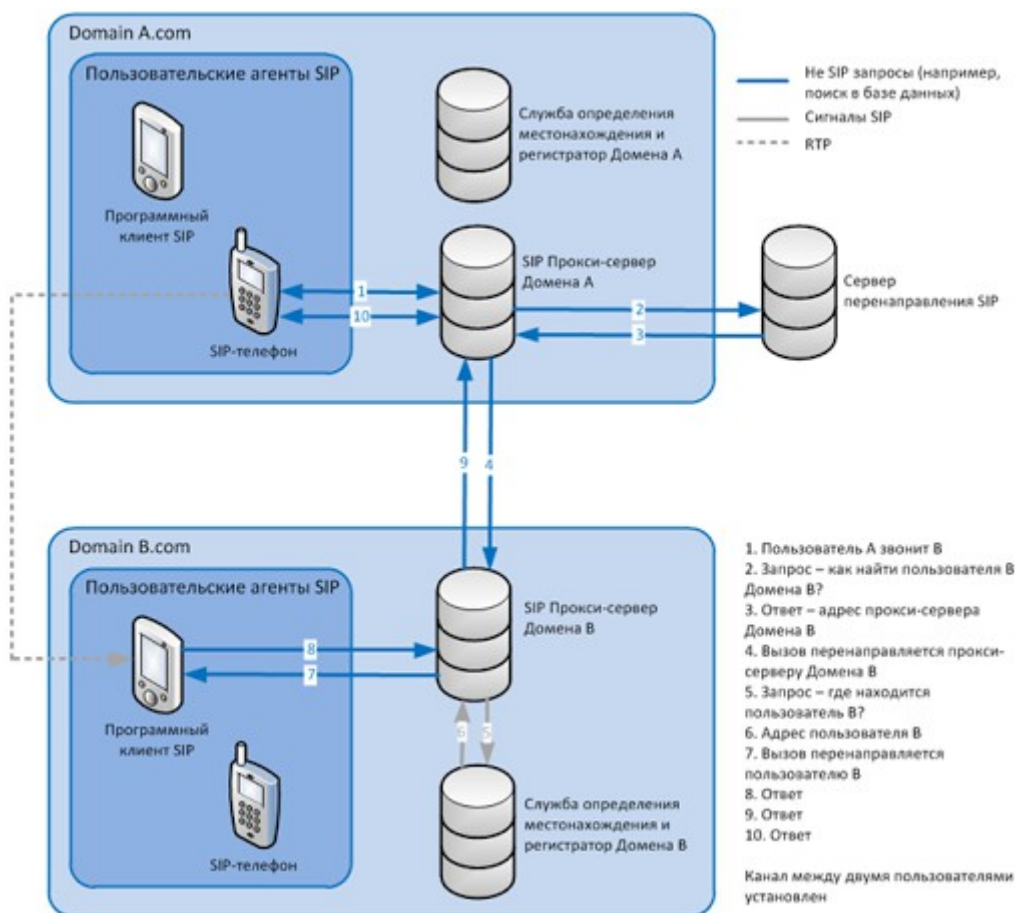


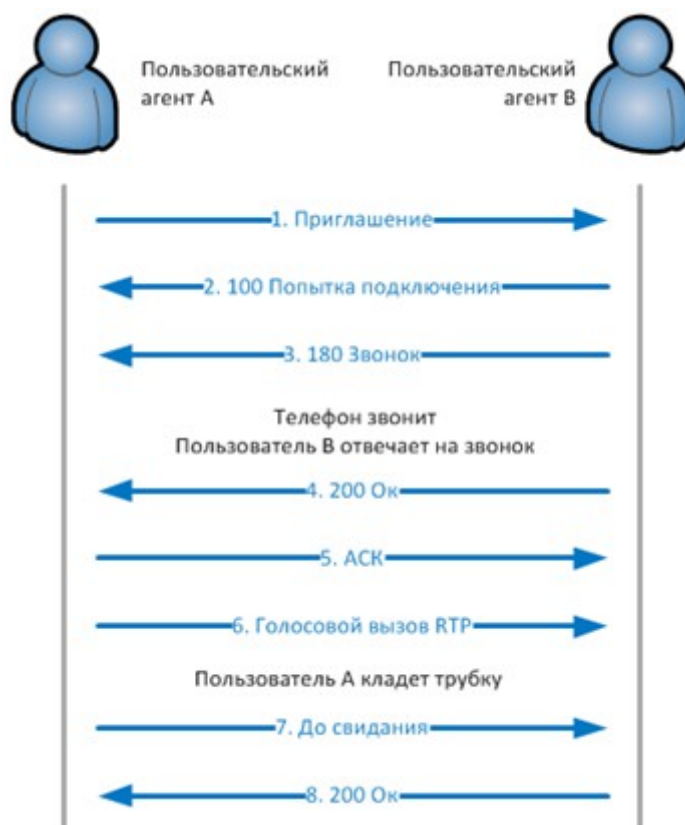
Рисунок 5-52. «Рукопожатие» SIP

Архитектура SIP состоит из трех различных типов серверов, которые совместно выполняют свои функции в процессе коммуникаций с использованием VoIP. Этими серверами являются: *прокси-сервер*, *сервер-регистратор* (registrar) и *сервер перенаправления* (redirect). Прокси-сервер используется для пересылки пакетов в сети между UAC и UAS. Также он пересылает

запросы звонящего на систему вызываемого абонента. Кроме того, прокси-серверы обычно используются для отображения имен, что позволяет им соединять внешние системы SIP с внутренними клиентами SIP.

Сервер-регистратор обеспечивает централизованное хранение записей актуального местоположения всех пользователей сети. Их адреса хранятся на сервере местоположения (location server). Сервер перенаправления позволяет устройствам SIP сохранять за собой свои SIP-идентификаторы, независимо от изменений их географического положения. Это позволяет устройствам оставаться доступными при физической смене своего местоположения и при перемещении между различными сетями. Это называется *интраорганизационной* конфигурацией (intraorganizational configuration).

Интраорганизационная конфигурация позволяет маршрутизировать трафик SIP в сети VoIP без его передачи через PSTN или внешнюю сеть, обеспечивая тем самым защиту коммуникационных сеансов.



Skype является популярной программой Интернет-телефонии, которая использует модель пиринговой (peer-to-peer) связи вместо традиционной для VoIP систем модели клиент/сервер. Сеть Skype не использует централизованные серверы для хранения каталогов ее пользователей. Вместо этого записи о пользователях поддерживаются всей распределенной системой узлов, входящих в ее состав. Поэтому сеть может быстро адаптироваться к резким скачкам активности пользователей без использования дорогой централизованной инфраструктуры и вычислительных ресурсов.

Проблемы IP-телефонии

Интеграция VoIP с протоколом TCP/IP ведет к большим сложностям с обеспечением безопасности, поскольку это позволяет злоумышленникам использовать свой опыт атак на TCP/IP для этой, относительно новой, платформы. Злоумышленники могут воспользоваться уязвимостями, как архитектуры, так и самих систем VoIP. При защите систем VoIP следует учитывать такие традиционные проблемы безопасности, как несанкционированный доступ, эксплуатация уязвимостей коммуникационных протоколов, распространение вредоносного программного кода. Даже основной причиной атак на VoIP по-прежнему является желание

хакеров получить скандальную известность. Кроме того, для атакующих очень привлекательна возможность получить возможность бесплатных телефонных звонков. Одним словом, сети VoIP телефонии стоят перед лицом тех же традиционных уязвимостей, что и обычные компьютерные сети.

Более того, устройства VoIP используют архитектуру, похожую на архитектуру обычных компьютеров – у них так же есть операционная система, они осуществляют взаимодействие через протоколы Интернет, они предоставляют комбинацию сервисов и приложений.

Передача сигналов SIP страдает из-за отсутствия шифрования и аутентификации при передаче управляющих сигналов. Атакующий может подключиться к каналу связи между сервером и клиентом SIP и выполнять перехват сетевых пакетов с помощью сниффера, что позволит ему получить идентификаторы пользователей, пароли (PIN-коды), номера телефонов. Получив такую информацию, атакующий легко может воспользоваться ей для выполнения телефонных звонков за чужой счет. Мошенничество – это наиболее серьезная угроза, перед лицом которой стоят VoIP сети. При реализации VoIP сети необходимо убедиться, что шлюзы между VoIP и PSTN надежно защищены от вторжений, что позволит минимизировать риски мошенничества.

Также атакующие могут скрыть свою реальную идентификационную информацию путем перенаправления управляющих пакетов SIP от вызывающего абонента фальшивому получателю, чтобы ввести в заблуждение вызывающего абонента, который будет взаимодействовать с иной системой. Как и любая сетевая система, устройства VoIP уязвимы к DoS-атакам. Подобно флудингу серверов TCP с помощью пакетов SYN в сетях TCP/IP, атакующие могут выполнять флудинг серверов RTP, направляя им большое количество запросов на выполнение вызова, превышающее их возможности обработки. Кроме того, атакующие знают о возможности подключения ноутбуков, эмулирующих IP-телефоны, к используемым IP-телефонами интерфейсам Ethernet. Они могут использоваться для получения несанкционированного доступа или проведения DoS-атак. А если атакующий сможет перехватывать голосовые пакеты, он сможет таким образом подслушивать телефонные разговоры в этой сети. Атакующий может перехватывать пакеты RTP, содержащие потоки голосовых или видео-данных сеанса связи, и вставлять в них иные аудио/видео данные, вызывая раздражение реальных пользователей.

Также атакующий может установить в сети фальшивый сервер и направлять от его имени клиентам VoIP такие команды, как BYE, CHECKSYNC, RESET. Команда BYE, например, заставит устройство VoIP завершить текущее соединение, команда CHECKSYNC может быть использована для перезагрузки терминала VoIP, а по команде RESET сервер сбрасывает и повторно устанавливает соединение, что занимает довольно значительное время.

ПРИМЕЧАНИЕ. Некоторое время назад появился новый вариант спама, для рассылки которого используются сети VoIP. Этот спам называют **SPIT** (Spam over Internet Telephony – Спам посредством Интернет-телефонии). SPIT приводит к значительным потерям полосы пропускания и к пустой трате времени пользователями атакованной сети. Поскольку SPIT не может быть также просто удален, как обычный спам, жертва вынуждена прослушать все сообщение до конца. SPIT также может стать причиной перегрузки серверов голосовой почты.

Борьба с угрозами безопасности VoIP требует хорошо продуманного плана реализации инфраструктуры. Аналогично традиционным сетям, в сетях VoIP ключевое значение играет соблюдение баланса между безопасностью и свободным прохождением потока трафика. Важным шагом, направленным на снижение уровня возможных угроз злоумышленных и несанкционированных действий в сети, является внедрение механизмов авторизации. Авторизация отдельных IP-терминалов гарантирует, что только предварительно включенные в список устройства смогут получать доступ к сети. Хотя этот метод и не обеспечивает стопроцентной защиты, он является первым уровнем защиты от возможного несанкционированного подключения устройств злоумышленником и переполнения сети

вредоносными пакетами. В дополнение к этой мере, целесообразно организовать выполнение аутентификации устройств VoIP при попытке соединения друг с другом. Идентификация устройств при этом может выполняться на основе постоянных аппаратных идентификационных параметров, таких как MAC-адреса, либо на основе программных кодов, которые могут быть присвоены серверам (например, с помощью методов многоуровневого шифрования).

Использование безопасных криптографических протоколов, таких как TLS (Transport Layer Security), гарантирует, что все пакеты SIP будут перемещены внутри зашифрованного безопасного туннеля. Использование TLS может предоставить безопасный канал для клиент-серверных коммуникаций VoIP и предотвратить возможный перехват и подделку пакетов.

Защитные меры для обеспечения безопасности VoIP. Хакеры могут перехватывать входящие и исходящие вызовы, проводить DoS-атаки, выполнять ложные телефонные звонки, а также подслушивать конфиденциальные разговоры. Большинство контрмер для противодействия этим атакам аналогичны контрмерам для традиционных сетей, ориентированных на передачу обычных данных:

- Своевременно устанавливайте патчи на каждое сетевое устройство, связанное с передачей VoIP:
 - Сервер менеджера вызовов
 - Сервер голосовой почты
 - Сервер-шлюз
- Отслеживайте появление неизвестных или несанкционированно подключенных телефонных устройств
 - Обеспечьте выполнение аутентификации, позволяющей работать в сети только авторизованным телефонным устройствам
- Установите и поддерживайте:
 - Межсетевые экраны с контролем состояния
 - VPN для передачи критичных голосовых данных
 - Средства выявления вторжений
- Заблокируйте неиспользуемые порты на маршрутизаторах, коммутаторах, компьютерах и IP-телефонах
- Выполняйте мониторинг в режиме реального времени для обнаружения атак, туннелирования и подозрительных образцов (поведения) посредством систем IDS/IPS
- Выполняйте мониторинг содержимого передаваемых данных
- Используйте шифрование при передаче данных (голос, факс, видео) через недоверенные сети
 - Используйте двухфакторную аутентификацию
 - Ограничьте количество одновременных вызовов с помощью шлюза среды
 - Закрывайте сеансы связи после их завершения

Резюме по технологиям WAN

В предыдущих разделах мы рассмотрели несколько WAN технологий. В Таблице 5-11 показаны наиболее важные характеристики каждой из них.

Технология WAN	Характеристики
Выделенная линия	<ul style="list-style-type: none"> - Выделенные (арендуемые) линии соединяют два различных места - Дороже других технологий WAN - Безопасная, поскольку только два места обладают доступом к каналу связи
Frame Relay	<ul style="list-style-type: none"> - Высокопроизводительный WAN-протокол, использующий технологию коммутации пакетов и работающий через публичные сети - Общая среда, используемая различными компаниями - Использует SVC и PVC - Стоимость основана на используемой полосе пропускания
X.25	<ul style="list-style-type: none"> - Первая технология коммутации пакетов, разработанная для работы через публичные сети - Общая среда, используемая различными компаниями - Обладает более низкой скоростью, чем Frame Relay, поскольку имеет высокие перегрузки - Является международным стандартом и используется во многих странах, не только в США - Использует SVC и PVC
SMDS	<ul style="list-style-type: none"> - Высокоскоростная технология коммутации, используемая поверх публичных сетей
ATM	<ul style="list-style-type: none"> - Технология коммутации и мультиплексирования с высокой пропускной способностью, имеющая низкие задержки - Использует ячейки фиксированного размера, равного 53 байтам - Очень быстрая, за счет низких перегрузок
SDLC	<ul style="list-style-type: none"> - Позволяет мейнфреймам взаимодействовать с удаленными офисами - Обеспечивает механизм опроса для предоставления возможности взаимодействия первичным и вторичным станциям
HDLC	<ul style="list-style-type: none"> - Новый и более совершенный протокол SDLC - Метод инкапсуляции данных для синхронных последовательных соединений - «Точка-точка» и многоточечные коммуникации
HSSI	<ul style="list-style-type: none"> - Интерфейс DTE/DCE для обеспечения высокоскоростных коммуникаций через соединения WAN
VoIP	<ul style="list-style-type: none"> - Объединяет голос и данные в рамках одной IP сетевой среды и протокола - Имеет меньшую стоимость внедрения и поддержки по сравнению с двумя различными сетями

Таблица 5-11. Характеристики технологий WAN

11. Удаленный доступ

Удаленный доступ включает в себя ряд технологий, которые позволяют удаленным и домашним пользователям подключаться к сети для получения доступа к сетевым ресурсам, необходимым им для выполнения своих задач. В большинстве случаев, подключение к удаленной сети производится через Интернет, поэтому эти пользователи должны сначала получить доступ в Интернет через своего местного провайдера.

Многим компаниям необходим удаленный доступ к ресурсам своей сети, поскольку это дает возможность их сотрудникам оперативно получать актуальную информацию, снижает расходы на создание и содержание сетей (поскольку для доступа используется сеть Интернет, а не дорогостоящие выделенные линии), кроме того это позволяет сотрудникам работать из дома или в дороге. Удаленный доступ может позволить использовать ресурсы и информацию посредством Интернет-соединения, что дает конкурентные преимущества, предоставляя партнерам, поставщикам и клиентам удобные соединения. Наиболее распространены такие методы удаленного подключения, как VPN, соединения Dial-Up, ISDN, кабельные модемы, а также соединения DSL.

11.1. Dial-Up и RAS

Удаленный доступ обычно организуется посредством подключения к серверу удаленного доступа (RAS – Remote Access Server), который выступает в качестве шлюза и может быть конечной точкой сеанса PPP. Пользователи звонят на сервер RAS, который выполняет их аутентификацию, сопоставляя предоставленные ему пользователем учетные данные с хранящейся на нем базой учетных данных. Для аутентификации при удаленных подключениях обычно используется RADIUS, который рассматривается в Домене 02. Администратор может настроить несколько конфигураций, выполняющих различные действия при подключении удаленных пользователей к RAS. Обычно при подключении у

пользователя запрашивается ввод имени и пароля, после чего сервер RAS перезванивает ему по заранее указанному телефонному номеру. Эта защитная мера направлена на обеспечение того, что только уполномоченные пользователи получают доступ к сети компании. Кроме того, эта мера позволяет перевести на компанию расходы по оплате телефонной связи. При этом, даже если злоумышленнику удастся получить действующие учетные данные для удаленного доступа, он вряд ли сможет ответить на звонок по заранее указанному телефонному номеру. Однако эта мера безопасности может быть скомпрометирована, если кто-то установит переадресацию вызовов.

Когда атакующие пытаются найти точку возможного входа в сеть, их целью часто являются системы удаленного доступа. Они знают, что многие компании используют модемный пул (или иной вариант точки доступа) для обеспечения возможности удаленного входа в сеть, при этом некоторые компании воображают, что их межсетевой экран, контролирующий интернет-трафик, каким-то магическим образом защищает удаленный доступ к их сети. Если компания не внедрила надлежащие средства контроля доступа к RAS, злоумышленники смогут без труда перемещаться по внутренней сети компании, не беспокоя ее межсетевой экран.

Часто атакующие используют методы **сканирования модемов** (wardialing) для выявления модемов удаленного доступа. Специальные программные средства последовательно набирают телефонные номера из заранее подготовленного атакующим списка, выявляя номера, с которых отвечают модемы. Эти программы журналируют информацию об успешных соединениях (обнаруженных модемах) и пытаются идентифицировать систему на другом конце телефонной линии. Некоторые из этих программ имеют также возможности для проведения атаки по словарю, в котором злоумышленники указывают часто используемые комбинации имен и паролей, в надежде получить доступ к сети. Сканирование модемов позволяет злоумышленнику обнаружить все модемы, предоставляющие удаленный доступ к сети. Далее злоумышленник пытается воспользоваться каждым обнаруженным модемом для получения доступа к сети. Незащищенные модемы (или любые другие технологии удаленного доступа) в инфраструктуре компании, как правило, значительно проще скомпрометировать, чем межсетевой экран.

Также в компаниях часто подключают модемы непосредственно к рабочим станциям. Сотрудники могут подключить их самостоятельно, либо ИТ-персонал может подключить их, а затем забыть об этом. Поэтому для компании важно самостоятельно периодически проводить сканирование модемов по своим телефонным номерам, чтобы убедиться в отсутствии несанкционированных модемов. Некоторые телефонные АТС имеют возможности для обнаружения сигналов модемов на аналоговой телефонной линии и ведения аудита/записи их использования. Эта функция может помочь в реализации соответствующего требования политики безопасности, запрещающего использование несанкционированно подключенных к сети (в т.ч. телефонной) устройств.

С точки зрения безопасности удаленного доступа, наиболее важны три правила:

- Все пользователи должны быть полностью аутентифицированы на оборудовании удаленного доступа перед началом его использования.
- Не должно допускаться использования скрытого или ненадлежащего доступа к коммуникационным каналам.
- После прохождения аутентификации, пользователи должны получать доступ только к тем сервисам, использование которых им разрешено, и ни к чему более.

11.2. ISDN

ISDN (Integrated Services Digital Network - Цифровая сеть с интегрированными услугами) – это коммуникационный протокол, предоставляемый телефонными компаниями и

провайдерами услуг. Этот протокол вместе с необходимым оборудованием позволяет передавать данные, голос и другие виды трафика в цифровом виде по каналам связи, которые ранее использовались только для аналоговой передачи голоса. Телефонные компании много лет назад перешли на цифровые технологии. Исключения составляют только местные линии связи, реализованные на базе медных проводов, которые соединят дома и здания различных организаций с центральным офисом провайдера услуг связи. В этих офисах установлено коммутационное оборудование, которое производит аналого-цифровое преобразование сигналов. Однако местные линии связи всегда аналоговые и, соответственно, более медленные. Изначально ISDN был разработан для замены устаревших аналоговых телефонных систем, но он до сих пор не достиг ожидаемого уровня распространения.

ISDN использует те же провода и среду передачи данных, что и аналоговые технологии Dial-Up, но он работает в цифровом виде. При использовании модема для связи с провайдером, модем преобразует данные из цифрового вида в аналоговый, для передачи через телефонную линию. Если компьютер имеет необходимое оборудование и настройки для работы ISDN, преобразовывать данные из цифрового вида в аналоговый не требуется, данные передаются в цифровом виде. Безусловно, при этом на другом конце требуется использование аналогичного оборудования для получения и правильной интерпретации такого вида коммуникаций. Передача информации в цифровом виде обеспечивает более высокую скорость и экономичность.

ISDN – это набор телекоммуникационных услуг, которые могут использоваться через общедоступные и частные телекоммуникационные сети. Он предоставляет цифровую среду «точка-точка» с коммутацией каналов и создает соединение между двумя взаимодействующими устройствами. Соединение ISDN может быть установлено по любому каналу, который может использоваться для работы модемов, при этом ISDN обеспечивает более широкую функциональность и высокую пропускную способность. Этот цифровой коммутируемый сервис может предоставлять полосу пропускания по требованию и может использоваться для соединения локальных сетей между собой без использования дорогих выделенных линий.

Как было отмечено ранее, аналоговые технологии используют для связи всю ширину канала, а ISDN может разбить этот канал на несколько подканалов для предоставления полнодуплексной связи, более высокого уровня контроля и обработки ошибок. ISDN предоставляет два основных домашних и корпоративных сервиса: **BRI** (Basic Rate Interface - Интерфейс базового уровня) и **PRI** (Primary Rate Interface - Интерфейс первичного уровня). BRI имеет два В-канала, которые позволяют передавать данные, и один D-канал, используемый для создания и управления соединением, контроля ошибок, идентификации вызывающего абонента и многого другого. Пропускная способность BRI позволяет достичь скорости 144 Кбит/с, тогда как скорость модема может достигать лишь 56 Кбит/с.

Канал D обеспечивает более быстрое создание и настройку соединения. На настройку соединения ISDN требуется лишь 2-5 секунд времени, тогда как при использовании модема это может занять до 45-90 секунд. Канал D – это внеполосное соединение между оборудованием местной линии связи и пользовательским терминалом. Это соединение является внеполосным, поскольку управляющие данные не смешиваются с пользовательскими данными. Это усложняет для злоумышленников задачу отправки оборудованию провайдера поддельных команд с целью проведения DoS-атаки, бесплатного получения услуги или выполнения иных деструктивных действий.

Служба BRI больше подходит для частного использования, а PRI, которая имеет 23 В канала и один D канал, более широко используются в организациях. ISDN, как правило, не является основным каналом связи для компаний, но он может быть использован в качестве резервного на случай неработоспособности основного канала. Также компания может выбрать его для

реализации DDR-маршрутизации (Dial-on-demand routing - маршрутизация при предоставлении канала по требованию), которая может работать через ISDN. DDR позволяет компании отправлять данные WAN по существующим телефонным линиям и использовать публичные сети с коммутацией каналов в качестве временного решения для организации соединений WAN. Как правило, такой вариант используется компаниями, которым требуется отправлять лишь небольшой объем трафика WAN, при этом такой вариант гораздо дешевле полноценной реализации WAN. Соединение активизируется, когда оно необходимо, а после использования оно переходит в режим бездействия.

Реализации ISDN. ISDN разбивает телефонную линию на отдельные подканалы и передает данные в цифровом виде, а не в аналоговом. Используются три реализации ISDN:

- **BRI ISDN.** Эта реализация использует существующие медные каналы местных линий связи, она предоставляет каналы для передачи оцифрованного голоса и данных. Она использует два В-канала и один D-канал с общей пропускной способностью 144 Кбит/с. Как правило, эта реализация используется домашними пользователями.
- **PRI ISDN.** Эта реализация имеет до 23 В-канала и 1 D-канал, каждый канал имеет скорость 64 Кбит/с. Общая пропускная способность эквивалентна T1 и составляет 1,544 Мбит/с. Это более приемлемо для компаний, поскольку им необходима более широкая полоса пропускания.
- **Широкополосный ISDN (BISDN – Broadband ISDN).** Эта реализация поддерживает одновременную работу нескольких различных служб, в основном она используется в телекоммуникационных магистралях оператора. При использовании BISDN в магистрали, технология ATM применяется для инкапсуляции данных на канальном уровне в ячейки, которые передаются через сеть SONET.

11.3. DSL

DSL (Digital Subscriber Line - Цифровая абонентская линия) является еще одним типом высокоскоростных технологий связи, она применяется для подключения домов или офисов компаний к центральному офису провайдера услуг. DSL может обеспечить скорость в 6-30 раз выше скорости ISDN и аналоговых технологий. DSL использует существующие телефонные линии, обеспечивая при этом возможность круглосуточного непрерывного подключения к Интернету. Это действительно звучит заманчиво, однако не все могут получить эту услугу, т.к. для ее использования нужно находиться не далее 4 километров от оборудования провайдера услуг DSL. При увеличении расстояния скорость передачи данных по DSL снижается.

DSL является технологией широкополосной связи, он может обеспечить скорость передачи до 52 Мбит/с без замены медного канала связи. Для использования DSL, оборудование конечного пользователя и оператора должно быть обновлено, именно поэтому многие компании, и жители не могут воспользоваться этой услугой в настоящее время. Пользователь и оператор должны иметь DSL-модемы, использующие одинаковый метод модуляции.

DSL обеспечивает более высокую скорость передачи данных за счет использования всех доступных частот канала на основе витой пары (UTP) голосового класса. Когда вы кому-то звоните, по этому каналу передаются данные вашего голоса, а провайдер услуг телефонной связи «очищает» передачу, удаляя высокие и низкие частоты. Человек при разговоре не использует такие частоты, поэтому все, что находится на этих частотах считается шумом и удаляется. Фактически, таким образом сокращается имеющаяся пропускная способность линии связи от дома до центрального офиса телефонной компании. При использовании DSL этого не происходит, поэтому высокие и низкие частоты могут быть использованы для передачи данных.

DSL предлагает несколько видов сервисов. При использовании симметричных сервисов, потоки трафика имеют одинаковую скорость передачи в обе стороны (в Интернет и из Интернета). При использовании асимметричных сервисов, скорость загрузки намного выше,

чем скорость отдачи. В большинстве случаев, асимметричное подключение лучше подходит для домашних пользователей, поскольку они обычно скачивают из Интернета значительно больше, чем выгружают в Интернет.

11.4. Кабельные модемы

Компании, оказывающие услуги кабельного телевидения, начали дополнительно предлагать своим клиентам услуги передачи данных. При этом клиенту для высокоскоростного подключения к сети Интернет требуется кабельный модем.

Кабельные модемы (cable modem) предоставляют высокоскоростной (до 50 Мбит/с) доступ к сети Интернет, используя для этого существующие коаксиальные и оптоволоконные кабели. Кабельный модем обеспечивает передачу данных в обоих направлениях.

Не все компании, предоставляющие услуги кабельного телевидения, также предлагают доступ в Интернет. В основном это связано с тем, что они не обновили свою инфраструктуру для перехода от однонаправленной к двунаправленной сети, что необходимо для использования сети Интернет.

xDSL. Существует множество различных вариантов DSL, каждый из которых имеет свои собственные характеристики и подходит для конкретных вариантов использования.

- **Симметричный DSL** (SDSL – Symmetrical DSL). Данные передаются в обоих направлениях с одинаковой скоростью. Полоса пропускания может варьироваться от 192 Кбит/с до 1,1 Мбит/с. Применяется, в основном, в компаниях, требующих высокой скорости передачи данных в обоих направлениях.
- **Асимметричный DSL** (ADSL – Asymmetrical DSL). Загрузка данных из сети Интернет производится быстрее, чем отправка в сеть Интернет. Скорость отправки варьируется от 128 Кбит/с до 384 Кбит/с, а скорость загрузки может достигать 768 Кбит/с. Обычно используются домашними пользователями.
- **ISDN DSL** (IDSL). Позволяет использовать DSL клиентам, которые не могут получить SDSL или ADSL из-за своей удаленности от офиса провайдера. IDSL позволяет работать клиентам на расстоянии до 11 километров от офиса провайдера. IDSL работает на симметричной скорости 128 Кбит/с.
- **Высокоскоростной DSL** (HDSL – High-bit-rate DSL). Обеспечивает скорость на уровне канала T1 (1,544 Мбит/с) через обычный медный телефонный провод без использования повторителей. Требуются два провода витой пары, чего не имеют многие линии UTP голосового класса.

Коаксиальные и оптоволоконные кабели используются для доставки пользователям сотен телевизионных каналов, при этом один или несколько каналов в этих кабелях выделяют для передачи данных. Полоса пропускания разделяется между пользователями в одной локальной области, поэтому скорость не всегда остается постоянной. Например, если Майк пытается скачать программу из сети Интернет в 19:30, он, скорее всего, получит гораздо меньшую скорость, чем в 10:00, потому что многие люди вечером приходят домой с работы и одновременно входят в Интернет. Поскольку большее число жителей одного дома используют доступ в сеть Интернет, скорость передачи данных у всех (в т.ч. у Майка) снижается.

Постоянное соединение. В отличие от подключений по модему, линии DSL и кабельные модемы подключены к сети Интернет постоянно. Не требуется выполнять те же шаги, как при установлении соединения Dial-Up. Однако при всем удобстве, это может привести к большой проблеме безопасности, поскольку хакеры ищут именно такие типы соединений. Системы, использующие эти типы соединений, всегда в сети и доступны для сканирования, анализа, взлома и атак. Часто именно такие системы используются при DDoS-атаках. Поскольку эти системы включены все время, нападающие устанавливают на них троянские программы, которые бездействуют до тех пор, пока не получают команды от злоумышленника о начале атаки на жертву. Многие DDoS-атаки используют системы, подключенные к Интернет с помощью DSL и кабельных модемов, в качестве сообщников. При этом владелец такого компьютера, как правило, не знает, что его компьютер используется для атак на другие системы.

Совместное использование одной среды передачи данных также вызывает множество проблем безопасности, поскольку в такой сети с помощью сниффера можно легко просматривать трафик соседей. В настоящее время многие кабельные модемы на канальном уровне выполняют шифрование данных, передаваемых по общим линиям связи.

Ссылки по теме:

- CableModemInfo.com

11.5. VPN

Виртуальная частная сеть (VPN – Virtual Private Network) – это безопасное частное соединение через общедоступные сети или иные небезопасные среды, как показано на Рисунке 5-53. Это частное соединение, поскольку в нем используются протоколы шифрования и туннелирования для обеспечения конфиденциальности и целостности передаваемых данных. Важно помнить, что технология VPN для работы требует создания туннеля, обеспечивающего шифрование трафика. Для VPN могут быть использованы такие протоколы, как PPTP (Point-to-Point Tunneling Protocol), IPSec и L2TP.



Рисунок 5-53. VPN обеспечивает виртуальную выделенную линию связи между двумя субъектами через публичную сеть

На отправляющей и принимающей стороне должно использоваться оборудование и программное обеспечение, необходимое для создания зашифрованного туннеля, который обеспечивает защиту (приватность) соединения. Протокол туннелирования шифрования используется для шифрования данных и защиты информации при ее передаче через недоверенные публичные сети, такие как Интернет.

Удаленные пользователи или мобильные сотрудники могут использовать VPN для подключения к сети компании и доступа к своей электронной почте, сетевым ресурсам и корпоративным активам. На компьютере удаленного пользователя должно быть установлено необходимое программное обеспечение, позволяющее ему использовать VPN. Пользователь сначала создает PPP-подключение к провайдеру, а провайдер соединяет его с нужной сетью. PPP инкапсулирует датаграммы для их передачи по телекоммуникационным каналам. После установления соединения PPP программа пользователя инициирует создание VPN-соединения с сетью компании. Поскольку данные, которыми будут обмениваться две стороны, должны шифроваться, обе стороны проходят через этап «рукопожатия», согласовывая тип шифрования и ключ, который будет использоваться для шифрования передаваемых данных. Провайдер участвует в этом только на этапе создания PPP-соединения, которое является фундаментом для VPN-соединения. Как только PPP-соединение установлено, участие провайдера заканчивается, и начинается этап заключения соглашения о параметрах VPN, которые будут использоваться в дальнейшем для обеспечения защищенного взаимодействия пользователя и сети компании. После завершения этого этапа, пользователь может безопасно взаимодействовать с сетью компании через новое виртуальное соединение.

Соединения VPN могут использоваться не только удаленными пользователями для доступа к сети, но и для обеспечения связи между двумя маршрутизаторами (часто это называют подключениями шлюз-шлюз). Это гибкое соединение, оно требует только наличия у каждой из соединяющихся сторон программного обеспечения VPN, необходимых протоколов и одинаковых механизмов шифрования. После того, как VPN-соединение будет установлено, пользователь сможет получить доступ к сетевым ресурсам.

До этого момента мы обсуждали VPN, работающий через соединения Dial-Up и Интернет, однако соединение VPN также может быть установлено между межсетевыми экранами, обладающими функциональностью VPN. В сети, устройство VPN находится на внешнем краю домена безопасности. Если компания устанавливает межсетевой экран, имеющий встроенную функциональность VPN, у нее появляется возможность централизовать администрирование VPN и самого межсетевого экрана. При использовании такого варианта, поступающие в сеть пакеты могут быть зашифрованы в рамках VPN-соединения и их необходимо предварительно расшифровать, чтобы межсетевой экран смог проверить пакеты и принять решение об их блокировке или пропуске в сеть. Поскольку выполняется дополнительная обработка поступающих пакетов, возникает больше накладных расходов, что приводит к снижению производительности. Однако на сегодняшний день наибольшая часть такой обработки производится на аппаратном уровне, обеспечивающем значительно более высокоскоростную работу, по сравнению с чисто программной обработкой.

Протоколы туннелирования

Ранее было сказано, что VPN использует протоколы туннелирования, но что они из себя представляют? *Туннель* (tunnel) – это виртуальный маршрут через сеть, по которому передаются инкапсулированные и, возможно, зашифрованные пакеты.

Если нужно соединить две находящиеся в разных странах сети, в которых используется протокол NetBEUI (NetBIOS Enhanced User Interface), в чем состоит основная проблема? Протокол NetBEUI не является маршрутизируемым. Поэтому для взаимодействия этих двух сетей пакеты NetBEUI должны быть инкапсулированы в маршрутизируемый протокол, такой как IP. Такая инкапсуляция происходит постоянно при работе в сети Интернет, а также при межсетевом взаимодействии. Если сеть Ethernet подсоединена к магистрали FDDI, сеть FDDI не понимает формат кадра Ethernet, поэтому пакеты должны быть инкапсулированы в протокол FDDI при их передаче через сеть FDDI. Если нужно соединить через Интернет две сети, использующие IPX, пакеты IPX должны быть инкапсулированы в протокол, который понимает Интернет, например, IP. Инкапсуляция является первой причиной использования туннеля.

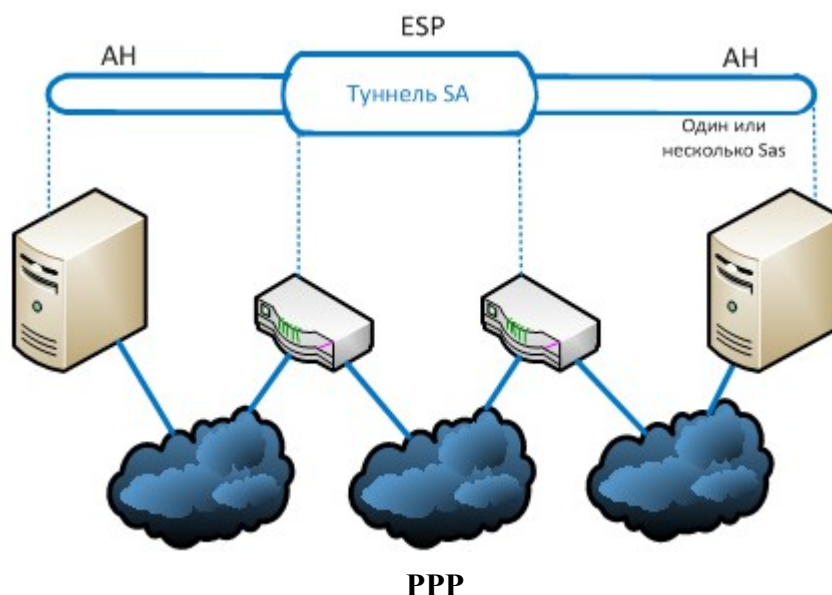
Второй причиной использования туннеля является применение инкапсуляции одновременно с шифрованием. При этом причины для использования инкапсуляции остаются те же, а шифрование требуется для защиты целостности и конфиденциальности передаваемых через недоверенную среду данных. Таким образом, применяются два способа туннелирования трафика через другую сеть.

Туннелирование является основным компонентом для VPN, т.к. именно таким образом создаются соединения VPN. В основном используются три основных протокола туннелирования для соединений VPN: PPTP, L2TP и IPSec (IPSec подробно рассматривается в Домене 06).

IPSec детально рассматривается в Домене 06, но давайте сейчас вкратце обсудим некоторые возможности для настройки этого набора протоколов.

IPSec может быть настроен для обеспечения транспортной смежности (transport adjacency), что означает, что к пакету применяется более одного протокола безопасности (ESP и AH). Также IPSec может быть настроен для обеспечения дополнительного туннелирования (iterated tunneling), при котором один туннель IPSec туннелируется через другой туннель IPSec, как показано на следующем рисунке. Дополнительное туннелирование применяется в случае, когда трафику требуются различные уровни защиты на различных частях его маршрута. Например, если IPSec

туннель начинается с внутреннего узла, трафик до внутреннего пограничного маршрутизатора может передаваться без шифрования, соответственно будет применяться только протокол АН. Но после пограничного маршрутизатора данные передаются через Интернет в другую сеть, на этом участке маршрута данные нуждаются большей защите. Таким образом, в рамках рассмотренного примера, пакеты сначала передаются через частично защищенный туннель по внутренней сети, а затем – через безопасный туннель по сети Интернет.



Перед углублением в подробности протоколов туннелирования, давайте еще раз взглянем на протокол инкапсуляции. Протоколы туннелирования и инкапсуляции связаны между собой, поскольку протоколы туннелирования используют инкапсуляцию. Инкапсуляция означает, что в кадр добавляется дополнительный заголовок, и, возможно, еще одно окончание. Протокол туннелирования добавляет эти заголовки и окончания для того, чтобы пакеты могли быть переданы по сети определенного типа. Таким образом, если вам необходимо передавать пакеты сетевого и транспортного протоколов IPX/SPX через сеть Интернет, шлюз будет добавлять необходимые заголовки и окончания TCP/IP, необходимые для маршрутизации в Интернете. После того, как эти пакеты будут доставлены до места назначения, дополнительные заголовки и окончания удаляются, и пакеты направляются на компьютер получателя.

PPP (Point-to-Point Protocol - Протокол "точка-точка") в действительности является не протоколом туннелирования, а протоколом инкапсуляции. Он не нуждается в добавлении специальных заголовков и окончаний, которые будут удалены в пункте назначения. PPP позволяет передавать трафик TCP/IP через среду, созданную для передачи голосовых данных.

PPP используется для инкапсуляции сообщений и их передачи через последовательную линию. Он позволяет передавать протоколы TCP/IP (и другие) через телекоммуникационные каналы. PPP используется для установления соединений между маршрутизаторами, от маршрутизатора до пользователя, а также между пользователями. Кроме того, он применяется для создания соединения между компьютером пользователя и точкой входа в Интернет (PoP – Point of Presence), которая обычно представляет собой пул модемов и сервер доступа, размещенный у провайдера. Пользователь соединяется с этой PoP через телекоммуникационную линию и взаимодействует с ней посредством PPP.

При подключении к Интернету вы можете использовать последовательное асинхронное соединение Dial-Up или соединение «точка-точка», например, линию T1. Линия T1 является *последовательным соединением* (serial connection), т.е. биты в ней следуют один за другим, в отличие от *параллельного соединения* (parallel connection), в котором биты передаются одновременно. Этим последовательным соединением является линия телефонной компании,

которая ничего не знает о сетевых протоколах, таких как IP, IPX, AppleTalk и т.д., которые используются для взаимодействия компьютеров. Поэтому, этот трафик должен быть инкапсулирован, прежде чем он попадет в это последовательное соединение – и это именно то, чем занимается PPP. PPP *инкапсулирует* данные, поступающие от компьютера или из сети, т.е. он преобразует данные в правильный формат, необходимый для передачи по телекоммуникационному соединению. PPP выполняет кадрирование данных с использованием стартовых и стоповых битов, которые позволяют на другой стороне понять, как следует обрабатывать эти данные. PPP позволяет нам настроить и установить соединение с сетью Интернет.

После установления соединения, пользователь должен пройти процедуру аутентификацию на сетевом сервере аутентификации на сайте провайдера. PPP может использовать PAP, CHAP или EAP для выполнения этой процедуры. Если пользователь пытается установить соединение с сетью своей компании, должен быть выполнен дополнительный шаг: протокол должен передать этот сеанс PPP в реальную корпоративную сеть компании. Что это означает? Теперь эти данные, инкапсулированные в PPP, должны быть переданы через Интернет, который не понимает PPP и использует IP. Поэтому возникает необходимость в том, чтобы еще один протокол инкапсулировал данные PPP в пакеты IP, и создал туннели для их передачи через Интернет в корпоративную сеть, как это показано на Рисунке 5-54. PPP инкапсулирует данные для передачи через частную сеть (между вами и вашим провайдером), а протокол туннелирования инкапсулирует данные для передачи через публичные сети, такие как Интернет.

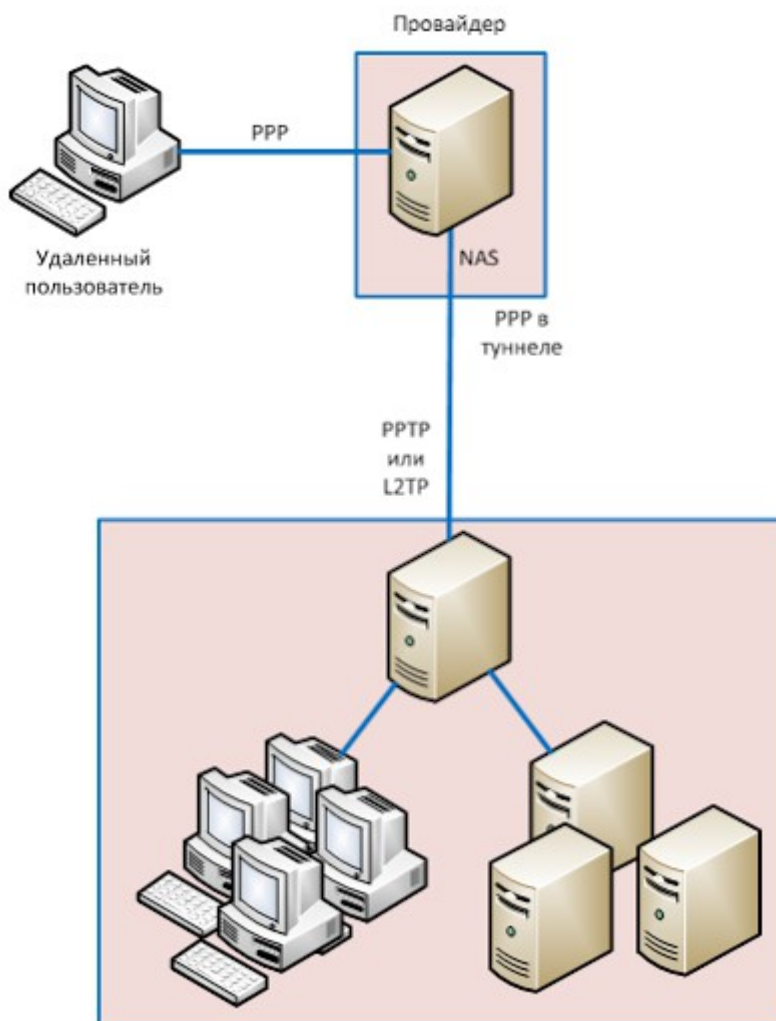


Рисунок 5-54. VPN-соединение может быть использовано в качестве безопасного расширения PPP-соединения

Для создания VPN применяются три основных протокола туннелирования: IPSec, PPTP и L2TP. Они снижают стоимость удаленного доступа к сети, поскольку пользователь может подключиться к местному провайдеру вместо того, чтобы устанавливать модемное соединение посредством междугородного (или международного) звонка непосредственно на модемный пул в корпоративной сети.

PPP в большинстве случаев заменил SLIP – старый протокол, который использовался для инкапсуляции данных, передаваемых через последовательные соединения. PPP имеет несколько возможностей, которых нет у SLIP:

- Осуществляет сжатие заголовков и данных для повышения эффективности и лучшего использования полосы пропускания
- Имеет коррекцию ошибок
- Поддерживает различные методы аутентификации
- Может инкапсулировать протоколы, отличные от IP
- Не требует наличия на обеих сторонах IP-адресов, присвоенных заранее, до начала передачи данных

PPTP

Зачем нужны протоколы туннелирования? Это связано с тем, что некоторые протоколы не могут маршрутизироваться через определенные сети. Выше мы говорили исключительно о кадрах PPP, которые не маршрутизируются через сеть Интернет, но Интернет не понимает и ряд других протоколов, например, IPX, NetBEUI и AppleTalk. Самостоятельно эти кадры не могут найти правильный маршрут через Интернет к получателю, для этого им нужна поддержка. Именно для этого и нужны протоколы туннелирования. Представьте себе паром, который перевозит автомобили на другой берег реки. Автомобиль не может сам переправиться на другую сторону, т.к. он не предназначен для перемещения по воде, а паром может перемещаться по воде, и может перевезти автомобиль. Аналогично протокол туннелирования может перемещаться по сети Интернет, основанной на IP.

PPTP (Point-to-Point tunneling protocol - Туннельный протокол "точка-точка") – это протокол Microsoft, позволяющий удаленным пользователям устанавливать PPP-соединения с локальным провайдером, а затем создавать безопасные соединения VPN до пункта назначения. В течение многих лет PPTP был де-факто стандартом протокола туннелирования, однако в настоящее время новым стандартом де-факто для VPN стал IPSec, который рассматривается в Домене 06.

Хотя по умолчанию туннелирование не обеспечивает шифрование пользовательских данных, в большинстве реализаций туннелирование и шифрование используются совместно. При использовании PPTP, содержимое PPP шифруется с помощью MPPE (Microsoft Point-to-Point Encryption), используя MS-CHAP или EAP-TLS. Используемые для этого ключи шифрования генерируются в процессе аутентификации между пользователем и сервером аутентификации.

Помимо шифрования, кадры должны быть инкапсулированы. В этой технологии применяется последовательность инкапсуляций. Пользовательские данные инкапсулируются в PPP, затем этот кадр инкапсулируется PPTP с заголовками GRE (Generic Routing Encapsulation) и IP, как показано на Рисунке 5-55. Такая инкапсуляция позволяет полученному в результате кадру маршрутизироваться через публичные сети, такие как Интернет.

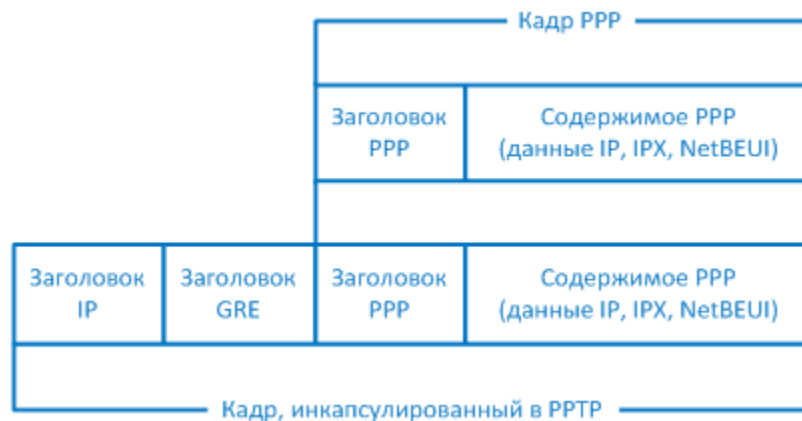


Рисунок 5-55. PPTP инкапсулирует кадр PPP

Единственным ограничением PPTP является то, что он может работать только через сети IP, поэтому для передачи данных через соединения Frame Relay, X.25 и ATM должны применяться другие протоколы. Cisco разработала протокол L2F (Layer 2 Forwarding - Протокол маршрутизации на втором уровне), который туннелировал PPP-трафик через указанные типы сетей, но IETF (Internet Engineering Task Force - Специальная комиссия интернет-разработок) предложила в рамках этой работы обеспечить совместимость с PPTP. В результате Cisco был разработан протокол L2TP (Layer 2 Tunneling Protocol - Протокол туннелирования на втором уровне), который сочетает в себе лучшее от PPTP и L2F.

L2TP

L2TP реализует функциональность PPTP, однако он в сочетании с IPSec обеспечивает более высокий уровень безопасности и, кроме того, может работать не только через сети IP. В L2TP не предусмотрены функции шифрования и аутентификации, поэтому при их необходимости он должен быть объединен с IPSec.

Процессы, которые L2TP использует для инкапсуляции, похожи на аналогичные процессы, используемые PPTP. Кадры PPP инкапсулируются в L2TP.

Когда система получателя получает кадр L2TP, она приступает к обработке заголовков. Когда она доходит до окончания IPSec, она выполняет проверку целостности и аутентификацию кадра. Затем она использует информацию из заголовка ESP (Encapsulating Security Payload), чтобы расшифровать остальные заголовки и сами данные.

Ниже указаны различия между PPTP и L2TP:

- PPTP может работать только в сетях IP. L2TP может работать и создавать туннели через другие сети, такие как Frame Relay, X.25 и ATM.
- PPTP – это шифрующий протокол, а L2TP – нет. В L2TP отсутствует безопасность, позволяющая назвать его чистым VPN-решением. L2TP часто используется в сочетании с IPSec для реализации шифрования.
- L2TP поддерживает TACACS + и RADIUS, а PPTP – нет.

Резюме по протоколам туннелирования

Point-to-Point Tunneling Protocol (PPTP):

- Предназначен для соединений клиент/сервер
- Устанавливает единственное соединение точка-точка между двумя компьютерами
- Работает на канальном уровне
- Передает данные только по сетям IP

Layer 2 Forwarding (L2F):

- Создан Cisco до L2TP
- Объединен с PPTP, результатом чего стал L2TP
- Обеспечивает взаимную аутентификацию
- Без шифрования

Layer 2 Tunneling Protocol (L2TP):

- Гибрид L2F и PPTP
- Устанавливает единственное соединение точка-точка между двумя компьютерами
- Работает на канальном уровне
- Передает данные по сетям нескольких различных типов, а не только по сетям IP
- Используется совместно с IPSec для обеспечения безопасности

IPSec:

- Работает с несколькими соединениями одновременно
- Обеспечивает защищенную аутентификацию и шифрование
- Поддерживает только сети IP
- Сосредоточен на коммуникациях LAN-LAN, а не пользователь-пользователь
- Работает на сетевом уровне и обеспечивает безопасность выше уровня IP
- Может работать в режиме туннелирования (при этом защищается содержимое и заголовок), либо в транспортном режиме (защищается только содержимое)

11.6. Протоколы аутентификации

PAP (Password Authentication Protocol - Протокол аутентификации пароля) используется удаленными пользователями для аутентификации через соединения PPP. Он обеспечивает идентификацию и аутентификацию пользователя, пытающегося с удаленной системы получить доступ к сети. Этот протокол требует, чтобы пользователь ввел пароль для аутентификации. Пароль и имя пользователя передаются по сети на сервер аутентификации через соединение, которое было установлено с помощью PPP. Сервер аутентификации имеет базу данных реквизитов доступа пользователей для сравнения с представленными пользователем учетными данными в процессе аутентификации.

PAP является одним из наименее безопасных методов аутентификации, т.к. он передает учетные данные открытым текстом, что позволяет легко перехватить их с помощью сетевого sniffера. Некоторые системы, в случае невозможности договориться о других методах аутентификации, возвращаются к PAP, хотя это не рекомендуется. Во время процесса «рукопожатия» при установлении соединения, две стороны договариваются о способе аутентификации, параметрах подключения, скорости потока данных, а также других параметрах. При этом стороны пытаются договориться об использовании наиболее безопасного метода аутентификации: они могут начать с EAP, но если один из компьютеров не поддерживает EAP, они будут пытаться согласовать CHAP, если один из компьютеров не поддерживает CHAP, они будут вынуждены использовать PAP. Если аутентификация с использованием PAP неприемлема, администратор должен настроить RAS для выполнения аутентификации только с использованием CHAP или более безопасных протоколов, а использование PAP должно быть исключено.

CHAP (Challenge Handshake Authentication Protocol - Протокол аутентификации по методу "вызов-приветствие") учитывает некоторые уязвимости, присущие PAP. Для выполнения аутентификации пользователя он использует механизм запрос/ответ, вместо отправки пароля. Если пользователю нужно установить соединение PPP и обе стороны договорились об использовании для аутентификации протокола CHAP, компьютер пользователя посылает серверу аутентификации запрос на вход. В ответ сервер отправляет пользователю запрос,

представляющий собой случайное значение. Этот запрос шифруется компьютером пользователя с использованием пароля пользователя в качестве ключа шифрования, зашифрованный ответ возвращается серверу. Сервер аутентификации использует хранящийся на своей стороне пароль этого пользователя в качестве ключа и расшифровывает значение ответа, сравнивая его с первоначально отправленным значением. Если два значения совпадают, сервер аутентификации делает вывод, что пользователь ввел правильный пароль, и аутентификация считается успешной. Эти шаги, выполняемые CHAP, показаны на Рисунке 5-56.

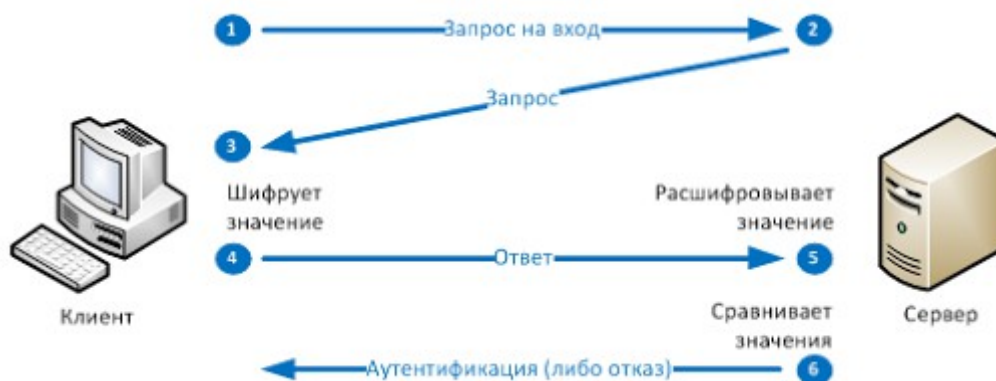


Рисунок 5-56. CHAP использует механизм запрос/ответ и не требует, чтобы пользователь отправлял свой пароль по сети

Сравнение PAP и CHAP

Характеристики PAP:

- Передает учетные данные открытым текстом
- Применение этого протокола снижается, поскольку он не обеспечивает высокого уровня безопасности
- Поддерживается большинством сетей и NAS

Характеристики CHAP:

- Используется так же как и PAP, но обеспечивает при этом более высокий уровень безопасности
- При аутентификации используется метод запрос/ответ
- Используется удаленными пользователями, маршрутизаторами и NAS для проведения аутентификации перед предоставлением подключения

PAP уязвим к сниффингу, т.к. он отправляет все данные (включая пароли) открытым текстом. Кроме того, PAP уязвим к атакам «человек посередине». В отличие от него, CHAP не подвержен атакам «человек посередине», поскольку он работает в режиме запрос/ответ, обеспечивая взаимодействие сервера аутентификации только с пользователем, который имеет необходимые учетные данные.

EAP (Extensible Authentication Protocol - Расширяемый протокол аутентификации) также поддерживается PPP. Фактически, EAP не является конкретным механизмом аутентификации, как PAP или CHAP. Вместо этого он представляет собой основу, позволяющую многим различным видам технологий аутентификации применяться в процессе установления соединений PPP. Как следует из названия, он расширяет возможности аутентификации, позволяя использовать одноразовые пароли, токены, биометрию, Kerberos и другие механизмы, которые могут появиться в будущем. Когда пользователь подключается к серверу аутентификации и оба они поддерживают EAP, они договариваются между собой о выборе возможного метода аутентификации из большего списка.

Ссылки по теме:

- Linux PPP HOWTO, Chapter 16, “If Your PPP Server Uses PAP,” by Corwin-Light Williams and Joshua Drake (2000)
- RFC 2284 – PPP Extensible Authentication Protocol (EAP)

11.7. Рекомендации по удаленному доступу

Удаленные пользователи должны быть идентифицированы и аутентифицированы, а их деятельность должна контролироваться на предмет отсутствия опасных действий и нарушений прав доступа. Всем удаленным пользователям должны быть предоставлены только действительно необходимые им права доступа к сети, кроме того, их права доступа должны пересматриваться на ежегодной основе. Например, если Брайан работал в группе разработчиков программного обеспечения и ему требовался удаленный доступ для работы из дома, а затем он был переведен в группу логистики, вероятно необходимость в удаленном доступе для него отпала, поэтому его права удаленного доступа следует отозвать.

Допустимые действия и поведение, связанные с удаленным доступом, должны быть отражены в политике безопасности компании, либо в политиках, непосредственно относящихся к удаленному доступу. В политике следует указать, кто, что и когда должен делать. С этой политикой должны быть ознакомлены все пользователи, и она должна быть им постоянно доступна, что поможет устранить путаницу и позволит компании принимать меры к нарушителям требований политики.

Модемы удаленного доступа должны быть настроены на ответ после четвертого гудка. Злоумышленники обычно настраивают свое программное обеспечение на переход к другому телефонному номеру после двух – трех гудков, поскольку в большинстве случаев столь длительное ожидание может происходить на голосовой телефонной линии, а не на линии, предназначенной для передачи данных. Настройка модема на ответ после четвертого (или более) гудка делается, чтобы перехитрить атакующего.

Все системы удаленного доступа должны быть, по возможности, сгруппированы вместе: все серверы удаленного доступа желательно разместить в одной серверной комнате, они должны администрироваться одним лицом (группой лиц). Это поможет обеспечить согласованность деятельности по поддержке и мониторингу отдельных точек удаленного доступа, а также сохранять уверенность, что вносимые изменения отражаются одновременно на всех точках доступа. Кроме того, это облегчает проведение централизованного аудита и журналирования событий, снижается вероятность, что администраторы забудут про отдельную точку доступа, позволяющую пользователям получить доступ к внутренней среде компании.

Должна применяться строгая двухфакторная аутентификация пользователей с использованием серверов RADIUS или TACACS+, которые были описаны в Домене 02. Если передаваемые пользователями данные являются конфиденциальными или критичными для компании, пользователи должны подключаться к сети компании через VPN. Должны обеспечиваться различные уровни безопасности для различных типов подключающихся к сети пользователей, каждый пользователь должен обладать только теми правами доступа и разрешениями, которые ему необходимы.

Компании могут ограничивать доступ на основе адреса внешнего компьютера, который пытается получить доступ. Если адрес неизвестен или не внесен в ACL, запрос на подключение должен отклоняться. Должен быть внедрен межсетевой экран, позволяющий удаленным пользователям получить доступ только к разрешенным администратором сервисам и портам.

На сервере RAS может быть настроено использование идентификации вызывающего абонента (Caller ID - определитель номера), выполнение обратного вызова, а также применение двухфакторной аутентификации. Функциональность Caller ID позволяет принимать решения о возможности доступа на основе номера телефона, с которого исходит

вызов – для этого должен быть предварительно настроен список разрешенных и/или запрещенных телефонных номеров. При этом атакующему для получения несанкционированного доступа потребуется выполнять вызов с разрешенного телефонного номера или скомпрометировать оборудование телефонной станции. Функция обратного вызова (callback) требует, чтобы сервер RAS самостоятельно выполнял звонок пользователю, запрашивающему доступ. При этом после аутентификации пользователя на сервере RAS, сервер RAS сбрасывает соединение и перезванивает пользователю по заранее настроенному номеру телефона.

Функции определения номера и обратного вызова – это прекрасно, но они часто оказываются непрактичными, поскольку требуют, чтобы пользователи всегда звонили со стационарных телефонов, когда им нужен доступ в сеть. А в реальности многим пользователям нужен удаленный доступ к сети, когда они находятся в дороге или постоянно перемещаются между различными офисами

Ссылки по теме:

- Windows 2000 Technical Resources: How It Works—Remote Access
- “Evolution of the Network Router Market,” by John Rendleman, InformationWeek (Aug. 19, 2002)
- The Evolution of the Remote Access Server (RAS) to a Universal Port-Enabled Platform, Internet Engineering Consortium

12. Беспроводные технологии

Беспроводные коммуникации используются гораздо чаще, чем мы думаем, в различных частотных диапазонах работает множество широкополосных беспроводных технологий передачи данных. Широкополосная беспроводная передача сигналов может использовать те же полосы частот, что и, например, микроволновые печи, спутники, радары и радиолубовительские передатчики. Мы используем беспроводные технологии для эфирного и спутникового телевидения, сотовых телефонов, шпионажа, наблюдения, а также для механизмов открывания гаражных дверей и многого другого. В этом разделе основное внимание уделяется использованию беспроводных технологий в среде LAN.

12.1. Беспроводные коммуникации

Когда два человека разговаривают друг с другом, они используют беспроводные коммуникации, поскольку их голосовые связки изменяют звуковые волны, которые без проводов переносят сигналы другому человеку. Беспроводные коммуникации используют передачу сигналов с помощью радиоволн через воздух или безвоздушное пространство.

Сигнал характеризуется *частотой* (frequency) и *амплитудой* (amplitude). Частота сигнала определяет, какое количество данных может быть передано и насколько далеко. Чем выше частота, тем больше данных может переносить сигнал, однако с повышением частоты растёт его подверженность атмосферным помехам. Т.е. более высокая частота может позволить передать больше данных (за единицу времени), но на более короткое расстояние.

В проводных сетях, каждый компьютер и устройство подключены к сети собственным кабелем определенного типа. В беспроводных технологиях, каждое устройство должно работать совместно с другими беспроводными устройствами в рамках общего отведенного радиочастотного спектра. Этот спектр частот ограничен, он не может расти по мере увеличения числа устройств, которым нужно его использовать. То же самое происходит в Ethernet – все компьютеры в сегменте используют общую среду, и только один компьютер может передавать данные в каждый момент времени, иначе возможны коллизии. Проводные сети Ethernet используют технологию CSMA/CD (выявление коллизий). Беспроводные технологии на самом деле очень похожи на Ethernet, но они используют CSMA/CA

(предотвращение коллизий). Беспроводное устройство посылает широковещательное сообщение о том, что оно собирается передавать данные. Получение такого сообщения другими устройствами, подключенными к той же общей среде, заставляет их отложить свои потребности в передаче информации. Это делается для устранения (или уменьшения вероятности) коллизий (обе версии CSMA были описаны ранее в этом Доме в разделе «CSMA»).

Множество технологий было разработано для совместного использования беспроводными устройствами этого ограниченного объема среды передачи данных. Мы рассмотрим две различные технологии распространения спектра: псевдослучайное изменение частоты (frequency hopping) и прямая последовательность (direct sequence). Цель каждой из этих беспроводных технологий заключается в разделении имеющиеся частоты на отдельные части, чтобы позволить устройствам эффективно совместно использовать этот ограниченный ресурс.

Расширение спектра

Для беспроводных технологий, как и для других технологий и отраслей, выделяются конкретные *спектры* (spectrum), или диапазоны частот, которые используются ими для передачи сигналов. В Соединенных Штатах, выделением частот занимается FCC (Federal Communications Commission - Федеральное агентство по связи), которое устанавливает свои собственные ограничения. **Расширение спектра** (spread spectrum) означает, что кто-то некоторым способом распространяет отдельные сигналы через выделенные частоты. При использовании технологии расширения спектра отправитель передает свои данные на частотах, для работы на которых он имеет разрешение. Это позволяет более эффективно использовать имеющуюся полосу пропускания, поскольку передающие системы могут одновременно использовать более одной частоты. Это можно сравнить с последовательной и параллельной передачей данных. При последовательной передаче все биты помещаются в один канал и следуют один за другим. При параллельной передаче биты могут помещаться в несколько каналов и передаваться одновременно. Параллельная передача обеспечивает более высокую скорость, т.к. одновременно используются несколько каналов передачи данных. Это похоже на очередь в кассу в продуктовом магазине. Если касса только одна, она не может обслужить всех покупателей за короткое время, и в кассу выстраивается очередь. Если директор магазина принимает решение создать несколько линий касс, чтобы распараллелить обслуживание клиентов, очереди можно будет избежать. Таким образом, широкополосное распространение сигнала позволяет передавать данные в параллельном режиме, позволяя отправителю и получателю передавать и принимать данные по более чем одной частоте.

Мы рассмотрим два типа расширения спектра: FHSS (frequency hopping spread spectrum - псевдослучайное изменение рабочей частоты) и DSSS (direct sequence spread spectrum - прямая последовательность).

Ссылки по теме:

- “Spread Spectrum Scene Primer,” Spread Spectrum Scene Online
- Get IEEE 802 Program: IEEE 802.11 LAN/MAN Wireless LANS, IEEE Standards Association

Метод расширения спектра с помощью псевдослучайного изменения рабочей частоты (FHSS – Frequency Hopping Spread Spectrum). FHSS берет общий объем полосы пропускания (спектр) и разбивает его на более мелкие подканалы. Отправитель и получатель работают на одном из этих подканалов определенное время, а затем переходят на другой подканал. Отправитель отправляет первую часть данных на одной частоте, вторую – на другой частоте и т.д. Алгоритм FHSS определяет, какие конкретно частоты будут использоваться и в каком порядке. Это называется последовательностью изменений частоты отправителем и

получателем.

Большой проблемой при использовании беспроводных технологий являются помехи, поскольку они могут разрушить сигналы в процессе их передачи. Помехи могут быть вызваны другими устройствами, работающими на той же частоте. Сигналы различных устройств «наступают друг другу на пятки» и искажают передаваемые данные. Подход FHSS позволяет снизить влияние помех за счет перескакивания между различными частотами таким образом, чтобы не оказывать существенного влияния на другие устройства, работающие на той же частоте.

Также следует отметить, что подход с перескакиванием между частотами существенно затрудняет для злоумышленников перехват и восстановление данных. FHSS широко используется в военных беспроводных коммуникационных устройствах, поскольку он может позволить врагу перехватить передаваемые данные, только если он будет знать последовательность изменений частоты. Однако в современных устройствах WLAN последовательность изменений частоты заранее известна и не обеспечивает какой-либо безопасности. Это связано с тем, что получатель должен знать эту последовательность, чтобы иметь возможность получать данные.

Но как работает FHSS? Отправитель и получатель перескакивают от одной частоты на другую, основываясь на заранее определенной последовательности. Несколько пар отправителей и получателей могут одновременно обмениваться данными через тот же набор частот, т.к. все они используют различные последовательности. Предположим, что мы с вами используем последовательность изменений частоты 1, 5, 3, 2, 4, а Николь и Эд – 4, 2, 5, 1, 3. Я отправляю свое первое сообщение на частоте 1, а Николь в то же время отправляет свое первое сообщение на частоте 4. Следующую часть своих данных я передаю на частоте 5, затем на частоте 3 и так до тех пор, пока все части данных не будут доставлены получателю, т.е. вашему беспроводному устройству. Ваше устройство сначала прослушивает полсекунды частоту 1, затем частоту 5 и так далее до тех пор, пока оно не получит все части данных, которые передаются в это время на этих частотах. Устройство Эда прослушивает те же частоты, но в другое время и в другом порядке, поэтому его устройство игнорирует мои сообщения – он не синхронизирован заранее с моей последовательностью. Не зная правильного кода (последовательности), Эд считает мои сообщения фоновым шумом и не обрабатывает их.

Метод расширения спектра с помощью прямой последовательности (DSSS – Direct Sequence Spread Spectrum) использует иной подход, применяя к суб-биты к сообщениям. Суб-биты используются отправляющей системой, чтобы сгенерировать различные форматы данных перед их передачей. Принимающая сторона использует эти суб-биты для того, чтобы пересобирать сигнал в исходный формат данных. Суб-биты называют *чипами* (chips), применяемая последовательность которых называется *чиппинг-кодом* (chipping code).

После того, как данные отправителя объединены с чипами, для всех, кто не знает чиппинг-код, эти сигналы будут выглядеть как случайный шум. Поэтому такие последовательности иногда называют шумоподобными последовательностями. После объединения данных отправителя с последовательностью чипов, полученная информация в новой форме модулируется с несущим радиосигналом, приводится к необходимой частоте и передается. Что это значит? При беспроводной передаче данных, информация передается с помощью радиосигналов, которые работают на определенных частотах. Любые данные, передаваемые в этом режиме, должны иметь несущий сигнал, который работает в своем собственном конкретном диапазоне, который и является частотой. Вы можете представить себе это таким образом: после комбинирования данных с чиппинг-кодом, они помещаются в автомобиль (несущий сигнал), и автомобиль начинает движение по определенной дороге (частоте), чтобы добраться до пункта назначения.

Получатель производит обратную обработку. Сначала он демодулирует данные от несущего

сигнала (удаляет данные из автомобиля). Получатель должен знать правильную последовательность чипов, чтобы привести полученные данные в их первоначальный вид. Соответственно отправитель и получатель должны быть надлежащим образом синхронизированы.

Суб-биты позволяют производить исправление ошибок, аналогично четности, используемой в технологиях RAID. Если переданный с помощью FHSS сигнал поврежден, он отправляется повторно. А при использовании DSSS сигнал может быть восстановлен даже из искаженного (в некоторой степени) сообщения. Именно биты чиппинг-кода позволяют восстановить потерянную информацию. Использование кода расщепления позволяет снизить влияние помех, отслеживать множество передач и обеспечивает возможность исправления ошибок.

FHSS и DSSS. FHSS использует только часть от общей полосы пропускания в каждый момент времени, тогда как технология DSSS постоянно использует всю доступную полосу пропускания. DSSS распространяет сигналы в более широком диапазоне частот, тогда как FHSS использует узкополосную передачу.

Поскольку DSSS отправляет данные на всех частотах одновременно, эта технология обладает более высокой пропускной способностью, чем FHSS. Первый стандарт WLAN (802.11) использовал FHSS, но после того, как возникли требования по повышению пропускной способности, была реализована технология DSSS. При использовании FHSS, стандарт 802.11 может обеспечить передачу данных со скоростью от 1 до 2 Мбит/с. Стандарт 802.11b, использующий DSSS, обеспечивает передачу данных со скоростью до 11 Мбит/с.

Типы распространения спектра. Эта технология передает данные путем их «распространения» через широкий диапазон частот.

- FHSS (Frequency hopping spread spectrum) передает данные, изменяя частоты.
- DSSS (Direct sequence spread spectrum) использует иной подход, применяя к суб-биты к сообщениям и используя все доступные частоты одновременно.
- OFDM (Orthogonal frequency-division multiplexing - Ортогональное частотное разделение каналов с мультиплексированием) является цифровой схемой модуляции с несколькими несущими, которая уплотняет несколько модулированных несущих, уменьшает требуемую полосу пропускания. Модулированные сигналы ортогональны (перпендикулярны) и не мешают друг другу. OFDM использует набор частот узких каналов для повышения ее производительности на высоких частотах.

12.2. Компоненты WLAN

WLAN (Wireless LAN – беспроводная локальная вычислительная сеть) использует трансивер, называемый **точкой доступа** (AP – Access Point), который подключается к проводной сети с помощью кабеля Ethernet. Точка доступа является связующим звеном, позволяющим беспроводным устройствам получить доступ к ресурсам проводной сети, как это показано на Рисунке 5-57. Точка доступа – это компонент, связывающий проводной и беспроводный миры. Точки доступа находятся в фиксированных местах по всей сети и работают как коммуникационные маяки (beacon). Предположим, у пользователя есть беспроводное устройство с беспроводной сетевой картой, модулирующей его данные в радиочастотные сигналы, которые принимаются и обрабатываются точкой доступа. Передаваемые точкой доступа сигналы принимаются беспроводной сетевой картой, и преобразуется в цифровой формат, который может понять устройство.

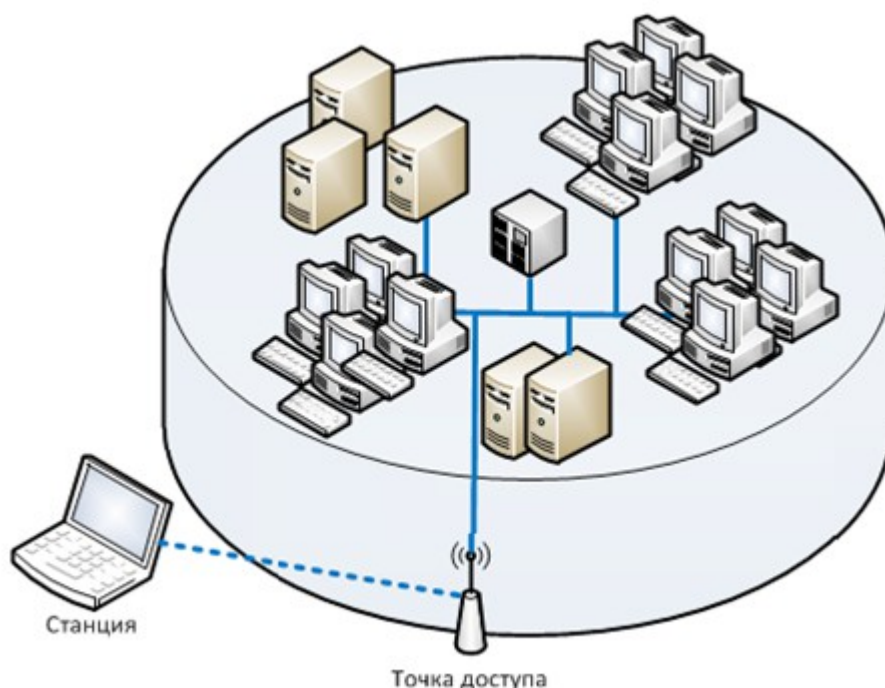


Рисунок 5-57. Беспроводное устройство должно аутентифицироваться на точке доступа

Если для соединения беспроводной и проводной сетей используются точки доступа, это называют **инфраструктурой WLAN** (infrastructure WLAN), которая используется для расширения существующей проводной сети. Если используется только одна точка доступа, которая не подключена к проводной сети, это называется работой в **автономном режиме** (standalone mode). При этом точка доступа работает просто в качестве беспроводного концентратора.

Для взаимодействия беспроводных устройств с точкой доступа, они должны быть настроены на работу по одному и тому же каналу. *Канал* (channel) – это определенная частота в рамках выделенного диапазона частот. Точка доступа настроена на работу по определенному каналу, а беспроводное устройство подстраивается на ту же частоту для возможности взаимодействия.

Любой узел, который хочет использовать WLAN, должен быть настроен на нужный **SSID** (Service Set ID). Различные узлы могут быть сегментированы по различным WLAN посредством использования различных SSID. Причины сегментирования WLAN аналогичны причинам сегментирования для проводных сетей: пользователям требуется доступ к различным ресурсам, они имеют различные бизнес-функции, либо имеют различные уровни доверия.

Специальный WLAN (Ad hoc WLAN) не имеет точек доступа. В таком WLAN беспроводные устройства взаимодействуют только друг с другом через беспроводные сетевые карты, и не используют централизованные устройства. Для организации специальной сети WLAN, на взаимодействующие узлы устанавливается программное обеспечение беспроводного клиента, которое настраивается на одноранговый режим работы. Затем пользователь открывает «Сетевое окружение» (на компьютере с Windows), операционная система находит другие компьютеры, работающие в аналогичном режиме, и отображает их пользователю.

ПРИМЕЧАНИЕ. Если беспроводные устройства работают в режиме инфраструктуры, точка доступа и беспроводные клиенты создают группу, называемую BSS (Basic Service Set). Этой группе присваивается имя, которое является значением SSID.

SSID обычно используется для аутентификации мобильного устройства на точке доступа. Чтобы устройство могло доказать, что ему разрешено взаимодействовать с проводной сетью, оно должно сначала предоставить действительное значение SSID. SSID не следует

рассматривать в качестве надежного механизма обеспечения безопасности, поскольку многие точки доступа транслируют свой SSID в сеть, что позволяет легко перехватить и использовать их нападающими.

Беспроводное устройство может проверить подлинность точки доступа двумя основными способами: OSA (Open System Authentication - Открытая аутентификация) и SKA (Shared Key Authentication - Аутентификация посредством общего ключа). OSA не требует, чтобы беспроводное устройство предоставляло точке доступа определенный криптографический ключ для аутентификации. Во многих случаях, мобильное устройство должно предоставить только правильное значение SSID. В реализациях OSA, вся передача ведется открытым текстом без применения шифрования. При этом злоумышленник может прослушивать трафик, перехватывать передаваемые данные при выполнении отдельных шагов аутентификация, а затем выполнить те же шаги, чтобы пройти аутентификацию и установить соединение с точкой доступа.

ПРИМЕЧАНИЕ. OSA просто означает, что устройству для аутентификации не требуется предоставлять свой криптографический ключ. В зависимости от продукта и его конфигурации, администратор сети может ограничить доступ для определенных MAC-адресов. Это по-прежнему будет считаться OSA.

Если точка доступа настроена на использование SKA, она направляет случайное значение беспроводному устройству. Устройство шифрует это значение своим криптографическим ключом и возвращает результат. Точка доступа расшифровывает его и сравнивает с первоначальным значением, если значения совпадают – устройство аутентифицировано. При этом подходе, беспроводное устройство аутентифицируется в сети, доказав, что оно обладает необходимым ключом шифрования. Этот метод основан на протоколе **WEP** (Wired Equivalent Privacy), который помимо этого обеспечивает зашифрованную передачу данных.

Как правило, WEP по умолчанию отключен на обычных беспроводных точках доступа. Если купивший его человек не обладает знаниями в области безопасности, он может не знать, как (и зачем) нужно настроить это устройство на безопасную работу. Без WEP не обеспечивается никакой секретности, точка доступа свободно распространяет данные о своем местонахождении и идентификационную информацию.

Было выявлено большое количество недостатков этого протокола. О новых решениях, разработанных для устранения этих недостатков, рассказано в разделе «802.11i» далее в этом Домене.

12.3. Беспроводные стандарты

Стандарты были разработаны для того, чтобы различные производители могли создавать различные продукты, которые могли бы без проблем работать совместно. Стандарты, как правило, разрабатываются на основе консенсуса, достигнутого различными производителями в конкретной отрасли. IEEE (Institute of Electrical and Electronics Engineers - Институт инженеров электротехники и электроники) разрабатывает стандарты для широкого спектра различных технологий, среди них и беспроводные технологии.

Первый стандарт WLAN – 802.11, был разработан в 1997 году, он обеспечивал скорость передачи данных 1-2-Мбит/с. Он работал в частотном диапазоне 2,4 ГГц. Этот диапазон не лицензирован FCC, т.е. компаниям и пользователям не нужно платить, чтобы работать в нем.

Стандарт 802.11 описывает взаимодействие беспроводных клиентов и точек доступа, спецификации их интерфейсов, определяет порядок передачи сигналов, описывает, как должна быть реализована аутентификация, связь и безопасность.

В основной стандарт 802.11 в настоящее время входит большой список стандартов (802.11a, 802.11b, 802.11i, 802.11g, 802.11h и т.д.). IEEE создал несколько исследовательских групп для работы по конкретным направлениям беспроводной связи. Каждая группа работает в своем направлении, она исследует и разрабатывает стандарты для определенного

направления. Буквы в названиях стандартов отражают порядок, в котором они были предложены и приняты.

802.11b

Стандарт 802.11b был первым расширением стандарта WLAN 802.11, в настоящее время он все еще является наиболее распространенным стандартом (хотя стандарт 802.11a был разработан и утвержден раньше, он не был реализован первым из-за возникших технических сложностей). 802.11b обеспечивает скорость передачи данных до 11 Мбит/с и работает в частотном диапазоне 2,4 ГГц. Он использует DSSS и обратно совместим с реализациями 802.11.

802.11a

Стандарт 802.11a стандарт использует другой метод модуляции данных на несущем радиосигнале. Тогда как 802.11b использует DSSS, 802.11a использует OFDM и работает в частотном диапазоне 5 ГГц. Из-за этих различий, 802.11a не имеет обратной совместимости с 802.11b и 802.11. Некоторые производители разработали продукты, которые могут работать с обоими реализациями (802.11a и 802.11b). При этом такие устройства должны быть правильно настроены или иметь возможность автоматически настраиваться на используемую технологию.

OFDM является схемой модуляции (FHSS и DSSS – два других примера схем модуляции), которая разбивает сигнал на несколько узкополосных каналов. Затем каналы модулируются и направляются по определенным частотам. Поскольку данные разделены на различные каналы (диапазоны), любые помехи окружающей среды будут негативно влиять лишь на небольшую часть сигнала. Это позволяет обеспечить более высокую пропускную способность. Также как FHSS и DSSS, OFDM является спецификацией физического уровня. OFDM может использоваться для передачи цифрового аудио и видео высокой четкости, а также трафика WLAN.

Эта технология дает преимущества в двух областях: скорость и частота. 802.11a обеспечивает скорость до 54 Мбит/с и не работает в переполненном диапазоне 2,4 ГГц. Диапазон частот 2,4 ГГц называют «грязными» частотами, поскольку в нем работают многие устройства – микроволновые печи, беспроводные телефоны и т.п. Во многих случаях это приводит к конкуренции при использовании этой частоты, вызывает потерю данных или неприемлемое качество сервиса. Поскольку стандарт 802.11a работает на более высоких частотах, он не подвержен этим проблемам (в отличие от стандартов 802.11b и 802.11g). Максимальная скорость при использовании 802.11a достигается на небольших расстояниях от точки доступа (до 7,5 метров).

Один из недостатков использования диапазона 5 ГГц состоит в том, что другие страны не выделили этот диапазон для использования в WLAN. Поэтому продукты 802.11a могут работать в США, но не обязательно, что они смогут работать в других странах.

802.11e

Стандарт 802.11e обеспечивает реализацию QoS и надлежащую поддержку мультимедийного трафика. Мультимедиа и другие виды критичных ко времени приложений, более чувствительны к задержкам в процессе передачи данных. QoS позволяет приоритезировать трафик и обеспечивает гарантированную доставку. Эта спецификация и ее возможности позволяют передавать различные типы данных по беспроводным сетям.

802.11f

Когда пользователь перемещается в пределах WLAN, беспроводному устройству часто приходится переключаться между различными точками доступа. Точка доступа покрывает небольшую площадь, и при перемещении пользователя из зоны одной точки доступа в зону другой точки доступа необходимо найти и поддержать сигнал, обеспечив непрерывное

подключение к сети. Это называется *роуминг*. Чтобы роуминг работал плавно, без разрывов, точки доступа должны взаимодействовать друг с другом. Если вторая точка доступа должна принять на себя коммуникации пользователя, она должна быть уверена, что этот пользователь был надлежащим образом аутентифицирован. Кроме того, она должна знать необходимые настройки соединения этого пользователя. Это означает, что первая точка доступа должна иметь возможность передавать эту информацию второй точке доступа. Передача информации между различными точками доступа при роуминге – это именно то, чем занимается стандарт 802.11f. В нем описывается, как эти данные могут использоваться совместно.

802.11g

Мы никогда не довольны, нам всегда нужно больше функций, больше возможностей и больше скорости. Стандарт 802.11g обеспечивает более высокую скорость передачи данных – до 54 Мбит/с. В основном это расширение, позволяющее увеличить скорость для существующих устройств 802.11b. Скорость работы устройств, соответствующих спецификации 802.11b составляет до 11 Мбит/с, а если устройство основано на 802.11g, оно остается обратно совместимым со старым оборудованием, но может при этом работать на более высокой скорости.

Так что же лучше, 802.11g или 802.11a? Они оба предоставляют высокую пропускную способность. 802.11g имеет обратную совместимость с 802.11b, поэтому его лучше применять, если у вас уже есть существующая инфраструктура. Но 802.11g по-прежнему работает в диапазоне 2,4 ГГц, в котором работает гораздо больше устройств. 802.11a работает в диапазоне 5 ГГц, в котором работает значительно меньше устройств, он не так переполнен. Но работа на более высоких частотах приводит к уменьшению зоны покрытия беспроводного устройства. Вполне вероятно, что выбор того или иного стандарта предопределил рынок, на котором победит один из них. Только время покажет, какой из стандартов выживет в этой войне стандартов.

802.11h

Как было отмечено ранее, 802.11a работает в диапазоне 5 ГГц, который может быть недоступен для передачи данных в странах, отличных от США. Стандарт 802.11h основан на спецификации 802.11a, при этом обеспечено соответствие требованиям европейских норм беспроводной связи, поэтому продукт, работающий в соответствии с этим стандартом, может использоваться в европейских странах.

802.11i

В стандарте 802.11 было выявлено множество недостатков, связанных с обеспечением безопасности. Это существенно уменьшило число желающих внедрять основанные на нем продукты, и создало серьезные уязвимости в системе безопасности у тех, кто все-таки внедрил такие продукты. Каждый из упомянутых выше стандартов WLAN основан на одной и той же модели безопасности, поэтому все они унаследовали те же недостатки.

Стандарт безопасности беспроводных технологий

Первый стандарт WLAN, IEEE 802.11, имеет огромное количество недостатков безопасности. Они содержатся как в основах самого стандарта, так и в различных его реализациях. Для решения этих проблем, новой исследовательской группой IEEE были проведены работы по улучшению стандарта и закрытию уязвимостей, которые были в нем выявлены. Среди недостатков оригинального стандарта 802.11 можно отметить отсутствие аутентификации пользователей, отсутствие взаимной аутентификации между беспроводным устройством и точкой доступа, недостатки протокола шифрования. Протокол шифрования позволяет изменить определенные биты в передаваемом сообщении так, чтобы получатель не смог выявить эти изменения. Отдельные компоненты механизма шифрования (ключ и векторы инициализации) не обеспечивают достаточной случайности процесса шифрования,

что позволяет легко взломать зашифрованные сообщения бесплатными инструментами, доступными на различных веб-сайтах.

ПРИМЕЧАНИЕ. Вопросы криптографии более детально рассмотрены в Домене 06. Если вы новичок в криптографии, имеет смысл еще раз вернуться к этому разделу после изучения Домена 06.

Использование EAP (Extensible Authentication Protocol - Расширяемый протокол аутентификации) и 802.1X (он будет рассмотрен чуть позже в этом разделе) для реализации функций аутентификации пользователей и взаимной аутентификации было интегрировано в новый стандарт WLAN 802.11i. Также в него был интегрирован MIC (Message Integrity Code - Код целостности сообщения) для выявления изменения битов в процессе передачи. Кроме того, в нем используется **TKIP** (Temporal Key Integrity Protocol - Протокол целостности временного ключа), который генерирует случайные значения, используемые в процессе шифрования, что существенно затрудняет взлом для злоумышленника. Для обеспечения еще более высокого уровня криптографической защиты, стандарт также включает в себя новый алгоритм AES (Advanced Encryption Standard), который используются в новых реализациях WLAN.

Далее в этом разделе мы рассмотрим эти различные компоненты, а также их взаимодействие друг с другом.

В чем заключаются проблемы WEP?

К сожалению, существует множество разновидностей атак, которые могут быть осуществлены на устройства и сети, использующие WEP. Реализации, основанные исключительно на первоначальном стандарте 802.11, подвержены множеству атак. Беспроводной трафик в них может быть легко перехвачен, данные могут быть изменены в процессе передачи без ведома получателя, злоумышленником могут быть установлены свои точки доступа (на которых пользователь может пройти аутентификацию и работать с ними, не зная при этом, что эта точка доступа установлена злоумышленником), зашифрованный беспроводной трафик можно быстро и легко расшифровать. К сожалению, эти уязвимости очень часто становятся открытой дверью в реальную проводную сеть, что может привести к гораздо более разрушительным атакам.

Многие производители продуктов WLAN разработали собственные средства и технологии обеспечения безопасности, позволяющие преодолеть недостатки и уязвимости 802.11, но реализации этих средств и технологий, как правило, похожи на некий «пластырь», за которым скрываются все те же глобальные проблемы, вызванные недостатками самого стандарта. К тому же при разработке производителями собственных решений, всегда возникают проблемы совместимости устройств различных производителей.

Стандарт 802.11i использует два различных подхода, которые обеспечивают гораздо большую безопасность и защиту, по сравнению с протоколом WEP, используемым в оригинальном стандарте 802.11. Повышение безопасности и защиты осуществляется за счет использования специальных протоколов, технологий и алгоритмов. Первым протоколом является TKIP, который обратно совместим со многими продуктами и сетями WLAN. В действительности TKIP использует ключевой материал WEP, который является исходными данными для создания новых динамических ключей. WEP использует алгоритм шифрования RC4, и текущая реализация алгоритма обеспечивает весьма слабую защиту. В процессе генерации ключей с использованием TKIP вводятся дополнительные сложности, которые существенно усложняют для нападающих взлом криптографических ключей. TKIP реализован рабочей группой IEEE, поэтому клиентам, достаточно просто получить обновления прошивки или программного обеспечения, без необходимости приобретения нового оборудования для дополнительной защиты.

802.11i предоставляет дополнительные возможности криптографической защиты с помощью алгоритма AES в режиме счетчика (counter mode) с CBC-MAC, который называется

протоколом ССМ (CCMP – CCM Protocol). Алгоритм AES больше подходит для беспроводных сетей, чем RC4, но он требует больше вычислительных ресурсов. AES не совместим с предыдущими продуктами WLAN, поэтому клиенты могут использовать такую конфигурацию, только если они еще не развернули у себя беспроводные сети.

ПРИМЕЧАНИЕ. Режим CBC (Cipher Block Chaining) описан в Домене 06.

Входящие в состав нового беспроводного стандарта алгоритмы, технологии и протоколы, достаточно сложны. Важно понимать как каждый компонент в отдельности, так и их совместную работу, позволяющую обеспечить более высокую степень защиты для будущих сред WLAN.

В следующем разделе мы рассмотрим различные процессы шифрования в стандарте 802.11i (CCMP и TKIP), а затем рассмотрим другие компоненты, входящие в состав этого стандарта и отсутствующие в первоначальном стандарте 802.11.

802.1X

Стандарт 802.11i можно представить в виде трех основных компонентов, содержащихся на двух отдельных уровнях. Нижний уровень содержит улучшенные алгоритмы шифрования (TKIP и CCMP), тогда как верхний уровень содержит 802.1X. Их совместная работа позволяет обеспечить больше уровней защиты, по сравнению с оригинальным стандартом 802.11.

Так что же такое 802.1X? Стандарт **802.1X** – это основанное на портах управление сетевым доступом, которое обеспечивает невозможность получения пользователем доступа к сети до тех пор, пока он не пройдет успешно аутентификацию. При этом пользователь не может использовать сетевые ресурсы, между беспроводным устройством и сетью не разрешается никакой трафик, кроме трафика аутентификации, до тех пор, пока пользователь не будет успешно аутентифицирован. В качестве аналогии можно привести пример цепочки на входной двери, которая позволяет открыть дверь лишь настолько, насколько необходимо для идентификации посетителя, и уже после того, как вы узнали посетителя, вы позволяете ему войти в ваш дом.

Стандарт 802.1X позволяет аутентифицировать *пользователя*, тогда как использование WEP позволяет аутентифицировать только *систему*. Аутентификация пользователя обеспечивает более высокую степень уверенности и защиты, чем аутентификация системы.

На самом деле, технология 802.1X обеспечивает платформу аутентификации и метод динамического распределения ключей шифрования. Тремя основными сущностями этой платформы являются: пользователь (беспроводное устройство), аутентификатор (точка доступа) и сервер аутентификации (как правило, сервер RADIUS). Если в сети нет сервера аутентификации, точка доступа может выполнять одновременно роль и аутентификатора, и сервера аутентификации.

Точка доступа, как правило, не обладает большой интеллектуальностью и работает в качестве посредника, передавая кадры между беспроводным устройством и сервером аутентификации. Обычно это хороший подход, поскольку он не требует существенных ресурсов для работы точки доступа, и точка доступа может управлять несколькими соединениями одновременно, а не аутентифицировать всех и каждого пользователя.

Точка доступа управляет всеми коммуникациями и позволяет беспроводным устройствам взаимодействовать с сервером аутентификации и проводной сетью только тогда, когда все шаги аутентификации выполнены успешно. Это означает, что беспроводное устройство не сможет передавать или получать трафик HTTP, DHCP, SMTP или любой другой тип трафика, пока пользователь не будет надлежащим образом авторизован. WEP не обеспечивает таких возможностей управления доступом. И это лишь один пример возможностей безопасности, обеспечиваемых стандартом 802.11i.

Другим недостатком оригинального стандарта 802.11 является отсутствие возможности взаимной аутентификации. При использовании одного только WEP беспроводное устройство может аутентифицироваться на точке доступа, но оно не может аутентифицировать сервер аутентификации. Соответственно, злоумышленником может быть установлена своя точка доступа для перехвата реквизитов доступа пользователей и другого трафика, при этом пользователи не будут знать, что они подверглись атаке. 802.11i решает эту проблему с помощью EAP. EAP обеспечивает возможность взаимной аутентификации между сервером аутентификации и беспроводным устройством, а также обеспечивает гибкость, позволяя пользователям аутентифицироваться с помощью паролей, токенов, одноразовых паролей, сертификатов, смарт-карт или Kerberos. Это позволяет беспроводным пользователям проходить аутентификацию с использованием существующей инфраструктуры и существующих технологий аутентификации. Беспроводное устройство и сервер аутентификации, соответствующие стандарту 802.11i, имеют различные модули аутентификации, которые поддерживаются 802.1X. Таким образом, 802.1X реализует платформу, которая позволяет сетевому администратору добавлять различные модули EAP. Две сущности (пользователь и аутентификатор) выбирают один из этих методов аутентификации (модулей EAP) в процессе «рукопожатия».

Стандарт 802.11i работает не со всем стеком протоколов, а затрагивает лишь канальный уровень модели OSI. Протоколы аутентификации работают на более высоком уровне, поэтому стандарт 802.11i не определяет детали протоколов аутентификации. Использование EAP позволяет различным производителям использовать различные протоколы. Например, Cisco использует платформу аутентификации исключительно на основе паролей, называемую LEAP (Lightweight Extensible Authentication Protocol). Другие производители, в том числе Microsoft, используют EAP и EAP-TLS, которые осуществляют аутентификацию с помощью цифровых сертификатов. Еще одним вариантом является PEAP (Protective EAP), в котором только сервер использует цифровой сертификат.

При использовании EAP-TLS, сервер аутентификации и беспроводное устройство для аутентификации обмениваются цифровыми сертификатами, выполняя шаги, похожие на те, которые выполняются при установлении соединения SSL между веб-сервером и браузером. После получения и проверки цифрового сертификата сервера, беспроводное устройство создает мастер-ключ, шифрует его на открытом ключе сервера и направляет серверу аутентификации. Теперь беспроводное устройство и сервер аутентификации имеют мастер-ключ, который они используют для генерации уникальных симметричных сеансовых ключей. Обе стороны используют сеансовые ключи для шифрования и расшифрования, а также создания безопасного канала между двумя устройствами.

Компании могут предпочесть использовать PEAP вместо EAP-TLS, поскольку при этом не возникает сложностей с установкой и поддержкой цифровых сертификатов для каждого беспроводного устройства. При использовании PEAP, пользователь беспроводного устройства посылает серверу пароль, а сервер аутентификации на беспроводное устройство – цифровой сертификат. В обоих случаях, должен быть реализован какой-либо вариант инфраструктуры PKI. Если компания еще не внедрила PKI, задача по его развертыванию только для беспроводной связи может стать очень трудоемкой и дорогостоящей.

Прежде чем покупать продукт WLAN, вы должны понять все требования и сложности каждого метода для уверенности, что вы точно знаете, что вы получите, и насколько это подходит для вашей среды.

Как было сказано ранее, Cisco использовала иной подход к аутентификации. Она реализовала LEAP, который основан исключительно на паролях. Для его использования не требуется PKI, а беспроводные устройства и серверы аутентифицируют друг с друга на основании предварительного обмена паролями.

Большую обеспокоенность в современных реализациях WLAN, использующих только WEP, вызывает то, что если отдельные беспроводные устройства украдены, они смогут легко пройти аутентификацию в проводной сети. 802.11i добавляет шаги, требующие проведения аутентификации пользователя в сети, а не только аутентификации беспроводного устройства. Используя EAP, пользователь должен отправить определенный набор реквизитов доступа, связанных с его личностью. При использовании только WEP, беспроводное устройство аутентифицируется, предоставляя симметричный ключ, который был заранее вручну введен в него. Поскольку пользователю при использовании WEP не нужно проходить аутентификацию, похищенное беспроводное устройство может позволить злоумышленнику получить доступ к вашей сети с драгоценными ресурсами.

Динамические ключи и использование Векторов инициализации

Тремя основными недостатками WEP являются: использование статических ключей шифрования, неэффективное использование векторов инициализации, а также отсутствие гарантий целостности пакета. Протокол WEP использует алгоритм RC4, который представляет собой поточный симметричный шифр. Слово *симметричный* означает, что отправитель и получатель должны использовать один и тот же ключ для шифрования и расшифрования. Стандарт 802.11 не предусматривает автоматизированного процесса обновления этих ключей, поэтому в большинстве сред симметричные ключи RC4 никогда не меняются. Как правило, все беспроводные устройства и точка доступа используют один и тот же ключ. Это все равно, что использовать во всей компании один и тот же пароль. Не очень хорошая идея. Таким образом, первая проблема – статический ключ шифрования WEP на всех устройствах.

Следующим недостатком является использование векторов инициализации (IV – initialization vector). Вектор инициализации – это числовое значение, которое используется вместе с симметричным ключом и алгоритмом RC4 для обеспечения большей случайности в процессе шифрования. Случайность – это чрезвычайно важно в шифровании, т.к. если что-то может позволить злоумышленнику понять, как работает процесс, это может дать ему возможность взломать используемый ключ шифрования. Ключ и значение IV передаются в алгоритм RC4 для генерации ключевого потока. Значения (единицы и нули) ключевого потока накладываются с помощью операции XOR на двоичные значения содержимого отдельных пакетов. В результате получается шифротекст или зашифрованные пакеты.

В большинстве реализаций WEP, в этом процессе постоянно используются одни и те же значения вектора инициализации, пока не изменится симметричный ключ (или общий секрет). Таким образом, отсутствует возможность обеспечить реальную случайность ключевого потока, генерируемого алгоритмом. Это позволяет нападающим провести «обратный инжиниринг» процесса для получения оригинального ключа шифрования, который затем может быть использован для расшифровки зашифрованного трафика.

Теперь можно перейти к третьему упомянутому выше недостатку – проблеме обеспечения целостности. Продукты WLAN, которые используют только стандарт 802.11, приводят к появлению уязвимостей, которые не всегда очевидны. Злоумышленник реально может изменить данные сетевых пакетов и одновременно с этим изменить Значение контроля целостности (ICV – Integrity Check Value), чтобы эти изменения не были замечены получателем. ICV работает аналогично функции CRC: отправитель рассчитывает ICV и вставляет его в заголовок кадра. Получатель вычисляет свое собственное значение ICV и сравнивает его с ICV в заголовке кадра. Если значения ICV совпадают, получатель считает, что в кадр не был изменен в процессе передачи. Если значения ICV отличаются, это говорит получателю о том, что произошли изменения, и получатель уничтожает этот кадр. В WEP существуют некоторые особенности, из-за которых получатель не может выявить изменения в кадре, поэтому в нем не существует реальных гарантий целостности.

Таким образом, проблемами стандарта 802.11 являются плохая аутентификация, статические

ключи WEP, которые могут быть легко получены нападающими, повторяющиеся значения вектора инициализации, не обеспечивающие необходимую степень случайности ключевого потока, а также недостатки контроля целостности данных. Использование технологии 802.1X в новом стандарте 802.11i обеспечивает управление доступом, ограничивая доступ к сети до момента завершения полной аутентификации и авторизации, и служит надежной платформой для аутентификации, позволяющей подключать к ней различные модули EAP. Эти две технологии (802.1X и EAP) работают вместе, обеспечивая взаимную аутентификацию между беспроводным устройством и сервером аутентификации. Но как при этом обстоят дела со статическими ключами, значениями векторов инициализации и проблемами целостности?

TKIP учитывает недостатки WEP, относящиеся к использованию WEP статичных ключей и использованию ненадежных значений векторов инициализации. Существуют две очень полезные, мощные и легкие в использовании программы, которые могут быть использованы для взлома шифрования WEP: AirSnort и WEP-Crack. Они эксплуатируют указанные недостатки, а также неэффективное использование алгоритма распределения ключей, применяемое в протоколе WEP. Если компания применяет продукты, которые реализуют шифрование только с использованием WEP, без каких-либо сторонних решений для шифрования (например, VPN), указанные выше программы могут без труда взломать ее зашифрованный трафик в течение очень короткого времени (от нескольких минут до нескольких часов). При этом взлом шифрования WEP гарантирован независимо от того, какие используются ключи (40-битные или 128-битные). Это одна из самых серьезных и опасных уязвимостей, связанных с оригинальным стандартом 802.11.

Использование TKIP позволяет выполнять ротацию ключей шифрования для противодействия атакам такого типа. Этот протокол увеличивает длину значения вектора инициализации и гарантирует, что для шифрования каждого кадра будет использоваться новое значение вектора инициализации. Значение вектора инициализации комбинируется с MAC-адресом отправителя и оригинальным ключом WEP, поэтому даже в случае использования статического ключа WEP, реальный ключ шифрования будет отличаться для каждого кадра (ключ WEP + значение вектора инициализации + MAC-адрес = новый ключ шифрования). Это обеспечивает значительно большую случайность процесса шифрования, что позволяет противодействовать криптоанализу и атакам на криптосистемы. Изменение значений вектора инициализации и получаемых в результате ключей, делает ключевой поток менее предсказуемым, что существенно затрудняет злоумышленнику выполнение «обратного инжиниринга» и получения оригинального ключа.

Кроме того, TKIP решает проблему целостности с помощью MIC вместо использования функции IVC. Если вы знакомы с работой Кода аутентификации сообщения (MAC – Message Authentication Code), то вам будет проще понять это, поскольку это то же самое. Функция хэширования использует симметричный ключ, что очень похоже на работу CRC, но более стойкую. Использование MIC вместо ICV позволяет получателю контролировать изменения в кадре, которые могли произойти во время передачи. Отправитель и получатель рассчитывают свои собственные независимые значения MIC. Если у получателя получилось значение MIC, отличающееся от указанного в заголовке кадра, кадр считается скомпрометированным и уничтожается.

ПРИМЕЧАНИЕ. Эти криптографические концепции и технологии рассматриваются в Домене 06.

Ответ на все наши молитвы?

До настоящего времени, если компании требовался более высокий уровень защиты, чем мог предоставить первоначальный стандарт 802.11, она была вынуждена устанавливать межсетевой экран между точкой доступа и проводной сетью. Также, на беспроводные устройства иногда устанавливали программное обеспечение VPN для реализации дополнительного, более стойкого шифрования. Помимо этого, как было отмечено ранее,

различные поставщики предложили свои собственные решения по обеспечению безопасности. Но сейчас мы надеемся, что к 802.11i не потребуется применять многочисленные заплатки и надстройки. Мы рассчитываем, что безопасность будет обеспечиваться самой технологией.

Так все-таки, дает ли использование EAP, 802.1X, AES и TKIP реально безопасную и доверенную реализацию WLAN? Возможно, но мы должны хорошо понимать, как все это правильно использовать. TKIP была создана для того, чтобы по-быстрому исправить наиболее критичные проблемы WEP. Использование TKIP не перестраивает сам стандарт 802.11, поскольку и TKIP, и WEP по-прежнему основываются на алгоритме RC4, являющимся не самым лучшим вариантом для таких технологий. Применение AES ближе к реальной перестройке стандарта, но оно не обеспечивает обратной совместимости с более ранними реализациями 802.11. Кроме того, следует понимать, что использование всех этих новых компонентов их объединение с уже используемыми компонентами 802.11 увеличивает сложности внедрения новых компонентов и добавляет шаги в этот процесс. А безопасность и сложность, как правило, не совместимы. Наибольшая безопасность достигается, как правило, самыми простыми и элегантными решениями, позволяющими гарантировать, что все точки доступа учтены и защищены. Новые технологии дают больше гибкости поставщикам при выборе механизмов аутентификации пользователей и серверов аутентификации, но это также может привести к проблемам совместимости устройств различных производителей, поскольку не все производители будут выбирать одни и те же методы. Это означает, что если компания покупает одну точку доступа у производителя А, а затем беспроводные сетевые карты у производителей В и С, могут возникнуть проблемы с их взаимодействием.

Получается, что вся эта работа была проделана зря? Нет. Стандарт 802.11i обеспечивает гораздо более высокий уровень защиты и безопасности, чем имел WEP когда бы то ни было. Разработавшая его рабочая группа была составлена из очень грамотных людей и нескольких крупных и влиятельных компаний, которые оказали помощь при разработке новых решений. Однако клиенты, покупающие эти новые продукты, должны понимать, что потребуется от них после оформления заказа. Например, при использовании EAP-TLS, каждое беспроводное устройство должно иметь собственный цифровой сертификат. Ваши нынешние беспроводные устройства могут работать с сертификатами? Как нужно правильно установить сертификаты на все беспроводные устройства? Как поддерживать сертификаты? Как организовать проверку списка отозванных сертификатов (CRL – Certificate revocation list) устройствами и сервером аутентификации, чтобы они могли убедиться, что сертификат не был отозван? Что делать, если обнаружен установленный злоумышленником сервер аутентификации или точка доступа с действительным цифровым сертификатом? Беспроводное устройство просто проверит этот сертификат, и будет считать, что это подлинный сервер, с которым оно собиралось взаимодействовать. А если центр сертификации будет скомпрометирован, то будет скомпрометирована и вся инфраструктура EAP-TLS, как и в любой среде PKI.

Оригинальный стандарт 802.11 получил очень много негативных отзывов, поэтому новая рабочая группа приложила все усилия, чтобы в новой версии стандарта были учтены все известные проблемы и расставлены все точки над «i». Но время покажет, насколько ей это удалось. Этот новый улучшенный стандарт обеспечивает гораздо больше безопасности, но существуют две вещи, которые нужно учитывать при внедрении новых продуктов. Во-первых, производители должны правильно интерпретировать стандарт и придерживаться его, для реального обеспечения уровня гарантий и безопасности, обещанного стандартом. Во-вторых, клиентам нужно объяснить все особенности новых продуктов и обучить их использованию этих продуктов, чтобы клиенты понимали, что они покупают и внедряют в свою среду. Слишком часто продукты и технологии внедряются компанией без необходимых знаний о том, как их правильно внедрять, тестировать, как обеспечивать их безопасность.

Истинная безопасность основывается на обучении, знаниях и опыте. Таким образом, конечной целью является получение всеобъемлющего стандарта, который не только применяется при производстве безопасной продукции, но и позволяет клиенту узнать все необходимое об этом новом продукте, чтобы реально достичь необходимого уровня защиты. Если хотя бы один из этих компонентов не выполнен, это может привести к хаосу.

Новые продукты. В настоящее время уже разрабатываются новые продукты WLAN, реализующие эти новые беспроводные стандарты. Многие из них позволяют использовать как TKIP, так и AES. Первый пригодится компаниям для обеспечения обратной совместимости с более ранними реализациями WLAN, уже внедренными у них. А вторым смогут воспользоваться компании, впервые разворачивающие свои беспроводные сети. Клиентам следует обращать внимание на наличие у приобретаемых беспроводных продуктов сертификации Wi-Fi Alliance, который проводит оценку систем на соответствие стандарту 802.11i. Таблица 5-12 показывает характеристики различных компонентов беспроводной безопасности, используемых в настоящее время.

	WEP	WPA	WPA2
Управление доступом	802.1X	802.1X или предварительно установленный общий ключ	802.1X или предварительно установленный общий ключ
Аутентификация	Методы EAP	Методы EAP или предварительно установленный общий ключ	Методы EAP или предварительно установленный общий ключ
Шифрование	WEP	TKIP (RC4)	CCMP (AES Counter Mode)
Целостность	Нет	Michael MIC	CCMP (AES CBC-MAC)

Таблица 5-12. Характеристики современных компонентов безопасности беспроводных технологий

802.11j

Многие страны разрабатывают свои собственные беспроводные стандарты, что неизбежно приводит к массе проблем при совместной работе. Это мешает как покупателю (он не может использовать определенные продукты), так и повышает себестоимость для производителя (у него появляется еще более длинный список спецификаций, которые он должен учесть, если хочет продавать свой продукт в различных странах). Если производитель не сможет учесть отдельные спецификации, соответствующий круг покупателей будет для него недоступен. Рабочая группа 802.11j работает над объединением множества различных стандартов и оптимизации их разработки, чтобы обеспечить лучшее взаимодействие между стандартами разных стран.

802.11n

Целью 802.11n WWiSE (World Wide Spectrum Efficiency) является попытка заменить имеющийся набор различных технологий Wi-Fi. 802.11n разработан для обеспечения более высокой скорости (до 100 Мбит/с), он работает в частотном диапазоне 802.11a (5 ГГц). Он обеспечивает обратную совместимость с текущими стандартами Wi-Fi, объединяя современные технологии. Этот стандарт позволяет использовать концепцию **MIMO** (multiple input, multiple output), обеспечивающую повышение пропускной способности. Для этого требуется использовать две приемных и две передающих антенны для параллельной широкополосной рассылки с использованием канала 20 МГц.

802.16

Рассмотренные ранее беспроводные стандарты являются WLAN-ориентированными. 802.16 – это беспроводной стандарт MAN, который позволяет беспроводному трафику покрывать гораздо большие географические площади. Эту технологию также называют беспроводным широкополосным доступом.

802.15

Этот стандарт работает со значительно меньшими по географическим размерам сетями, которые называют **WPAN** (wireless personal area network). Эта технология позволяет

соединить локальные устройства, например, обеспечить связь между компьютером и КПК, сотовым телефоном и гарнитурой, и т.п. Целью здесь, как и в других беспроводных технологиях, является обеспечение возможности беспроводной передачи данных.

Bluetooth

Беспроводная технология **Bluetooth** на самом деле является частью стандарта 802.15. Она позволяет передавать данные со скоростью 1-3 Мбит/с и работает в радиусе около 10 метров. Например, если у вас есть сотовый телефон и КПК, которые поддерживают Bluetooth и имеют функционал календаря, вы можете организовать автоматическую синхронизацию календаря на обоих этих устройствах без необходимости их физического соединения. При этом когда вы добавите новую встречу или задачу на сотовом телефоне, вам достаточно просто положить КПК рядом с телефоном, чтобы эта информация была перенесена на КПК. КПК «почувствует» наличие поблизости другого устройства и попытается установить сетевое соединение с ним. Когда соединение будет создано, произойдет синхронизация календаря между двумя устройствами. Это звучит прекрасно и большинство портативных устройств уже могут работать таким образом. Bluetooth работает в частотном диапазоне других 802.11-устройств (2.4 ГГц).

Однако при передаче незащищенных данных через Bluetooth возникают реальные риски безопасности, особенно в общественных местах, т.к. любое устройство в определенном радиусе может перехватить передаваемые таким способом данные.

Другая атака, которой подвержен Bluetooth, называется **Bluejacking**. При этой атаке некто отправляет незатребованное сообщение на устройство с включенным Bluetooth. Атакующий находит такое устройство (телефон, КПК, ноутбук) и отправляет ему сообщение. Чаще всего атакующий пытается отправить некий текст в формате визитной карточки, которая добавится в список контактов жертвы. Контрмерой против этой атаки является перевод Bluetooth-устройства в невидимый режим, чтобы другие не могли найти и идентифицировать устройство. Если вы получили подобное сообщение, оглянитесь вокруг. Bluetooth работает на расстоянии всего около 10 метров, поэтому сообщение пришло от кого-то рядом с вами.

ПРИМЕЧАНИЕ. Некоторые из этих стандартов пока еще находятся в стадии разработки, и не все рассмотренные стандарты и технологии сохранятся в будущем. Пройдет время, прежде чем они будут реализованы производителями и внедрены покупателями. Беспроводные технологии пока еще остаются достаточно «незрелыми», они страдают ужасными болезнями, очень сложно определить, за каким стандартом будущее. Но беспроводные технологии подобны пожару, все ими интересуются, они растут и поглощают рынки.

Ссылки по теме:

- IEEE 802 Working Groups
- IEEE Wireless Standards Zone
- “802.11 Alphabet Soup,” by Jim Geier, Small Business Computing (Aug. 6, 2002)
- “Exploiting and Protecting 802.11b Wireless Networks,” by Craig Ellison, ExtremeTech (Sept. 4, 2001)
- Official Bluetooth Special Interest Group (SIG) web site

12.4. WAP

WAP (Wireless Application Protocol - Протокол приложений для беспроводной связи) сам по себе не является стандартом. Фактически WAP – это некий маркетинговый и промышленный стек протоколов. В чем разница? Когда управляющая организация, такая как IEEE, решает, что необходима новая технология, она создает рабочую группу для разработки соответствующего стандарта. Этот стандарт используется в качестве “эскиза” всеми производителями, которые хотят разработать соответствующую технологию. Если

общепризнанного стандарта не существует, организации могут собраться вместе и самостоятельно разработать такой эскиз для новой технологии, которому все они будут следовать.

WAP предоставляет базовую архитектуру для беспроводных устройств, позволяющую им взаимодействовать через Интернет. Он позволяет беспроводным устройствам отправлять и принимать данные (в дополнение к голосовым данным), обеспечивает возможность подключения мобильных телефонов и КПК к контент-провайдерам в Интернете, позволяет загружать биржевые котировки, информацию о погоде, получать доступ к электронной почте и т.д. Также, это открывает дверь для создания новых видов беспроводных устройств.

Многие из этих беспроводных устройств не имеют ресурсов полноценного компьютера (процессорной мощности, памяти, объема дискового пространства и т.п.). Поэтому эти устройства не могут использовать тот же стек протоколов (как в TCP/IP), который используют более мощные системы, они не могут запускать те же приложения. Стек протоколов WAP разработан для использования системами с ограниченными ресурсами, для приложений, созданных для работы в новой среде.

WAP – это набор коммуникационных протоколов, используемых для стандартизации способов взаимодействия беспроводных устройств друг с другом и с сетью Интернет. Модель WAP содержит протоколы, выполняющие функции, аналогичные протоколам стека TCP/IP. WAP реализует способ представления веб-страниц. Персональные компьютеры и серверы используют HTML или XML для представления веб-материалов и JavaScript для выполнения фоновой обработки. Для выполнения аналогичных задач, WAP использует XML-совместимый язык WML (Wireless Markup Language - Язык беспроводной разметки) и WMLScript. WAP имеет собственные сеансовый и транспортный протоколы, а также протокол безопасности транспортного уровня *WTLS* (Wireless Transport Layer Security - Беспроводной протокол защиты транспортного уровня), похожий на TLS и SSL. На беспроводных устройствах установлен WAP-микробраузер, который отображает веб-страницы пользователю.

Эта технология позволяет устройствам проверять электронную почту, принимать голосовые сообщения, календарь и множество других видов данных без необходимости физического подключения к сети. Пользователь может участвовать в биржевых торгах со своего КПК, читая книгу в автобусе. Ему нужно нажать всего несколько кнопок, чтобы продать свои акции в случае предстоящего падения рынка. Также пользователи могут со своего мобильного телефона управлять своими банковскими счетами, получать ежедневные новости и т.п. Беспроводные устройства могут использоваться для работы с любыми организациями, предоставляющими информацию и/или возможности выполнения некоторых операций через Интернет.

Поскольку эти устройства используют иной набор протоколов, необходим шлюз для трансляции данных между WAP- и интернет-протоколами, а также различными типами приложений, как показано на рисунке 5-58. Этот шлюз могут предоставить провайдеры услуг, одновременно с другими услугами, которые они предоставляют пользователям и компаниям для доступа в Интернет.



Рисунок 5-58. Шлюз WAP необходим для трансляции протоколов WAP в Интернет-протоколы

Данные, поступающие с беспроводного мобильного устройства, могут быть зашифрованы с помощью WTLS. Эти данные должны быть транслированы шлюзом в TLS или SSL. Поскольку беспроводные устройства используют для передачи данных Интернет, а Интернет не понимает WTLS, протокол WTLS должен быть транслирован в такой протокол, который может применяться в сети Интернет. Эта трансляция выполняется на шлюзе провайдера услуг.

При этом возникает проблема безопасности, поскольку эти данные должны быть расшифрованы на оборудовании провайдера и повторно зашифрованы с помощью SSL или TLS. Это означает, что данные какое-то время (1-2 секунды) не защищены. Это называют **«окном» в WAP** (gap int the WAP), оно является одной из проблем, которую специалисты по безопасности должны учитывать.

Работа WTLS похожа на SSL/TLS, он шифрует данные и обеспечивает аутентификацию между взаимодействующими устройствами. WTLS имеет три класса, которые определяют процесс аутентификации между беспроводным устройством и сервером-шлюзом:

- **Класс 1.** Анонимная аутентификация. Беспроводное устройство и сервер не аутентифицируют друг друга.
- **Класс 2.** Серверная аутентификация. Сервер аутентифицирует беспроводное устройство.
- **Класс 3.** Двухсторонняя аутентификация. Сервер и беспроводное устройство аутентифицируют друг друга.

В большинстве случаев сервер-шлюз указывает тип аутентификации, который должен использоваться, однако при этом беспроводное устройство должно быть правильно настроено, чтобы оно реально могло работать с этим типом аутентификации.

12.5. i-Mode

WAP и *i-Mode* являются двумя основными протоколами, используемыми сотовыми телефонами и КПК для беспроводной передачи данных сети Интернет. WAP разработан консорциумом компаний для корпоративного использования. i-Mode разработан японской компанией NTT DoCoMo и в большей степени ориентирован на предоставлении развлекательных услуг, чем на корпоративном применении.

i-Mode работает с урезанной версией HTML, названной cHTML (Compact HTML), тогда как WAP использует WML – язык разметки, основанный на XML. Обе эти технологии (i-Mode и WAP) используют архитектуру, состоящую из сервера, содержащего контент, шлюза и пользовательского терминала. Сервером может быть WAP- или HTTP-сервер, а шлюз выполняет трансляцию протоколов, чтобы они могли использоваться в Интернете. Пользовательский терминал – это мобильный телефон или иное мобильное устройство, имеющее микробраузер, интерпретирующий и представляющий веб-контент пользователю.

i-Mode имеет огромную популярность в Японии, сейчас активно распространяется в Азии и некоторых частях Европы. Современная (вторая) версия WAP используется в основном в Северной Америке. Две эти новые родственные технологии продолжают распространяться, т.к. мы хотим иметь доступ в Интернет всегда и везде.

12.6. Безопасность мобильных телефонов

Технологии, используемые на мобильных телефонах и КПК, также имеют проблемы безопасности. Политики безопасности большинства компаний запрещают использование устройств таких видов. Все было прекрасно, пока телефоны были просто телефонами, но сегодня это маленькие компьютеры, которые могут подключаться к другим компьютерам и сетям, поэтому они являются новой точкой входа для злоумышленников. Поскольку мобильные устройства являются компьютерами со своими операционными системами, многие сотрудники хранят на них критичные данные (например, пароли, контакты, файлы компании и т.п.). Причем обычно эта информация хранится в открытом виде. Большинство сотовых телефонов имеют встроенные камеры, и компании должны понимать, что это новый способ, с помощью которого критичные данные и действия могут быть «скопированы» и переданы во внешний мир. Каждая компания должна учесть эти новые технологии и источники угроз в своей политике и программе безопасности.

Другой распространенной проблемой безопасности, связанной с мобильными телефонами, является то, что мобильные телефоны должны пройти аутентификацию на базовой станции, прежде чем им будет разрешено делать звонки, однако аутентификация самой базовой станции мобильным телефоном не выполняется. Это открывает дверь злоумышленникам для установки поддельных базовых станций. Когда сотовый телефон отправляет аутентификационные данные базовой станции злоумышленника, тот может перехватить их и использовать в дальнейшем для прохождения аутентификации и получения несанкционированного доступа к сотовой сети.

Сотовые телефоны **клонировуют** на протяжении уже многих лет и эту деятельность нельзя искоренить за короткое время. Обычный сотовый телефон может быть украден и затем перепрограммирован злоумышленником, либо использован для получения чужих учетных данных. Это часто делают криминальные организации и те, кто не хочет, чтобы их информацию читали государственные службы. Телефоны GSM (Global System Mobile) используют чиповую SIM-карту (Subscriber Identity Module - Модуль идентификации абонента), на которой хранятся данные аутентификации, номер телефона, сохраненные сообщения и т.п. Прежде чем GSM-телефон сможет получить доступ к сотовой сети, в телефон должна быть установлена SIM-карта. Злоумышленник может клонировать SIM-карту, что позволит ему делать звонки со своего сотового телефона за счет владельца SIM-карты.

Нужно понимать, что сотовые телефоны передают данные по радиоволнам и затем эти данные попадают в проводную сеть телефонной компании или провайдера услуг. Таким образом, часть расстояния трафик передается без проводов, а оставшаяся часть расстояния может быть проводной. Поэтому, если кто-то шифрует свои данные и отправляет их на свой сотовый телефон, обычно эти данные остаются зашифрованными только на этапе передачи по беспроводной части сети. Как только данные попадают в проводную часть сети, они расшифровываются и дальше передаются в открытом виде. Таким образом, при передаче зашифрованных данных на сотовый телефон или КПК, они не обязательно остаются зашифрованными на всем протяжении своего пути.

К сожалению, атаки на сотовые телефоны никогда не прекратятся. Как только мы внедрим дополнительные защитные меры, плохие парни найдут новые пути эксплуатации уязвимостей, о которых мы до сих пор не думали. Это все та же игра в кошки-мышки, происходящая в обычных сетях по всему миру. Но в отношении атак на сотовые телефоны, основной проблемой является то, что они обычно не включены в корпоративную программу безопасности и даже не считаются угрозой. Это позволяет производить все больше атак через эти новые точки доступа, разрабатывать новые вирусы, вмешиваться во взаимодействие сотовых телефонов и КПК с корпоративной сетью. Ниже перечислены некоторые из проблем, с которыми сталкиваются компании при использовании сотовых телефонов:

- Могут быть созданы фальшивые базовые станции
- Могут быть похищены конфиденциальные данные
- Используется функционал камеры
- Доступ в сеть Интернет в обход межсетевых экранов компании
- SMS-спам
- Может быть загружен вредоносный код
- Шифрование может быть слабым, при этом оно может не обеспечиваться на всей протяженности маршрута передачи данных

ПРИМЕЧАНИЕ. UMTS (Universal Mobile Telecommunications System - Универсальная мобильная телекоммуникационная система) – это новый стандарт для третьего поколения мобильных коммуникаций.

Существуют межсетевые экраны для сотовых телефонов, позволяющие обеспечить защиту от следующих видов проблем безопасности:

- Передача сигналов VoIP (VoIP signaling) и DoS-атаки
- Мошенничество с оплатой
- Неправильное использование полосы пропускания VoIP
- Заражение вирусами
- Ограничение передачи файлов
- Атаки на голосовую почту и АТС
- Несанкционированные подключения сотрудников
- Wardialing

Внедрение такого межсетевого экрана сегодня редкость, но вполне вероятно, что их количество будет расти, т.к. все больше компаний будут сталкиваться с перечисленными выше видами угроз.

Ссылки по теме:

- WAP 2.0 Forum Releases

12.7. Вардрайвинг

Очень распространенной атакой на беспроводную сеть является *вардрайвинг* (war driving), при которой один или несколько человек ходят вокруг или ездят на машине с включенным беспроводным оборудованием и специальным программным обеспечением, которое позволяет выявить точки доступа и взломать их. Обычно это делается с помощью ноутбука и перемещения на машине вокруг здания, в котором установлено оборудование WLAN, однако сегодня даже КПК может применяться для атак этого типа.

Для перехвата информации (мониторинга) точек доступа могут использоваться, например, такие программы, как Kismet и NetStumbler. Когда такая программа находит сигнал точки доступа, она заносит в журнал название сети, SSID, MAC-адрес точки доступа, наименование ее производителя, прослушиваемый канал, силу сигнала, соотношение сигнал/шум, факт использования WEP. Атакующий перемещается вокруг на своей машине с ноутбуком, на котором запущена программа NetStumbler, производящая активный поиск точек доступа. Обычно, она может обнаружить точки доступа в радиусе до 100 метров, но с более мощной антенной атакующий может находить точки доступа и гораздо дальше. NetStumbler рассылает тестовые запросы каждую секунду, ожидая ответа точки доступа. Если на точке доступа используется WEP, для взлома и перехвата ключей могут использоваться такие программы, как Aircrack-ng, Aircrack-ng или WEP-Crack.

ПРИМЕЧАНИЕ. Находчивые люди придумали, как можно использовать металлическую банку от чипсов Pringles или пивную банку в качестве недорогой антенны для подобных атак. Такие «антенны» могут легко найти точку доступа за полтора километра.

Ниже приведены некоторые лучшие практики, относящиеся к внедрению WLAN:

- *Включайте WPA или WPA2*, реализованные в стандарте 802.11i.
- *Изменяйте установленный по умолчанию SSID*. Каждая точка доступа имеет настроенный по умолчанию SSID.
- *Блокируйте широковещательную рассылку SSID на точке доступа*. Большинство точек доступа позволяет отключить это.
- *Используйте дополнительный уровень аутентификации (RADIUS, Kerberos)*. Прежде чем получить доступ в сеть, пользователь должен пройти аутентификацию.
- *Физически размещайте точку доступа в центре здания*. Точка доступа имеет определенную зону покрытия. Лучше, чтобы сигнал точки доступа не выходил далеко за пределы здания.
- *Логически размещайте точку доступа в DMZ, устанавливайте межсетевой экран между этой DMZ и внутренней сетью*. Включите на межсетевом экране функции проверки трафика перед его допуском в проводную сеть.
- *Внедрите VPN для использования беспроводными устройствами*. Это добавит еще один уровень защиты данных в процессе их передачи.
- *Настройте точку доступа для допуска в сеть только известных MAC-адресов*. Разрешите выполнение аутентификации только для известных устройств. Но не забывайте, что MAC-адреса передаются открытым текстом, позволяя атакующим перехватить их и использовать для несанкционированного доступа в сеть.
- *Присваивайте статические IP-адреса беспроводным устройствам, отключайте DHCP*. Если атакующий получит доступ к сети, в которой включен DHCP, он легко получит правильный рабочий адрес для работы в ней.

- *Проводите тесты на проникновение в WLAN.* Используйте описанные в этом разделе инструменты для выявления точек доступа и пытайтесь взломать применяющиеся в них схемы шифрования.
- *Переходите на продукты, соответствующие стандарту 802.11i.*

12.8. Спутники

В настоящее время спутники используются для организации беспроводной связи между различными местами. Чтобы между двумя местами могла быть установлена связь через спутник, эти места должны находиться в зоне прямой видимости и в **зоне обслуживания** (footprint) спутника. Отправитель информации (наземная станция) модулирует данные в радиосигнал, который передает на спутник. Транспондер на спутнике принимает сигнал, усиливает его и пересылает получателю. Получатель должен иметь антенну, похожую на круглую тарелку, такие антенны часто можно увидеть на крышах высоких зданий. На такой антенне установлен один или несколько микроволновых приемников, в зависимости от количества спутников, с которых планируется принимать сигнал.

Спутники обеспечивают широкополосную связь, обычно используемую для телевидения и доступа компьютеров в Интернет. Для телевидения настраивается однонаправленная передача. Если требуется доступ в Интернет, необходима двухсторонняя сеть. Доступная ширина полосы пропускания зависит от антенны, типа терминала и сервисов, предоставленных провайдером услуг. Приложения, чувствительные ко времени передачи данных, при использовании спутниковой связи могут страдать от задержек, вызванных временем передачи данных до спутника и обратно. Используемые для таких целей спутники размещают на низкой орбите, что обеспечивает минимизацию расстояния от наземной станции до спутника. Кроме того, это позволяет использовать приемник меньшего размера, делая спутники на низкой орбите идеальным средством для двухстороннего пейджинга, международной сотовой связи, телевидения, предоставления доступа в Интернет.

Размер зоны обслуживания зависит от типа спутника. Она может быть большой, как страна, или охватывать только несколько сотен метров в диаметре. Зона обслуживания покрывает определенную область на земле лишь несколько часов или даже меньше, поэтому провайдер услуг обычно имеет большое количество спутников, распределенных таким образом, чтобы они постоянно покрывали нужную область.

В большинстве случаев для спутникового доступа в Интернет применяется гибридная система, в которой для отправки запросов с компьютера пользователя используется обычная телефонная линия и модем, а спутниковая связь используется только для приема отправленных пользователю данных, как показано на Рисунке 5-59.

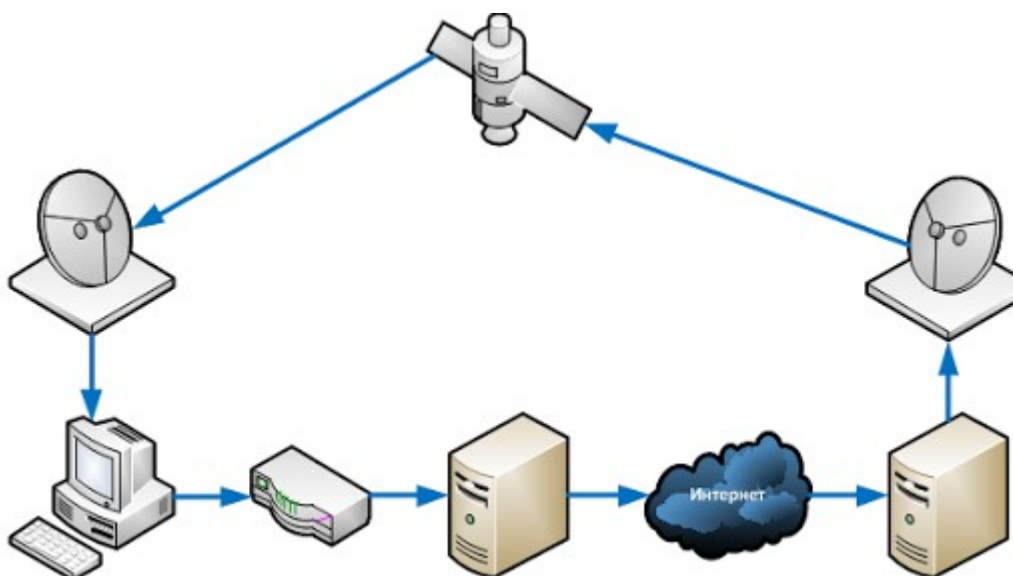


Рисунок 5-59. Широковещательная спутниковая передача данных

12.9. Беспроводные коммуникации 3G

Третьим поколением мобильных технологий являются технологии, которые часто называют *широкополосными беспроводными технологиями*. Первое поколение использовало аналоговую передачу только голосовых данных через сети с коммутацией каналов. Второе поколение позволяло передавать голос и данные в цифровом виде между беспроводным устройством, таким как сотовый телефон, и контент-провайдером. Второе поколение обеспечивало скорость передачи данных около 19,2 кбит/с. TDMA (Time Division Multiple Access - Множественный доступ с разделением по времени), CDMA (Code Division Multiple Access - Множественный доступ с кодовым разделением), GSM (Global System for Mobile Communications) и PCS (Personal Communications Services) – все они являются мобильными технологиями 2G. Это поколение технологий может передавать данные через сети с коммутацией каналов, поддерживает шифрование данных, передачу факсов и коротких сообщений (SMS).

3G – это большой шаг вперед, который обеспечил высокоскоростной мобильный доступ в Интернет на базе сервисов IP. Современные сотовые телефоны могут отображать веб-страницы, выполнять функции клиента электронной почты, проигрывать музыку и видео, позволяют играть в игры. Они имеют операционную систему, приложения, ну и, конечно, телефон. Технологии 3G обеспечивают скорость передачи данных до 384 кбит/с при движении и до 2 Мбит/с в стационарном режиме.

В Таблице 5-13 приведены различные характеристики отдельных поколений мобильных технологий.

	1G	2G	3G	4G
Диапазон	900 МГц	1 800 МГц	2 ГГц	40 ГГц и 60 ГГц
Тип мультимплексирования	Аналоговый FDMA	TDMA	CDMA	OFDM
Поддержка голосовых функций	Основные функции телефонии	Определитель номера, голосовая почта	Конференц-связь, видео низкого качества	Телеприсутствие, видео высокого качества
Передача сообщений	Нет	Только текст	Графика и форматированный текст	Передача полностью унифицированных сообщений
Поддержка передачи данных	Нет	Коммутация каналов (коммутация пакетов в 2,5G)	Коммутация пакетов	IPv6
Теоретическая скорость передачи данных	Нет	14,4 кбит/с (около 115 кбит/с в 2,5G)	2 Мбит/с (10 Мбит/с в 3,5G), 14 Мбит/с с использованием HSPA+	100 Мбит/с
Реальная скорость передачи данных	2,4 кбит/с	9,6 кбит/с (около 40 кбит/с в 2,5G)	64 кбит/с	Неизвестно
Интерфейс с другими устройствами	Акустический адаптер	Последовательный кабель RS232 или IrDA	IEEE 802.11 или Bluetooth	Бесконтактное соединение различными способами
Период использования	1980-1994	1995-2001	2002-2005	2006-2010

Таблица 5-13. Основные характеристики различных поколений телефонии

ПРЕДУПРЕЖДЕНИЕ. Очень важно при сдаче экзамена CISSP более детально разбираться в технологиях беспроводной передачи данных. Для получения дополнительной информации вы можете ознакомиться со статьей « Широкополосные беспроводные коммуникации».

Ссылки по теме:

- WarDriving.com
- Wardrive.net wardriving tools, software, and utilities
- “War Driving by the Bay,” by Ken Poulsen, SecurityFocus (April 12, 2001)
- “Packet Leashes: A Defense Against Wormhole Attacks in Wireless Networks,” by Yih-Chun Hu, Adrian Perrig, and David B. Johnson (IEEE Infocom 2003 conference)

Поколения мобильных технологий. Как и многие другие технологии, технологии мобильной связи прошли через несколько различных поколений.

Первое поколение (1G):

- Аналоговые сервисы
- Только голосовые сервисы

Второе поколение (2G):

- В основном голос, низкоскоростная передача данных (коммутация каналов)
- Телефоны уменьшились в размерах
- Добавилась функциональность электронной почты, передачи SMS, определитель номера

Поколение 2,5 (2,5G):

- Более широкая полоса пропускания, по сравнению с 2G

- Технология «всегда включен» для электронной почты и обмена сообщениями

Третье поколение (3G):

- Интеграция голоса и данных
- Технология с коммутацией пакетов вместо коммутации каналов

13. Руткиты

Часто после успешного взлома компьютера пользователя, хакер пытается повысить свои привилегии для получения прав уровня администратора или «суперпользователя» (root). Работа в контексте безопасности таких привилегированных пользователей позволяет злоумышленнику выполнять максимально опасные действия. После получения такого уровня доступа, злоумышленник может загрузить на взломанный компьютер комплект инструментальных средств, называемых **руткитом** (rootkit). Первая вещь, которую обычно устанавливает злоумышленник – это бэкдор-программа, которая позволяет злоумышленнику войти во взломанную систему в любое время без какой-либо аутентификации (бэкдор – это просто сервис, который прослушивает определенный порт). Другие инструменты, входящие в руткит, могут быть различными. Как правило, в состав руткита входят утилиты, позволяющие скрыть деятельность атакующего. Например, каждая операционная система имеет стандартные утилиты, с помощью которых администратор (или root) может выявить присутствие руткита, сниффера и бэкдора. Хакер заменяет эти стандартные утилиты их «новыми версиями», которые имеют те же названия. Они называются «троянскими программами», поскольку помимо своей основной функциональности, они выполняют какую-либо дополнительную вредоносную деятельность в фоновом режиме. Например, в Unix-системе утилита `ps` (состояние процесса) выводит список всех процессов, запущенных в системе, и их состояние. Утилита `top` показывает список процессов, их состояние, а также объем памяти, используемый каждым процессом. Большинство руткитов имеют троянские программы, которые заменяют эти утилиты, иначе root может запустить `ps` или `top` и увидеть, что работает бэкдор, и выявить факт атаки. Но когда root запускает троянскую версию этой утилиты, она перечисляет все процессы, за исключением процесса бэкдора. Большинство руткитов имеют снифферы, позволяющие перехватывать данные, которые затем может просмотреть злоумышленник. Для работы сниффера сетевая карта должна быть переведена в режим прослушивания (promiscuous mode), что позволяет ей «слышать» весь проходящий через сетевой кабель трафик. Стандартная утилита `ipconfig` с помощью специального параметра позволяет пользователю root проверить, работает ли сетевая карта в режиме прослушивания. Поэтому руткит содержит троянскую версию утилиты `ipconfig`, которая скрывает факт работы сетевой карты в режиме прослушивания.

ПРИМЕЧАНИЕ. `Ipconfig` – это утилита, используемая в среде Windows для просмотра сетевых настроек. В Unix/Linux эта утилита называется `ifconfig`.

Также, руткиты обычно содержат «очистители логов» (log scrubber), которые удаляют следы деятельности злоумышленника из системных журналов регистрации событий. Кроме того, руткиты могут содержать троянские версии Unix-утилит `find` и `ls`, позволяющие скрыть наличие относящихся к руткиту файлов, когда пользователь делает листинг содержимого определенной директории.

Некоторые наиболее мощные руткиты вносят изменения в ядро системы, а не просто заменяют отдельные утилиты. Ядро – это мозг операционной системы, поэтому изменение его кода дает злоумышленнику гораздо больший контроль над системой. Кроме того, изменения ядра очень трудно обнаружить, это гораздо сложнее выявления замены утилит. Дело в том, что большинство систем IDS уровня узла контролируют изменения в размерах и датах создания файлов, относящихся к утилитам и программам, но не к ядру операционной системы (за некоторыми исключениями).

ПРИМЕЧАНИЕ. Иногда возникает достаточно интересная ситуация: злоумышленник,

скомпрометировав систему и установив на нее руткит, укрепляет систему, защищая ее от других нападающих. Т.е. когда злоумышленник получает доступ к системе, он делает все то, что должен был сделать администратор – например, отключает ненужные службы и учетные записи пользователей, устанавливает критичные обновления системы и т.д. Злоумышленник делает это для того, чтобы никто другой не мог использовать эту систему или установленный на ней руткит.

Контрмерами против руткитов являются надлежащее укрепление системы, своевременная установка обновлений безопасности, установка антивирусных и антишпионских программ, поддержка их актуальности. Другим механизмом защиты является использование IDS уровня узла (см. Домен 02), которые могут выявить подозрительную активность и контролируют целостность системы. Однако, как было отмечено ранее, функциональные возможности HIDS ограничены и они, как правило, не позволяют выявлять изменения в ядре. Таким образом, лучшей защитой является использование монолитного ядра, а не отдельных модулей ядра. Если вы знакомы с системами Unix/Linux, вы знаете, что они позволяют установить операционную систему в виде отдельных модулей ядра или использовать одно большое ядро. Руткиты, изменяющие ядро, загружаются в виде модулей ядра. Гораздо сложнее (почти невозможно) изменять ядро или влиять на него, если оно является одним целым. Любая основанная на Unix/Linux система, обеспечивающая какую-либо функциональность защиты (прокси-сервер, межсетевой экран, IDS), должна быть установлена с монолитным ядром.

13.1. Шпионское и рекламное программное обеспечение

Определение терминов « шпионская программа» (spyware) и « рекламная программа» (adware) отличается в зависимости от того, у кого вы о нем спрашиваете. Обычно считается, что это программное обеспечение, установленное на компьютере пользователя без его ведома. Такое программное обеспечение, как правило, выполняет сбор определенных данных, например, таких, которые могут быть использованы продавцами для повышения продаж продуктов пользователю, либо данных, которыми могут воспользоваться злоумышленники. Термин «рекламное программное обеспечение» относится, как правило, к программам, которые различные компании используют для отслеживания покупок пользователей, их привычек использования интернет-ресурсов. Обычно они основаны на анализе куки-файлов. Они позволяют продавцу узнать, как эффективно продавать этому пользователю свою продукцию. Некоторые разновидности рекламных программ, устанавливаемые на компьютер пользователя, постоянно показывают ему всплывающие рекламные сообщения, когда он подключен к сети Интернет. Рекламное программное обеспечение может быть частью другого программного продукта, установленного пользователем, либо оно может устанавливаться скрытно.

Шпионское программное обеспечение обычно считается более опасным, чем рекламное, поскольку оно может быть предназначено для перехвата вводимых с клавиатуры символов, перехвата системной информации или установки бэкдора для удаленного несанкционированного входа в систему. С помощью клавиатурных шпионов злоумышленники могут перехватывать пароли, данные кредитных карт или другие критичные данные. Злоумышленники собирают номера счетов, паспортные данные, номера банковских карт, реквизиты доступа к платежным системам, а затем используют их для выполнения незаконной деятельности или для мошенничества. К сожалению, не все антивирусные программы могут выявлять шпионское и рекламное программное обеспечение. Антивирусные программы занимаются поиском известных им сигнатур вирусов, а также признаки деятельности по воспроизводству вирусов, но ни рекламные, ни шпионские программы в настоящее время не занимаются самовоспроизводством и распространением, как вирусы, поэтому они могут продолжать оставаться в системе и делать свою коварную работу даже после полной антивирусной проверки вашей системы и получения отчета об отсутствии вирусов.

ПРИМЕЧАНИЕ. Были разработаны отдельные продукты, позволяющие выявлять рекламное и

шпионское программное обеспечение, а производители антивирусных средств постепенно начинают включать эту функциональность в свои продукты.

ПРЕДУПРЕЖДЕНИЕ. Для сдачи экзамена CISSP очень важно более детально понимать угрозы, связанные с электронной почтой. Рекомендуется прочесть статью « Угрозы электронной почты».

13.2. Передача мгновенных сообщений

Средства для обмена мгновенными сообщениями (IM – Instant messaging) позволяют людям общаться друг с другом в реальном времени и в персональном чате. Эти средства уведомляют пользователей о статусе подключения других пользователей, находящихся в их списке контактов, давая возможность общаться с ними в режиме реального времени. Кроме того, эта технология позволяет передавать файлы между системами этих пользователей. Технологии обмена мгновенными сообщениями основаны на клиент/серверной модели. Пользователь устанавливает на свой компьютер IM-клиент (AOL, ICQ, Yahoo Messenger и т.д.), регистрируется в системе и получает уникальный идентификатор. Этот идентификатор он может сообщить тем людям, с которыми он хочет общаться через систему IM.

Обмен мгновенными сообщениями является эффективной и популярной технологией, однако эта технология имеет множество проблем безопасности, которые необходимо понимать. Передаваемый средствами IM трафик обычно не зашифрован, поэтому передаваемые конфиденциальные данные могут быть перехвачены. Поскольку IM обеспечивает возможность передачи файлов между системами, они могут быть использованы для распространения вирусов, червей и троянских программ. Из-за отсутствия строгой аутентификации, учетные записи могут быть подделаны злоумышленником, и пользователь может в действительности взаимодействовать со злоумышленником, а не с реальным пользователем. Также имеются многочисленные возможности для выполнения атак переполнения буфера, атак с использованием неправильных пакетов. Эти атаки успешно проводились против многих популярных программ IM. Как правило атаки на IM проводятся с целью получения несанкционированного доступа к системе жертвы.

Многие межсетевые экраны не располагают возможностями для анализа трафика такого типа с целью выявления подозрительной активности. Блокирование доступа к сервисам IM с помощью блокирования отдельных портов на межсетевом экране, как правило, оказывается не эффективным, поскольку IM-трафик может использовать другие популярные порты, которые должны быть открыты, например, порт 80 (HTTP) или 21 (FTP). Многие IM-клиенты автоматически настраиваются на работу через другой порт, если порт по умолчанию недоступен и блокируется межсетевым экраном.

Тем не менее, даже зная обо всех этих проблемах и потенциальных уязвимостях, многие компании позволяют своим сотрудникам использовать эту технологию, поскольку она обеспечивает быструю и эффективную связь. Если вам абсолютно необходимо разрешить эту технологию в вашей сети, нужно сделать некоторые вещи, чтобы снизить уровень угроз. Ниже приведены лучшие практики по защите окружения от проблем безопасности, связанных со средствами обмена мгновенными сообщениями:

- Разработайте соответствующую политику безопасности с указанием в ней ограничений при использовании средств IM.
- Установите на всех компьютерах комплексные средства, реализующие функциональность персонального меж сетевого экрана и антивируса.
- Настройте межсетевой экран на блокировку неразрешенного IM-трафика.
- Используйте наиболее безопасные версии средств IM, устанавливайте обновления безопасности для них.
- Установите корпоративные серверы IM для общения сотрудников по внутренней сети компании.

- Либо запретите сотрудникам использовать технологии IM, разрешив им общаться только старомодными способами посредством электронной почты и телефона.

Спам через сети обмена мгновенными сообщениями (SPIM – Instant Messaging Spam) – это разновидность спама, для рассылки которого используются средства обмена мгновенными сообщениями. Хотя такой вид спама пока не так сильно распространен, как спам через электронную почту, его объем постоянно растет. При этом межсетевые экраны не могут блокировать SPIM, что делает его более интересным для спамеров. Единственным эффективным способом для противодействия SPIM является настройка IM-клиента на прием сообщений только от известных отправителей.

Ссылки по теме:

- “Rootkit: Attacker Undercover Tools,” by Saliman Manap, National ICT Security and Emergency Response Centre (NISER)
- “Intro to Spyware,” SpywareGuide
- “Symptoms of Spyware and Other Pests,” Intranet Journal
- “Instant Insecurity: Security Issues of Instant Messaging,” by Neal Hindocha, SecurityFocus (Jan. 13, 2003)
- “Securing Instant Messaging,” Symantec Advantage, Issue 14 (Spring 2002)

14. Резюме

В этом Домене мы коснулись многих технологий, относящихся к различным типам сетей, в том числе мы рассмотрели, как они работают совместно, обеспечивая среду для взаимодействия пользователей, совместного использования ресурсов и обеспечения продуктивности. Каждый аспект создания сети имеет важное значение для безопасности, т.к. уязвимости и слабые места в инфраструктуре могут появиться на любом этапе создания сети. Важно, чтобы вы понимали, каким образом различные устройства, протоколы, механизмы аутентификации, сервисы работают по отдельности и какой они имеют интерфейс для взаимодействия с другими компонентами и технологиями. Это может показаться очень сложной задачей, поскольку используются большое количество различных технологий. Однако знания и упорный труд позволят вам всегда быть на шаг вперед хакеров.

Тест

Вопросы экзамена CISSP являются концептуальными, поэтому они сформулированы соответствующим образом. Задачей кандидата является выбор наилучшего из всех представленных вариантов ответа. Среди вариантов ответа может не быть идеального ответа на поставленный вопрос - кандидат должен выбрать лучший ответ из имеющихся вариантов.

1. Что происходит при выполнении атаки Bluejacking?

- ☐ A. На телефон отправляется незатребованное его владельцем сообщение
- ☐ B. Сотовый телефон клонируется
- ☐ C. Отправляется «червь» по каналу передачи мгновенных сообщений (IM)
- ☐ D. Перехватывается трафик

2. Каким образом TKIP обеспечивает более высокий уровень защиты для среды WLAN?

- ☐ A. Он использует алгоритм AES
- ☐ B. Он уменьшает размер вектора инициализации (IV) и использует алгоритм AES
- ☐ C. Он использует дополнительный ключевой материал
- ☐ D. Он использует фильтрацию по MAC и IP

3. Что из перечисленного ниже не является характеристикой стандарта IEEE 802.11a?

- ☐ A. Он работает в диапазоне 5 ГГц
- ☐ B. Он использует технологию расширения спектра OFDM
- ☐ C. Он обеспечивает пропускную способность 52 Мбит/с
- ☐ D. Он покрывает меньшую площадь по сравнению с 802.11b

4. Что может быть использовано для обхода функции обратного вызова?

- ☐ A. Пассивное прослушивание телефонных разговоров (wiretapping)

- ☐ В. Переадресация вызовов
- ☐ С. Спуфинг пакетов
- ☐ D. Брутфорс-атака

5. Что не считается компонентом архитектуры межсетевого экрана, используемым для защиты сетей?

- ☐ А. Экранированный узел
- ☐ В. Экранированная подсеть
- ☐ С. Шлюз NAT
- ☐ D. Двухуровневая DMZ

6. Почему коммутируемая среда более безопасна, чем маршрутизируемая?

- ☐ А. В ней сложнее перехватывать трафик с помощью sniffера, поскольку компьютеры работают через выделенные виртуальные соединения
- ☐ В. Она также небезопасна, как и некоммутируемые среды
- ☐ С. Шифрование на канальном уровне не позволяет перехватывать передаваемую информацию
- ☐ D. Коммутаторы являются более интеллектуальными устройствами по сравнению с мостами, и они содержат механизмы безопасности

7. Какая функция «кладет трубку» после аутентификации пользователя и ищет в таблице заранее указанный правильный номер телефона?

- ☐ А. Определитель номера (Caller ID)
- ☐ В. RAS
- ☐ С. Обратный вызов (Callback)
- ☐ D. NOS

8. Какой из указанных ниже протоколов выполняет предварительное установление соединения?

- ☐ А. IP
- ☐ В. ICMP
- ☐ С. UDP
- ☐ D. TCP

9. Что из перечисленного ниже лучше всего описывает передачу трафика Ethernet через локальную сеть (LAN)?

- ☐ А. Трафик направляется шлюзу, который посылает его системе получателя
- ☐ В. Трафик хаотичен по своей природе, осуществляется его широковещательная передача всем узлам в подсети
- ☐ С. Трафик передается в виде потока, широковещательная передача данных не производится
- ☐ D. Трафик остается в пределах коллизийного, но не широковещательного домена

10. Какой из перечисленных ниже прокси не может принимать решение о доступе на основе команд протоколов?

- ☐ А. Прикладного уровня
- ☐ В. С фильтрацией пакетов
- ☐ С. Сетевого уровня
- ☐ D. С контролем состояния

11. Какая проблема безопасности часто присутствует в распределенных средах и системах?

- ☐ А. Неизвестны правильные адреса прокси и шлюза по умолчанию
- ☐ В. Неизвестно, кому можно доверять
- ☐ С. Неизвестен наиболее предпочтительный метод аутентификации
- ☐ D. Неизвестно, каким образом преобразовывать имена узлов

12. Какой протокол чаще всего используется для аутентификации пользователей, работающих через соединение dial-up?

- ☐ А. PPTP
- ☐ В. IPSec
- ☐ С. CHAP
- ☐ D. L2F

13. Что из приведенного ниже соответствует последовательности уровней 2, 5, 7, 4 и 3?

- ☐ А. Канальный, сеансовый, прикладной, транспортный и сетевой
- ☐ В. Канальный, транспортный, прикладной, сеансовый и сетевой
- ☐ С. Сетевой, сеансовый, прикладной, сетевой и транспортный
- ☐ D. Сетевой, транспортный, прикладной, сеансовый и представительский

14. Что является другим названием VPN?

- ☐ А. Транспортный сеанс
- ☐ В. Туннель
- ☐ С. Сквозное (end-to-end) соединение
- ☐ D. Полоса пропускания

15. Почему в случае, если безопасность имеет важное значение, следует использовать оптоволоконный кабель?

- ☐ А. Он обеспечивает более высокую скорость передачи данных и менее подвержен помехам
- ☐ В. Он выполняет мультиплексирование данных, что вызывает сложности у атакующих
- ☐ С. Он обеспечивает мощные функции для выявления и исправления ошибок при передаче данных
- ☐ D. Перехват данных очень сложен

16. Почему среды с мейфреймами считаются более безопасными, по сравнению со средами локальных вычислительных сетей (LAN)?

- ☐ А. Обычно в них меньше точек входа
- ☐ В. Они используют более сильные механизмы аутентификации
- ☐ С. Они имеют больше функций для журналирования событий и шифрования данных
- ☐ D. В действительности они являются менее защищенными, чем сети LAN

17. Что подразумевается, когда говорят, что компьютеры взаимодействуют друг с другом физически и логически?

- ☐ А. Они взаимодействуют физически с помощью заголовков и окончаний, а логически – посредством физических соединений
- ☐ В. Они взаимодействуют физически через PVC, а логически – через SVC
- ☐ С. Они взаимодействуют физически при подключении к магистральной сети, а логически – при взаимодействии в рамках одной локальной сети (LAN)
- ☐ D. Они взаимодействуют физически с помощью электронов и сетевых кабелей, а логически – посредством различных уровней модели OSI

18. Как работает инкапсуляция данных и стек протоколов?

- ☐ А. Каждый протокол или сервис на каждом уровне модели OSI добавляет другие пакеты к данным по мере их перемещения вниз по стеку протоколов
- ☐ В. Каждый протокол или сервис на каждом уровне модели OSI добавляет собственную информацию к данным по мере их перемещения вниз по стеку протоколов
- ☐ С. Пакет инкапсулирован и растет по мере прохождения от одного маршрутизатора к другому
- ☐ D. Пакет инкапсулирован и растет по мере прохождения вверх по стеку протоколов

19. Системы, построенные на основе модели OSI, считаются открытыми системами. Что это означает?

- ☐ А. По умолчанию в них не настроен механизм аутентификации
- ☐ В. Они имеют проблемы совместимости
- ☐ С. Они построены с использованием принятых на международном уровне протоколов и стандартов, поэтому они могут легко взаимодействовать с другими открытыми системами
- ☐ D. Они построены с использованием принятых на международном уровне протоколов и стандартов, поэтому при их использовании можно выбирать, с какими типами систем они будут взаимодействовать

20. Какая из приведенных ниже последовательностей протоколов работает на следующих уровнях: прикладной, канальный, сетевой и транспортный?

- ☐ А. FTP, ARP, TCP и UDP
- ☐ В. FTP, ICMP, IP и UDP
- ☐ С. TFTP, ARP, IP и UDP
- ☐ D. TFTP, RARP, IP и ICMP

21. Для чего предназначен представительский уровень?

- ☐ А. Адресация и маршрутизация
- ☐ В. Синтаксис и форматирование данных
- ☐ С. «Сквозное» (end-to-end) соединение
- ☐ D. Кадрирование

22. Для чего предназначен канальный уровень?

- ☐ А. «Сквозное» (end-to-end) соединение
- ☐ В. Управление диалогом (сеансом)
- ☐ С. Кадрирование
- ☐ D. Синтаксис данных

23. Что происходит на сеансовом уровне?

- ☐ А. Управление диалогом (сеансом)
- ☐ В. Маршрутизация
- ☐ С. Упорядочивание пакетов
- ☐ D. Адресация

24. На каком уровне работает мост?

- ☐ А. Сеансовый
- ☐ В. Сетевой
- ☐ С. Транспортный

☐ D. Канальный

25. Что из перечисленного ниже является наилучшим описанием протокола IP?

☐ A. Протокол без предварительного установления соединения, который обеспечивает установление, поддержку и уничтожение диалога (сеанса)

☐ B. Протокол без предварительного установления соединения, который обеспечивает адресацию и маршрутизацию пакетов

☐ C. Протокол с предварительным установлением соединения, который обеспечивает адресацию и маршрутизацию пакетов

☐ D. Протокол с предварительным установлением соединения, который упорядочивание пакетов, выявление ошибок и управление потоком

Домен 06. Криптография.

Криптография – это способ хранения и передачи данных, позволяющий лишь уполномоченным лицам (процессам) читать и обрабатывать их. Криптография – это наука защиты информации путем ее преобразования в нечитаемый вид. Криптография является эффективным способом защиты критичной информации при ее хранении и передаче по недоверенным каналам связи.

Одной из целей криптографии и лежащих в ее основе механизмов, является скрывание информации от неуполномоченных лиц. Однако хакеры, обладающие достаточным объемом времени, ресурсов и мотивации, могут взломать почти любой алгоритм и получить доступ к зашифрованной информации. Более реалистичной целью криптографии является попытка сделать взлом зашифрованной информации слишком сложной и длительной по времени задачей для злоумышленника, обладающего ограниченными ресурсами.

Первые методы шифрования появились около 4000 лет назад, по большей части они использовали графические методы. Позднее криптография была адаптирована для использования в военных, коммерческих, правительственных и других целях – там, где секреты нуждались в защите. Незадолго до появления Интернета, шифрование получило новое применение – оно стало важным инструментом для ежедневных операций. На протяжении всей истории люди и правительства работали над защитой передаваемой информации с помощью ее шифрования. В результате алгоритмы и устройства шифрования становились все более сложными, постоянно внедрялись новые методы и алгоритмы. В настоящее время шифрование стало неотъемлемой частью компьютерного мира.

У криптографии интересная история, она претерпела множество изменений на протяжении веков. Сохранение секретов очень важно для работы цивилизации. Это также относится и к отдельным людям и группам людей, которые используют возможность скрывать свои реальные намерения для получения конкурентного преимущества и снижения уязвимостей и т. п.

Изменения, которым подверглась криптография, напрямую связаны с совершенствованием технологий. Самые первые методы криптографии использовались людьми при письме на дереве или камне, которые передавались другим людям, имевшим все необходимое для расшифрования сообщений. Криптография прошла долгий путь. Сегодня она применяется в потоках двоичного кода, проходящих по сетевым проводам, коммуникационным маршрутам Интернета и радиоволнам.

1. История криптографии

Корни криптографии уходят в Египет 2000 года до нашей эры, когда для украшения гробниц использовались иероглифы, рассказывающие историю жизни умершего. Фактически, иероглифы использовались, в первую очередь, не для скрывания информации, а для того, чтобы показать историю жизни более благородной, церемониальной и волшебной.

Методы шифрования применялись в основном для того, чтобы продемонстрировать практическое применение возможности скрывания информации от других.

Еврейский криптографический метод использовал перевернутый алфавит так, что каждой букве в нормальном алфавите соответствовала буква в перевернутом (или сдвинутом) алфавите. Этот метод шифрования называется атбаш (atbash), он использовался для скрывания реального смысла сообщения. Пример ключа шифрования, используемого в схеме шифрования атбаш, показан ниже:

ABCDEFGHIJKLMNOPQRSTUVWXYZ
ZYXWVUTSRQPONMLKJIHGFEDCBA

Например, слово «security» при шифровании преобразуется в «hvxfirgb».

Это пример **шифра подстановки** (substitution cipher), т.к. в нем каждая буква заменяется на другую букву. Этот тип шифра подстановки называется **моноалфавитным шифром подстановки** (monoalphabetic substitution cipher), поскольку он использует только один алфавит. Существует **полиалфавитный шифр подстановки** (polyalphabetic substitution cipher), использующий несколько алфавитов.

Этот простейший метод шифрования работал в те древние времена, но, в конце концов, потребовались более сложные алгоритмы.

ПРИМЕЧАНИЕ. Термин «шифр» - это то же самое, что «алгоритм шифрования».

Около 400 года до нашей эры Спартанцы использовали систему шифрования информации, с помощью которой они писали сообщения на листах папируса, а затем оборачивали его вокруг деревянной палки. Эта палка доставлялась получателю, который оборачивал папирус вокруг другой палки. Получатель мог прочитать сообщение только в том случае, если он обернул папирус вокруг палки правильного размера, как показано на Рисунке 6-1. Это называется **шифром скитала** (scytale cipher). Пока папирус не был обернут вокруг правильной палки, надписи на нем казались просто кучей случайных символов.

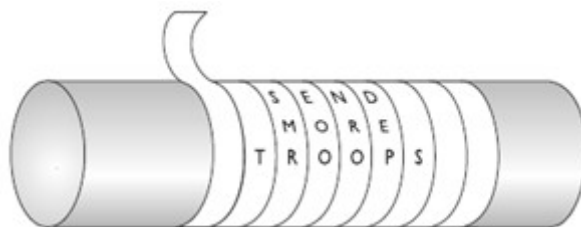


Рисунок 6-1. Скитала использовалась Спартанцами для шифрования и расшифрования сообщений

Позднее в Риме Юлий Цезарь (100-44 годы до нашей эры) разработал простой метод сдвига букв алфавита, похожий на схему атбаш. Он просто сдвигал алфавит на три буквы. В примере ниже показан стандартный алфавит и сдвинутый алфавит. Алфавит служит алгоритмом, а ключом является число позиций, на которое должны быть сдвинуты буквы в нем в процессе шифрования и расшифрования.

Стандартный алфавит:

ABCDEFGHIJKLMNOPQRSTUVWXYZ

Сдвинутый алфавит:

DEFGHIJKLMNOPQRSTUVWXYZABC

Рассмотрим в качестве примера зашифрование сообщения «Logical Security». Мы берем первую букву этого сообщения «L» и сдвигаем ее на три позиции по алфавиту. Зашифрованная версия первой буквы будет «O», которую мы записываем. Следующей буквой нашего сообщения является «O», которая соответствует букве «R» в сдвинутом на три позиции алфавите. Мы продолжаем этот процесс, пока не зашифруем все сообщение. Зашифрованное сообщение мы передаем получателю, который выполняет тот же процесс в обратном порядке.

Исходный текст:

LOGICAL SECURITY

Шифротекст:

ORJLFDO VHFUXULWB

Сегодня такие техники выглядят слишком примитивными, чтобы быть реально

эффективными, однако во времена Юлия Цезаря, когда вообще немногие умели читать, это обеспечивало высокий уровень защиты. Шифр Цезаря также является примером моноалфавитного шифра. Когда больше людей научилось читать и смогло проводить обратный инжиниринг процесса такого шифрования, криптографы усложнили методику, создав полиалфавитные шифры.

ROT13. Еще совсем недавно (в 1980-х годах) применялся метод шифрования ROT13, который на самом деле использовал те же идеи, что и шифр Цезаря. Он использовал сдвиг алфавита на 13 букв вместо трех. Он не использовался для реальной защиты данных, потому что в то время люди уже могли легко справиться с такой задачей. Он использовался в различных онлайн-форумах для публикации запрещенной информации, для ее распространения среди пользователей.

В 16-м веке во Франции Вижнер разработал полиалфавитный шифр подстановки для Генри III. Он был основан на шифре Цезаря, но увеличивал сложность процессов шифрования и расшифрования.

Как показано на Рисунке 6-2, нам нужно зашифровать сообщение SYSTEM SECURITY AND CONTROL. У нас есть ключ, который имеет значение SECURITY. Также у нас есть таблица Вижнера (или алгоритм), который в действительности является просто более продвинутым вариантом шифра Цезаря. Тогда как Цезарь использовал для шифрования только один сдвинутый алфавит, шифр Вижнера использовал 27 сдвинутых алфавитов, в каждом из которых алфавит был сдвинут на одну букву дальше, чем в предыдущем.

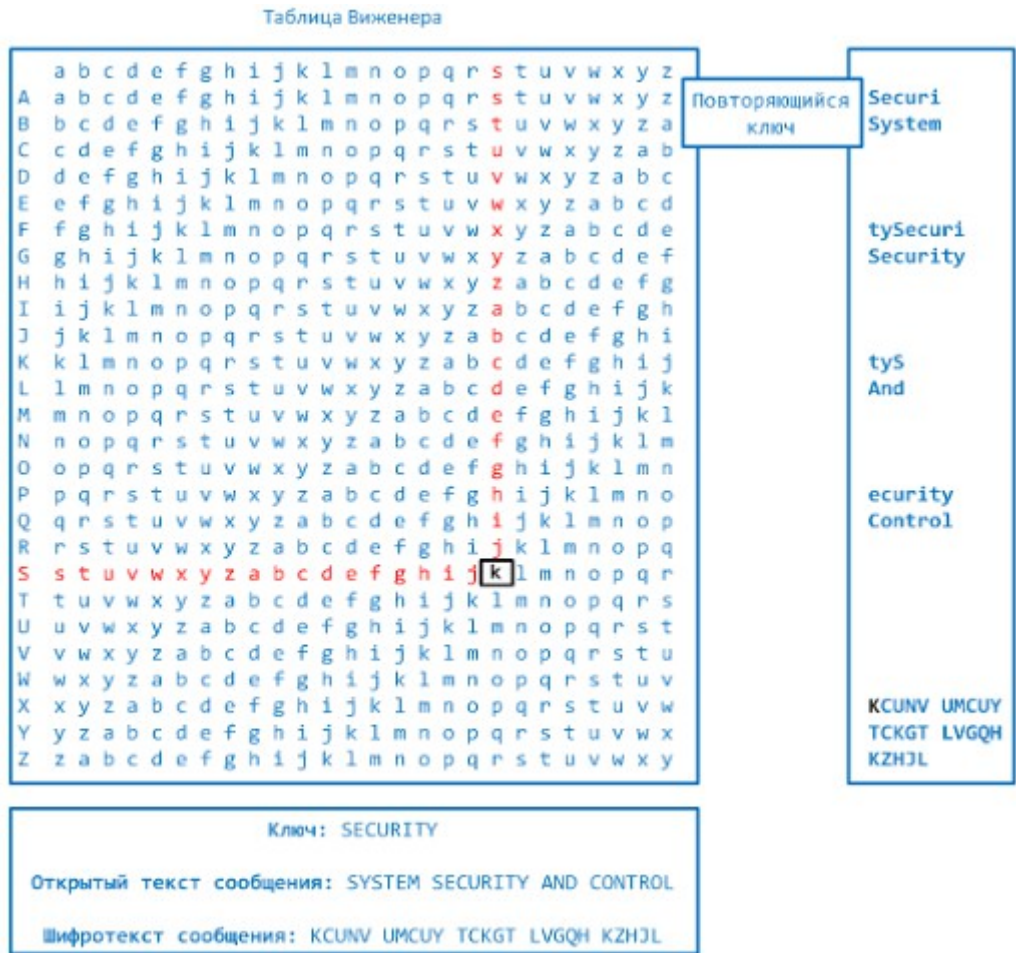


Рисунок 6-2. Полиалфавитный алгоритм был разработан для повышения сложности шифрования

ПРИМЕЧАНИЕ. *Открытый текст* (plaintext) – это читаемая версия сообщения. Полученный в результате выполнения процесса шифрования текст называют *шифротекстом* (ciphertext).

Итак, взглянув на пример на Рисунке 6-2, мы берем первую букву ключа «S» и начинаем с первого алфавита в нашем алгоритме, отыскивая в нем столбец «S». Затем мы берем первую

букву открытого текста «S» и опускаемся по выбранному столбцу вниз, до строки «S». На пересечении этих строки и столбца мы видим букву «K». Это и есть первая буква шифротекста для нашего исходного сообщения – записываем ее. Затем мы берем следующую букву ключа «E» и следующую букву открытого текста «Y». На пересечении столбца «E» и строки «Y» мы видим букву «C». Это вторая буква шифротекста, которую мы записываем рядом с первой. Мы продолжаем этот процесс для всего сообщения (если длина сообщения превышает длину ключа, буквы ключа циклически повторяются). Полученный в результате шифротекст мы отправляем получателю. Получатель должен использовать тот же самый алгоритм (таблицу Виженера) и иметь тот же ключ (SECURITY) для выполнения обратного процесса преобразования и получения исходного текста сообщения.

В Средние века эволюция криптографии продолжилась. Различные страны улучшали свои методы, используя новые инструменты. В конце 19 века криптография стала неотъемлемой частью взаимодействия между военными подразделениями.

На протяжении Второй Мировой Войны, устройства шифрования широко использовались для передачи тактической информации, они были существенно улучшены с помощью механических и электромеханических технологий, которые ранее были даны миру телеграфом и радиосвязью. Роторные шифровальные машины являлись устройствами, которые замещали буквы с помощью различных роторов внутри машины. Это был огромный прорыв в военной криптографии. Роторные шифровальные машины многократно повысили сложность, усложнив возможности взлома. Одной из самых известных роторных шифровальных машин является немецкая шифровальная машина Энигма. Машина Энигма имела отдельные роторы, наборное поле и зеркальный ротор.

Перед началом процесса шифрования, оператор (источник сообщения) должен был произвести начальную настройку Энигмы. Затем оператор печатал первую букву сообщения, а машина заменяла эту букву на другую букву и показывала ее оператору. Шифрование заканчивалось после перемещения роторов заранее определенное число раз. Например, если оператор печатал букву «Т» в качестве первой буквы, машина Энигма могла показать ему букву «М» в качестве первого замененного значения. Оператор записывал эту букву «М» на лист. Затем оператор передвигал ротор вперед и вводил следующую букву. Каждый раз перед вводом новой буквы оператор перемещал ротор. Процесс продолжался пока все сообщение не было зашифровано. Затем зашифрованный текст передавался по радиоволнам обычно на немецкие подводные лодки класса U. Выбор замены для каждой буквы зависел от установки ротора. Таким образом, наиважнейшей секретной частью этого процесса (ключом) была начальная установка и то, как оператор передвигал ротор в процессе шифрования и расшифрования сообщения. Оператор на каждой стороне должен был знать эту последовательность перемещений. Это позволяло немецким военным взаимодействовать между собой.

Хотя механизмы Энигмы были со временем усложнены, команда польских криптографов смогла взломать код, что позволило Британии проникнуть в немецкие планы атак и перемещений войск. Говорят, что взлом механизма шифрования сократил продолжительность Второй Мировой Войны на два года. После войны была опубликована детальная информация об Энигме, одна из машин выставлена в Смитсоновском Институте.

У криптографии богатая история. Мария, Королева Шотландская, поплатилась жизнью из-за того, что отправленное ей зашифрованное сообщение было перехвачено. Во время Войны за Независимость Бенедикт Арнольд использовал кодовую книгу (codebook cipher) для обмена информацией о перемещениях войск и стратегических военных успехах. Военные всегда лидировали в использовании криптографии как для шифрования информации, так и в попытках дешифровать зашифрованную врагами информацию. Уильяма Фридерика Фридмана, который опубликовал в 1920 году *«Показатель совпадения и его применение в криптографии»*, называют отцом современной криптографии. Он взломал множество

сообщений, перехваченных в ходе Второй Мировой Войны. Шифрование использовалось многими правительствами и военными. Оно внесло большой вклад в Великую Победу, т.к. позволило выполнять скрытые секретные маневры. Однако немецкой армии оно принесло великое поражение, когда ее криптосистемы были вскрыты и дешифрованы.

Когда были изобретены компьютеры, возможности методов и устройств шифрования возросли экспоненциально, криптография была многократно усилена. Эра компьютеров дала разработчикам криптографии беспрецедентные возможности для разработки новых методов шифрования. Самым известным и успешным проектом был *Люцифер*, разработанный IBM. Люцифер использовал сложные математические уравнения и функции, которые позднее были адаптированы и модифицированы NSA (National Security Agency – Агентство национальной безопасности США) в выпущенном им в 1976 году стандарте DES (Data Encryption Standard – Стандарт шифрования данных), ставшим правительственным стандартом. DES использовался во всем мире для проведения финансовых и иных транзакций, он был встроен в огромное количество коммерческих приложений. DES имеет богатую историю, он использовался на протяжении 25 лет.

Большинство сетевых протоколов, разработанных на заре компьютерной эры, были усовершенствованы путем включения в них криптографических возможностей и добавления необходимых уровней защиты. Шифрование используется в аппаратных устройствах и программном обеспечении, оно предназначено для защиты данных, банковских транзакций, корпоративных сетевых коммуникаций в экстрасетях, сообщений электронной почты, веб-транзакций, беспроводных коммуникаций, хранения конфиденциальной информации, факсов, телефонных звонков и многого другого.

Усилия взломщиков кодов и криптоаналитиков, а также возможности обработки огромных объемов чисел заполнившими рынок микропроцессорами, с каждым годом ускоряли эволюцию криптографии. Поскольку плохие парни становились все умнее и получали в свое распоряжение больше ресурсов, хорошие парни должны были увеличивать свои усилия и улучшать стратегии. **Криптоанализ** (cryptanalysis) – это наука, изучающая и взламывающая секреты шифровальных процессов, компрометирующая схемы аутентификации и выполняющая обратный инжиниринг алгоритмов и ключей. Криптоанализ является важной частью криптографии и криптологии. Криптоанализ, выполняемый хорошими парнями, предназначен для выявления недостатков и слабостей, заставляя разработчиков вернуться к чертежной доске и усовершенствовать компоненты. Также криптоанализ выполняется любопытными и мотивированными хакерами, для выявления тех же недостатков, но только с целью получения ключа шифрования для несанкционированного доступа с его помощью к конфиденциальной информации.

ПРИМЕЧАНИЕ. Криптоанализ – это очень сложная наука, которая охватывает широкий спектр тестов и атак. Мы рассмотрим эти атаки в конце этого Домена. Криптология, с другой стороны, изучает криптоанализ и криптографию.

Нашей цивилизацией использовались различные типы криптографии. Сегодня криптография находится глубоко в корнях телекоммуникационных и компьютерных технологий. Автоматизированные информационные системы и криптография играют гигантскую роль в эффективности военных, функциональности правительств, экономике частного бизнеса. Поскольку наша зависимость от технологий возрастает, растет и наша зависимость от криптографии, т.к. всегда возникает потребность в хранении секретов.

Ссылки по теме:

- Chapter 2.1, “Security Strategies for E-Companies,” by Fred Cohen
- Trinity College Department of Computer Science Historical Cryptography web site

2. Определения и концепции криптографии

Шифрование – это метод преобразования читаемых данных, называемых **открытым текстом** (plaintext), в форму, которая выглядит случайной и нечитаемой, она называется **шифротекстом** (chiphertext). Открытый текст – это форма текста, понятная каждому человеку (документ) или компьютеру (исполняемый код). После того, как открытый текст будет преобразован в шифротекст, ни человек, ни машина не смогут правильно обработать его до выполнения расшифрования. Это позволяет передавать конфиденциальную информацию через незащищенные каналы, не опасаясь несанкционированного доступа к ней. При хранении данных на компьютере, они обычно защищаются логическими и физическими механизмами контроля доступа. Однако при передаче данных по сети, они больше не защищаются этими механизмами и находятся в более уязвимом состоянии.



Системы или продукты, которые предоставляют функции зашифрования и расшифрования, называются **криптосистемами** (cryptosystem), они могут создаваться в виде аппаратных компонентов или программного кода в приложениях. Криптосистемы используют алгоритмы шифрования (которые определяются как простые или сложные с точки зрения процесса шифрования), ключи, а также необходимые программные компоненты и протоколы. Большинство алгоритмов являются сложными математическими формулами, которые в определенной последовательности применяются к открытому тексту. Большинство методов шифрования используют секретное значение, называемое **ключом** (обычно ключ представляет собой длинную последовательность битов), который используется в процессе работы алгоритмом для зашифрования и расшифрования текста.

Алгоритм – это набор правил, также называемый шифром. Он определяет, как должно происходить зашифрование и расшифрование. Многие математические алгоритмы, используемые сегодня в компьютерных системах, являются публично доступными и широкоизвестными – процесс шифрования не является секретом. Раз внутренние механизмы алгоритма не секретны, значит секретным должно быть что-то другое. Секретной частью общеизвестного алгоритма шифрования является ключ. Здесь можно провести аналогию с замком, чтобы проиллюстрировать это. Замок можно купить в любом магазине, множество людей используют замки одинакового производителя. Однако это вовсе не означает, что они могут открыть дверь друг друга и получить несанкционированный доступ в квартиру. Каждый замок имеет свой собственный ключ, который может открыть только соответствующий ему конкретный экземпляр замка.

В шифровании, **ключ** (криптопеременная) – это значение, которое состоит из длинной последовательности случайных битов. Но действительно ли это просто случайный набор битов, собранных вместе? На самом деле – нет. Алгоритм использует **ключевое пространство** (keyspace), являющееся диапазоном значений, которые могут использоваться для создания ключа. Когда алгоритму нужно сгенерировать новый ключ, он использует случайные значения из этого ключевого пространства. Чем шире ключевое пространство, тем больше доступных значений можно использовать для создания ключа, а чем больше случайных вариантов ключей, тем сложнее взломщику подобрать их. Например, если алгоритм позволяет использовать ключи длиной 2 бита, ключевое пространство для этого алгоритма составляет всего 4 возможных значения, это максимальное количество возможных вариантов различных ключей для этого алгоритма. Это очень узкое ключевое пространство, поэтому атакующему не потребуется много времени, чтобы найти правильный ключ и воспользоваться им.

Широкое ключевое пространство включает в себя гораздо больше возможных вариантов ключей. Сегодня используются ключи, которые чаще всего имеют длину 128, 256, 512 или 1024 бита. Если размер ключа равен 512 бит, он обеспечивает 2^{512} возможных вариантов

ключей (ключевое пространство). Алгоритмы шифрования должны использовать все ключевое пространство и выбирать значение для нового ключа максимально случайным образом. Если используется небольшое ключевое пространство, оно может обеспечить слишком мало вариантов для генерации ключа, как это показано на Рисунке 6-3. Это повышает шансы атакующего взломать ключ и дешифровать защищенную информацию.

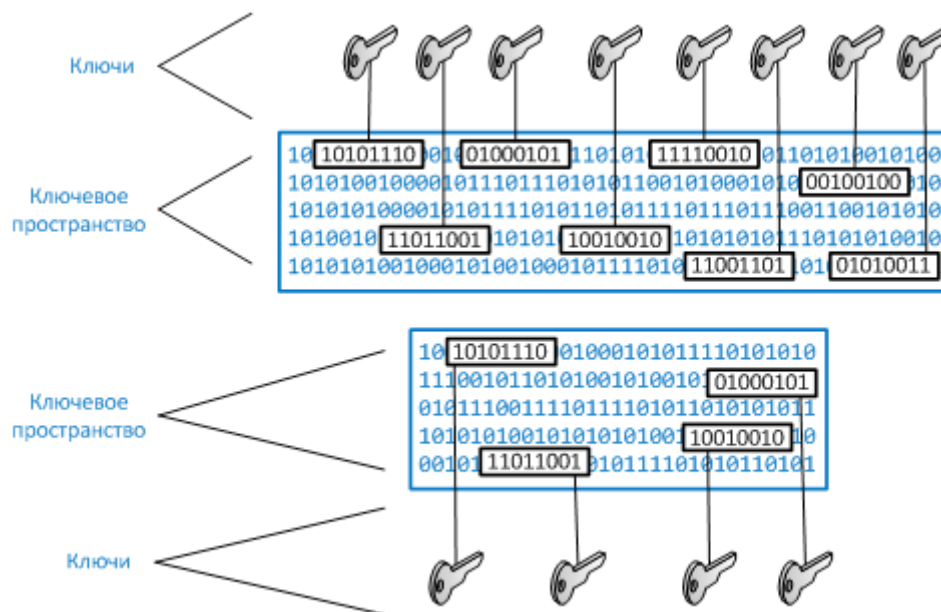


Рисунок 6-3. Более широкое ключевое пространство позволяет получить гораздо большее количество возможных вариантов значений ключей

Если злоумышленник перехватит сообщение, передаваемое между двумя людьми, он сможет просмотреть это сообщение, однако, если оно передается в зашифрованном виде, оно нечитаемо. Даже если злоумышленник знает алгоритм, используемый этими людьми для зашифрования и расшифрования информации, без знания ключа эта информация будет бесполезной для злоумышленника, как показано на Рисунке 6-4.

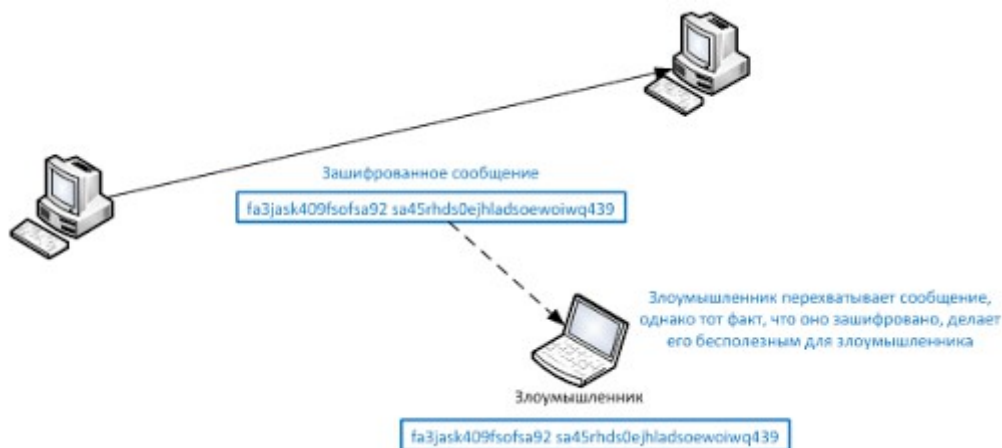


Рисунок 6-4. Без знания правильного ключа, перехваченное сообщение будет бесполезно для злоумышленника

Криптосистемы. Криптосистема содержит все необходимые компоненты для выполнения зашифрования и расшифрования. Примером криптосистемы является PGP (Pretty Good Privacy). Криптосистема содержит как минимум следующее:

- Программное обеспечение
- Протоколы
- Алгоритмы

- Ключи

2.1. Принцип Керкхофса

Огюст Керкхофс опубликовал в 1883 году статью, в которой он заявил, что в криптосистеме единственным секретом должен быть ключ, алгоритм должен быть общеизвестен. Он утверждал, что если секретность основана на слишком большом количестве секретов, она становилась более уязвимой, и этим можно было воспользоваться.

Но какое нам дело до того, что сказал кто-то почти 130 лет назад? Дело в том, что дебаты на эту тему продолжаются до сих пор. Криптографы, работающие в коммерческих и академических областях, согласились с принципом Керкхофса, потому что сделать алгоритм публично доступным означало, что гораздо больше людей увидит исходный код, протестирует его, найдет все его недостатки и слабости. Они сочли, что много голов лучше, чем одна. Когда кто-то вскрыет какие-либо недостатки, разработчик сможет исправить их и предоставить обществу более сильный алгоритм.

Однако не все приняли эту философию. Правительства по всему миру создают собственные алгоритмы, не делая их публичными. Они считают, что чем меньше людей знает, как работает алгоритм, тем меньше людей узнает, как взломать его. Криптографы в коммерческой области не согласились с таким подходом и не доверяют алгоритмам, которые не проверили сами.

Это очень похоже на происходящие в настоящее время дебаты о программном обеспечении с открытым кодом по сравнению с предварительно скомпилированным.

2.2. Стойкость криптосистем

Стойкость методов шифрования основывается на алгоритме, обеспечении секретности ключа, длине ключа, векторах инициализации, а также том, как все это работает вместе в рамках криптосистемы. При обсуждении стойкости шифрования, говорят о сложности вскрытия алгоритма или ключа, даже если алгоритм не является публичным. При попытке взлома криптосистемы, обычно выполняют перебор всех возможных значений (обычно количество возможных значений колоссально), чтобы найти именно то значение (ключ), которое позволяет расшифровать конкретное сообщение. Стойкость метода шифрования находится в прямой связи с величиной мощностей и количеством времени, необходимых для взлома криптосистемы (перебора всех возможных значений) и получения правильного ключа. Взлом криптосистемы может выполняться с помощью брутфорс-атаки, при которой проверяется каждое возможное значение ключа до тех пор, пока полученный в результате расшифрования исходного сообщения с помощью очередного ключа открытый текст не станет понятным. В зависимости от алгоритма и длины ключа, это может быть как легкой задачей, так и практически невозможной. Если ключ можно взломать за три часа на компьютере с процессором Pentium II, шифр считается абсолютно нестойким. Если ключ может быть взломан только при использовании многопроцессорной системы с тысячей процессоров за 1,2 миллионов лет, шифр считается очень стойким.

ПРИМЕЧАНИЕ. Векторы инициализации мы рассмотрим далее в этом Домене.

В процессе создания нового метода шифрования, основной задачей является создание условий, при которых взлом станет слишком дорогим или требующим слишком много времени. Синонимом стойкости криптосистемы является **фактор трудозатрат**, который указывает на оценку необходимых ресурсов для взлома криптосистемы атакующим.

Требуемая стойкость механизма защиты зависит от критичности защищаемых данных. Нет смысла шифровать информацию о субботнем барбекю с другим алгоритмом, предназначенным для защиты совершенно секретной информации. В то же время защита собранной разведывательной информации с помощью PGP вряд ли будет хорошей идеей. Каждый тип механизма шифрования имеет свое место и свое назначение.

Даже если алгоритм очень сложный и совершенный, другие проблемы шифрования могут ослабить его. Например, ослабить даже самый стойкий шифр может ненадлежащая защита секретного значения ключа.

Для обеспечения стойкости шифрования важно использовать алгоритмы, не имеющие недостатков, применять длинные ключи, при генерации ключей использовать весь возможный диапазон ключевого пространства, а также хорошо защищать ключ. Если хотя бы один из этих пунктов не выполняется, это может оказать негативное влияние на весь процесс.

2.3. Сервисы криптосистем

Криптосистемы могут предоставлять следующие сервисы:

- **Конфиденциальность.** Информация приводится в форму, нечитаемую для всех, кроме уполномоченных людей (или систем).
- **Целостность.** Данные не могут быть несанкционированно изменены в процессе их создания, передачи или хранения.
- **Аутентификация.** Проверка личности пользователя или системы, создавшей информацию.
- **Авторизация.** После идентификации человек вводит ключ или пароль, который позволяет получить доступ к определенному ресурсу.
- **Неотказуемость (nonrepudiation).** Обеспечивает невозможность отрицания отправителем факта отправки.

В качестве примера, иллюстрирующего работу этих сервисов, представим, что ваш начальник отправляет вам сообщение об увеличении вашей зарплаты вдвое. Сообщение зашифровано, поэтому вы можете быть уверены, что оно исходит от вашего начальника (аутентичность), что никто не изменил сообщение до момента его получения вашим компьютером (целостность), что никто не смог прочитать его в процессе передачи по сети (конфиденциальность), и что ваш начальник не сможет позднее сказать, что не отправлял вам это сообщение (неотказуемость).

Различные типы сообщений и транзакций в различной степени нуждаются в некоторых (или во всех) этих сервисах, обеспечиваемых криптографией. Военные и разведывательные агентства больше всего беспокоятся о сохранении своей информации в тайне, поэтому они выбирают механизмы шифрования, обеспечивающие высокий уровень секретности. Финансовые организации также заботятся о конфиденциальности, но еще важнее для них вопросы целостности данных при их передаче, поэтому выбираемые ими механизмы шифрования немного отличаются от механизмов шифрования, выбираемых военными. При передаче финансового сообщения, даже если в нем будет изменен всего один символ, это может привести к очень серьезным последствиям. Юридические фирмы в большей степени заботятся об аутентичности получаемых сообщений. Если полученная информация должна быть представлена в суде, ее аутентичность не должна вызывать никаких сомнений. Поэтому используемые этими фирмами методы шифрования должны обеспечивать аутентичность, подтверждающую личность отправителя информации.

ПРИМЕЧАНИЕ. Если Дэвид отправил некое сообщение, а затем заявляет, что он не отправлял его, это называется отказом от авторства (act of repudiation). Если криптографический алгоритм обеспечивает неотказуемость, отправитель не сможет отрицать факт отправки им сообщения (конечно он может попытаться, но криптосистема скажет обратное). Это способ подтверждения честности отправителя.

Количество типов и способов применения криптографии растет с каждым годом. Раньше криптография в основном применялась для сохранения секретности (обеспечения конфиденциальности), но сегодня мы используем криптографию также для обеспечения

целостности данных, аутентификации сообщений, для подтверждения факта получения сообщения, для управления доступом и т.д. В этом Домене мы рассмотрим различные типы криптографии, которая предоставляет различную функциональность, а также связанные с этим проблемы.

Определения в криптографии. В криптографии важно правильно понимать следующие определения:

- **Управление доступом.** Ограничение и контроль попыток доступа субъектов к объектам.
- **Алгоритм.** Набор математических правил, используемых для зашифрования и расшифрования.
- **Шифр.** Другое название алгоритма.
- **Криптография.** Наука секретного письма, которая позволяет сохранять и передавать данные в форме, доступной только уполномоченным лицам.
- **Криптосистема.** Аппаратная или программная реализация криптографии, которая преобразует исходное сообщение в шифротекст, либо шифротекст обратно в открытый текст.
- **Криптоанализ.** Практическая работа по взлому криптосистем.
- **Криптология.** Изучение криптографии и криптоанализа.
- **Аутентификация источника данных** (data origin authentication). Проверка источника сообщения (аутентификация системы).
- **Зашифрование** (encipher). Действие по преобразованию исходных данных в нечитаемый формат.
- **Аутентификация отправителя** (entity authentication). Проверка личности отправителя сообщения.
- **Расшифрование** (decipher). Действие по преобразованию шифротекста обратно в читаемую форму.
- **Ключ.** Секретная последовательность битов и инструкций, которые управляют выполнением действий при зашифровании и расшифровании.
- **Кластеризация ключей** (key clustering). Случай, когда с помощью двух различных ключей генерируется одинаковый шифротекст из одинакового открытого текста.
- **Ключевое пространство** (keyspace). Диапазон возможных значений, используемый при создании ключа.
- **Открытый текст** (plaintext). Данные в читаемом формате, также называемые простым текстом (cleartext).
- **Квитанция** (receipt). Подтверждение получения сообщения.
- **Фактор трудозатрат** (work factor). Предполагаемое время, усилия и ресурсы, необходимые для взлома криптосистемы.

Если некоторые из приведенных выше терминов вам пока не совсем понятны, просто запомните их определение. Мы более подробно рассмотрим их далее в этом Домене.

2.4. Одноразовый шифровальный блокнот

Одноразовый шифровальный блокнот (one-time pad) – это прекрасная схема шифрования, т.к. при правильной реализации она невзламываема. Она была создана Гилбертом Вернамом в 1917 году и иногда называется шифром Вернама.

Этот алгоритм не использует сдвиг алфавитов, как шифры Цезаря и Виженера, описанные ранее, вместо этого он использует блокнот, заполненный случайными значениями, как показано на Рисунке 6-5. Нам нужно зашифровать некое сообщение и преобразовать его в биты, для этого мы используем наш одноразовый шифровальный блокнот, который заполнен случайными битами. В процессе шифрования используется двоичная математическая

функция «исключающее ИЛИ» (XOR).

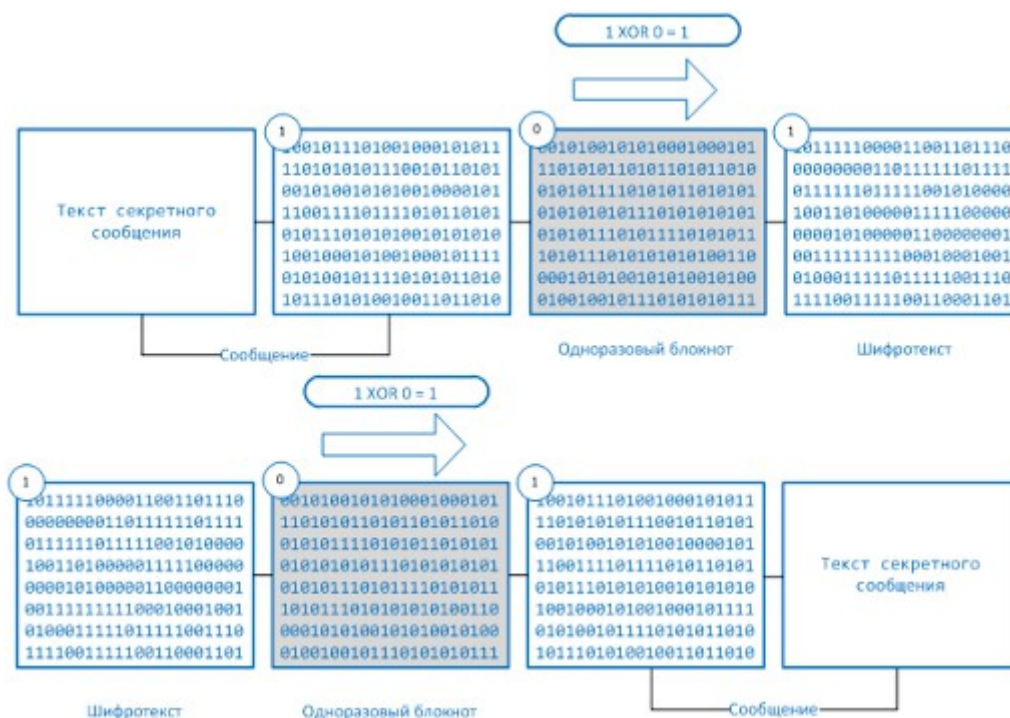


Рисунок 6-5. Одноразовый шифровальный блокнот

XOR – это операция над двумя битами, она часто используется в двоичной математике и методах шифрования. При выполнении XOR над двумя битами, имеющими одинаковое значение, в результате получается 0 ($1 \text{ XOR } 1 = 0$), если значение битов разное, в результате получается 1 ($1 \text{ XOR } 0 = 1$). Например:

Поток сообщения: 1001010111
Ключевой поток: 0011101010
Поток шифротекста: 1010111101

Так, в нашем примере выполняется операция XOR над первым битом сообщения (1) и первым битом в одноразовом блокноте (0), что дает первое значение шифротекста (1). Затем выполняется XOR над следующим битом сообщения (0) и следующим битом в блокноте (0), что дает второе значение шифротекста (0). Этот процесс продолжается пока все сообщение не будет зашифровано. Полученное в результате зашифрованное сообщение отправляется получателю.

На Рисунке 6-5 мы видим, что получатель должен иметь точно такой же шифровальный блокнот для расшифрования сообщения путем выполнения обратного процесса. Получатель выполняет XOR над первым битом зашифрованного сообщения и первым битом в блокноте. В результате он получает первый бит открытого текста. Получатель продолжает этот процесс, пока не расшифрует все сообщение.

Схема шифрования с использованием одноразового шифровального блокнота считается невзламываемой только в том случае, если в процессе ее реализации выполнены следующие условия:

- Блокнот должен использоваться только один раз. Если он используется более одного раза, это может привести к появлению шаблонов (pattern) в процессе шифрования, что поможет злоумышленнику взломать шифр.
- Блокнот должен существовать ровно столько же времени, что и само сообщение.

Если он уничтожен раньше, не удастся расшифровать сообщение. А если он используется и в дальнейшем, его многократное применение создает описанную выше проблему с появлением шаблонов.

- *Блокнот должен распространяться безопасным образом и защищаться получателем.* Это очень сложный и неудобный процесс, поскольку блокноты обычно представляют из себя просто отдельные листы бумаги, которые нужно доставлять с доверенным курьером и надежно охранять в каждом пункте назначения.
- *Блокнот должен быть заполнен действительно случайными значениями.* Это кажется простой задачей, однако даже современные компьютерные системы не обладают генераторами действительно случайных чисел, на них используются генераторы псевдослучайных чисел.

ПРИМЕЧАНИЕ. Генератор чисел (number generator) используется для создания потока случайных значений. Предварительно он должен быть инициализирован начальным значением. Соответствующая часть программного обеспечения берет в качестве начального значения сочетание нескольких значений переменных состояния компьютерной системы (время, циклы процессора и т.п.). Несмотря на то, что компьютерная система сложна, она все же является предсказуемой средой, поэтому получаемые значения в любом случае предсказуемы, они не являются действительно случайными, а только *псевдослучайными*.

Хотя подход одноразовых шифровальных блокнотов может обеспечить очень высокий уровень безопасности, во многих ситуациях он оказывается не практичным из-за его многочисленных требований. Каждая пара субъектов, которым потенциально может понадобиться взаимодействовать таким способом, должна безопасным способом получить блокнот на время, равное времени актуальности сообщения. Такой способ управления ключами может быть чрезвычайно сложным, он ведет к большим накладным расходам, часто превышающим преимущества от его использования. Распространение блокнотов может быть очень сложным процессом, получатель и отправитель должны быть хорошо синхронизированы, чтобы каждый использовал одинаковый блокнот.

На протяжении своей истории, одноразовые блокноты применялись для защиты различных типов критичных данных. Сегодня они продолжают применяться военными в качестве резервного варианта шифрования на случай, если основной процесс шифрования (который требует наличия компьютеров и источников энергии) недоступен по причинам начавшейся войны или в результате атаки.

Требования при использовании одноразовых блокнотов. Чтобы схема шифрования с одноразовыми блокнотами была невзламываемой, каждый блокнот в схеме должен:

- Состоять из действительно случайных значений
- Использоваться только один раз
- Безопасно передаваться получателю
- Должен быть надежно защищен как на стороне отправителя, так и на стороне получателя
- Срок его жизни не должен отличаться от срока жизни сообщения

2.5. Динамические и скрытые шифры

Двумя шифрами из шпионских романов являются шифры с динамическим ключом (running key) и скрытые шифры (concealment cipher). **Шифр с динамическим ключом** может использовать ключ, которому не нужен электронный алгоритм и замена битов, вместо этого он использует компоненты окружающего его мира. Например, алгоритмом может быть набор книг, принятый отправителем и получателем. Ключом в таком шифре может быть страница в книге, номер строки и буквы. Например, если я получаю сообщение от суперсекретного шпиона, в котором написано «14916с7.29913с7.91115с8», оно может означать, что мне нужно взять первую книгу из заранее определенного набора книг, открыть 49 страницу, найти шестую сверху строку и в ней седьмую от края страницы букву. Я

записываю эту букву. Следующий числовой набор в сообщении говорит мне, что нужно взять вторую книгу, открыть 99 страницу, найти третью строку и в ней седьмую букву. Последнюю букву я беру из девятой книги, с 11 страницы, из пятой строки, восьмую в строке. Таким образом, я получаю важное секретное сообщение. Шифры с динамическим ключом могут использоваться многими различными способами, в том числе и более сложными, но основная идея в них одна и та же.

Скрытие шифра – это передача сообщения внутри другого сообщения. Например, я и мой суперсекретный шпион договариваемся, что значением ключа будет являться каждое третье слово в сообщении. Когда я получаю от него сообщение, я выделяю и записываю каждое третье слово. Предположим, я получаю от него сообщение, в котором написано «The saying, 'The time is right' is not cow language, so is now a dead subject». Поскольку ключом является выбор каждого третьего слова, я получаю сообщение «The right cow is dead».

ПРИМЕЧАНИЕ. Скрытие шифра также называют нулевым шифром (null cipher), это является разновидностью стеганографии. Стеганография описана далее в этом разделе.

Не имеет смысла особо выделять два этих типа шифров, поскольку роли их алгоритмов и ключей остаются у них теми же самыми, несмотря на отсутствие в них математических формул. В шифре с динамическим ключом алгоритмом может быть заранее определенный набор книг. Ключ указывает на книгу, страницу, строку и слово (букву) в строке. В шифрах подстановки алгоритмом является замена букв с использованием заранее определенного алфавита или последовательности символов, а ключ указывает, каким образом каждый символ должен заменяться на другой символ. В математических алгоритмах, алгоритмом является набор математических функций, которые должны выполняться над сообщением, а ключ может указывать на то, в какой последовательности эти функции должны выполняться. Даже если атакующий узнает алгоритм и поймет, как он работает, если он не знает ключа, сообщение будет бесполезно для него.

2.6. Стеганография

Стеганография – это метод скрытия данных в данных другого типа, что очень эффективно. Только отправитель и получатель могут увидеть передаваемое таким образом сообщение, поскольку оно скрыто в графическом или звуковом файле, другом документе и т.п. Сообщение не зашифровано, оно просто скрыто. Зашифрованное сообщение само по себе может привлечь внимание злоумышленника, который может решить, что в нем есть что-то важное. А с помощью стеганографии, секретное сообщение может быть скрыто в фотографии бабушки, которая не вызовет интереса у злоумышленника, хотя внутри файла с этой фотографией может передаваться то же самое секретное сообщение. Стеганография является разновидностью «безопасности посредством неизвестности».

Стеганография выполняет скрытие информации внутри компьютерных файлов. Информация может быть скрыта внутри документов, графических файлов, исполняемых файлов программ, внутри протоколов, в неиспользуемом пространстве на жестком диске или секторах, помеченных испорченными. Аудио-, видео- и графические файлы являются идеальным форматом для стеганографии, поскольку обычно они имеют большой размер. Например, отправитель может взять совершенно безобидный графический файл и изменить цвет каждого сотого пикселя на буквы передаваемого сообщения. Такую замену очень сложно заметить, если специально не искать ее.

Давайте рассмотрим отдельные составляющие, используемые стеганографией:

- **Носитель (carrier).** Сигнал, поток данных или файл, внутри которого скрыта передаваемая информация.
- **Стеганографическая среда (stego-medium).** Среда, в которой скрыта информация.
- **Полезная нагрузка (payload).** Собственно информация, которая должна быть скрыта

и передана.

Таким образом, секретное сообщение, которое вам нужно безопасно передать, является полезной нагрузкой. Изображение, в которое вы «встраиваете» ваше сообщение, является файлом-носителем, а стеганографической средой является формат этого графического файла, например, JPEG.

Метод встраивания сообщения в некую среду использует подход **самого младшего бита** (LSB – least significant bit). Многие типы файлов содержат некоторое количество битов, которые могут быть изменены без заметного влияния на содержимое файла в целом. Это именно те биты, в которых могут быть скрыты секретные данные. Для скрытия информации с использованием подхода LSB лучше всего использовать графические файлы высокого разрешения, либо звуковые файлы с высоким битрейтом. В таких файлах производимые изменения отдельных битов практически не заметны и они не ведут к увеличению размера этих файлов. Изображение с 24-битным цветом использует по 8 бит для представления каждого из трех цветов (красный, зеленый и синий) каждого из пикселей изображения. Это 256 градаций каждого из цветов. Человеческий глаз не сможет различить рядом стоящие значения цветов (например, градации синего цвета 11111111 и 11111110). Поэтому в данном случае может использоваться метод самого младшего бита для хранения информации, не имеющей отношения к цвету.

Цифровое изображение – это просто файл, в котором хранится информация о цветах отдельных пикселей, составляющих это изображение. Чем больше файл, тем больше битов в нем может быть незаметно изменено.

Существует множество различных инструментов для скрытия сообщений внутри файлов-носителей.

Скрытый шифр (нулевой шифр), рассмотренный ранее, является примером стеганографического метода. Нулевые значения не являются частью секретного сообщения, они просто используются для скрытия секретного сообщения.

Но что если вам нужно передать мне секретное сообщение в нецифровом формате? Вы должны использовать физические методы передачи секретного сообщения, вместо компьютерных. Вы можете написать сообщение невидимыми чернилами, а мне нужно будет применить определенные химические вещества, чтобы прочесть это сообщение. Вы можете сделать очень маленькую фотографию сообщения, называемую **микрофотоснимком** (microdot), и поместить ее на изображении штампа. Другим вариантом физической стеганографии является передача очень сложной картины, на которую может быть нанесено секретное сообщение, увидеть которое можно только при определенном освещении, глядя на картину под определенным углом. Все это является примерами способов передачи сообщений, защищенных стеганографическими методами.

Цифровые водяные знаки. Вам встречались копии документов, на которые нанесен логотип или торговая марка некоторой компании? Встроенные в компьютерные файлы логотипы и торговые марки называются **цифровыми водяными знаками** (digital watermark). В отличие от секретных сообщений, которые встраиваются таким образом, чтобы оставаться незаметными, цифровые водяные знаки, как правило, делают заметными. Нанесенные цифровые водяные знаки предупреждают людей от несанкционированного использования цифровых материалов, не являющихся их собственностью. Такую разновидность стеганографии называют **управлением правами на цифровые материалы** (DRM – Digital Rights Management). DRM применяют для ограничения использования материалов, являющихся собственностью конкретной компании или человека.

Ссылки по теме:

- Steganography and digital watermarking resource links, Johnson & Johnson Technology Consultants
- “Steganography Revealed,” by Kristy Westphal, SecurityFocus (April 9, 2003)

3. Типы шифров

Симметричные шифры делятся на два основных типа: подстановки (substitution) и перестановки (transposition, permutation). **Шифры подстановки** заменяют биты, символы или блоки на другие биты, символы или блоки. **Шифры перестановки** не меняют исходный текст, вместо этого они перемещают исходные значения внутри исходного текста – они переставляют биты, символы или блоки символов для скрытия первоначального смысла.

3.1. Шифры подстановки

Шифры подстановки используют ключ, который указывает, как следует выполнять подстановку. В **шифре Цезаря** каждый символ заменялся символом, расположенным на три позиции дальше него в алфавите. Алгоритмом был алфавит, а ключом – инструкция «сдвигать на три символа».

Подстановка используется современными симметричными алгоритмами, но это сложно сравнить с таким простейшим методом, как шифр Цезаря. Однако шифр Цезаря является простым и наглядным примером концепции работы шифра подстановки.

3.2. Шифры перестановки

В шифре перестановки значение перемешивается (scrambled) или ставится в другом порядке. Ключ определяет позицию, на которую следует переместить значение, как показано на Рисунке 6-6.

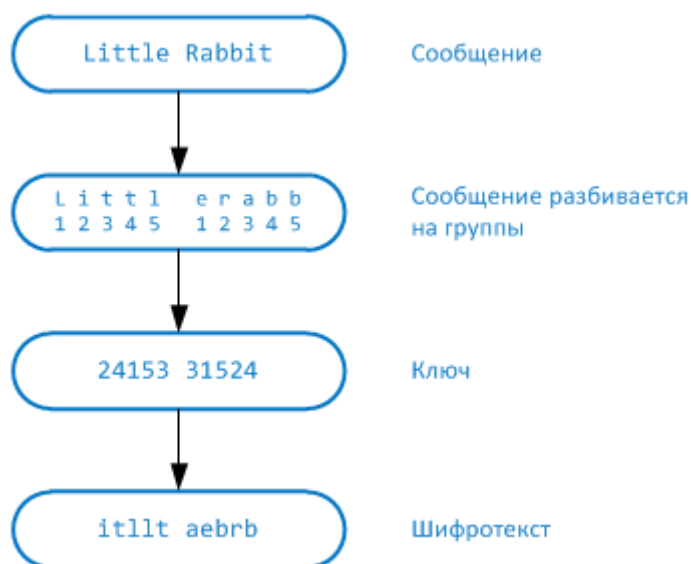


Рисунок 6-6. Шифр перестановки

Это простейший пример шифра перестановки, он показывает только способ выполнения перестановки. Если применяются сложные математические функции, перестановка может стать достаточно сложной для взлома. Современные симметричные алгоритмы используют одновременно и длинные последовательности сложных подстановок и перестановок символов шифруемого сообщения. Алгоритм содержит *возможные* способы для процессов подстановки и перестановки (представленные в математических формулах). Ключ является инструкциями для алгоритма, точно указывая, как *должна* происходить обработка и в какой последовательности. Чтобы понять связь между алгоритмом и ключом, взгляните на Рисунок 6-7. Образно говоря, алгоритм создает различные ящики, каждый из которых имеет свой (отличный от других) набор математических формул, указывающих шаги подстановки и перестановки, которые должны быть совершены над попадающими в этот ящик битами. Для шифрования сообщения, значение каждого бита должно пройти через различные ящики. Однако если каждое наше сообщение будет проходить через один и тот же набор ящиков в одинаковой последовательности, злоумышленник легко сможет провести обратный

инжиниринг этого процесса, взломать шифр и получить открытый текст нашего сообщения.

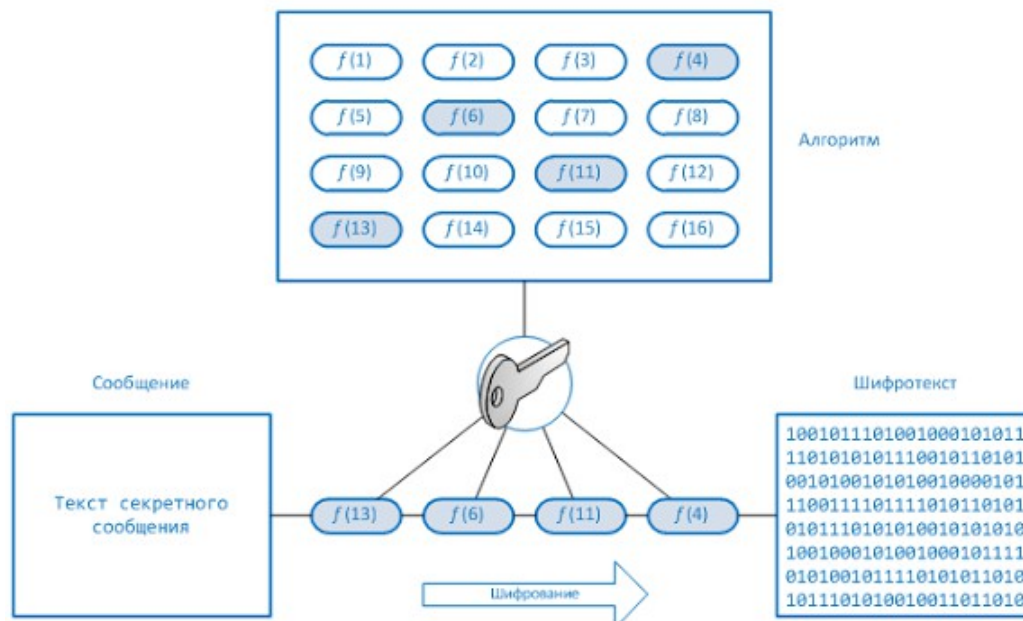


Рисунок 6-7. Связь ключа и алгоритма

Чтобы помешать злоумышленнику, используется ключ, представляющий из себя набор значений, которые указывают, какие ящики должны использоваться, в какой последовательности и с какими значениями. Так, если сообщение А шифруется ключом 1, ключ требует, чтобы сообщение прошло через ящики 1, 6, 4 и 5. Когда нам нужно зашифровать сообщение В, мы используем ключ 2, который требует, чтобы сообщение прошло через ящики 8, 3, 2 и 9. Ключ добавляет случайность и секретность в процесс шифрования.

Простые шифры подстановки и перестановки уязвимы к атакам, выполняющим **частотный анализ** (frequency analysis). В каждом языке некоторые слова и шаблоны используются чаще, чем другие. Например, в тексте на английском языке обычно чаще используется буква «е». При выполнении частотного анализа сообщения, взломщик ищет самые часто повторяющиеся шаблоны из 8 бит (составляющих символ). Если в коротком сообщении он нашел, например, 12 восьмибитных шаблонов, он может сделать вывод, что это вероятнее всего буква «е» - самая часто используемая буква в языке. Теперь взломщик может заменить эти биты на букву «е». Это даст ему опору в процессе, который позволит ему провести обратный инжиниринг и восстановить исходное сообщение.

Современные симметричные алгоритмы используют в процессе шифрования методы подстановки и перестановки, но при этом используется (должна использоваться) слишком сложная математика, чтобы позволить быть успешной такой простейшей атаке частотного анализа.

Функции генерации ключей. Для генерации сложных ключей обычно сначала создается мастер-ключ, на основе которого затем генерируются симметричные ключи. Например, если приложение отвечает за создание сеансового ключа для каждого обратившегося к нему субъекта, оно не должно просто раздавать экземпляры одного и того же ключа. Различным субъектам при каждом соединении нужны различные симметричные ключи, чтобы минимизировать продолжительность времени их использования. Даже если атакующий перехватит трафик и взломает ключ, он сможет ознакомиться с переданной информацией только в пределах соответствующего сеанса. В новом сеансе будет использоваться другой ключ. Если два или более ключей формируются на основе мастер-ключа, они называются **субключами** (subkey).

Функции генерации ключей (KDF – key derivation function) используются для генерации ключей, состоящих из случайных значений. Различные значения могут использоваться независимо или совместно в качестве случайного ключевого материала. Созданы алгоритмы, использующие

определенные хэши, пароли и/или «соль», которые много раз проходят через математические функции, указанные алгоритмом. Чем больше раз этот ключевой материал пройдет через указанные функции, тем больший уровень уверенности и безопасности сможет обеспечить криптосистема в целом.

ПРИМЕЧАНИЕ. Помните, что алгоритм остается статичным. Случайность процессов криптографии обеспечивается в основном за счет ключевого материала.

4. Методы шифрования

Хотя процесс шифрования состоит из множества частей, можно выделить две его основные части, которыми являются алгоритмы и ключи. Как было сказано ранее, алгоритмы, используемые в компьютерных системах, являются сложными математическими формулами, диктующими правила преобразования открытого текста в шифротекст. Ключ является строкой случайных битов, которая используется алгоритмом для добавления случайности в процесс шифрования. Чтобы два субъекта могли взаимодействовать с использованием шифрования, они должны использовать один и тот же алгоритм и, в ряде случаев, один и тот же ключ. В некоторых технологиях шифрования получатель и отправитель используют один и тот же ключ, тогда как в других технологиях они должны использовать различные, но связанные ключи для зашифрования и расшифрования информации. Следующие разделы объясняют различия двумя этими типами методов шифрования.

4.1. Симметричные и Асимметричные алгоритмы

Криптографические алгоритмы делятся на **симметричные алгоритмы**, которые используют симметричные ключи (также называемые секретными ключами (secret key)), и **асимметричные алгоритмы**, которые используют асимметричные ключи (называемые также открытыми (public key) и закрытыми ключами (private key)).

Симметричная криптография

В криптосистеме, в которой применяется симметричная криптография, отправитель и получатель используют два экземпляра одного и того же ключа для зашифрования и расшифрования информации, как показано на Рисунке 6-8. Таким образом, ключ имеет двойную функциональность и применяется как в процессе зашифрования, так и в процессе расшифрования. Симметричные ключи также называют *секретными ключами*, т.к. этот тип шифрования предполагает, что каждый из пользователей хранит ключ в секрете и надлежащим образом защищает его. Если атакующий получит этот ключ, он сможет расшифровать с его помощью любое перехваченное зашифрованное на нем сообщение.

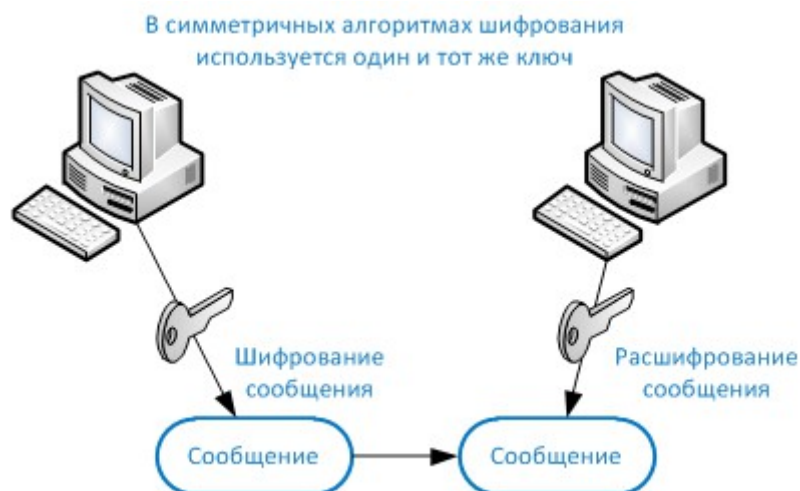


Рисунок 6-8. При использовании симметричного алгоритма, отправитель и получатель используют один и тот же ключ для зашифрования и расшифрования данных

Каждой паре пользователей, для защищенного с помощью симметричной криптографии

обмена данными, требуется два экземпляра одного и того же ключа. Например, если Дену и Ирине нужно обмениваться данными, им обоим нужно получить копию одного ключа. Если Ден хочет также с использованием симметричной криптографии взаимодействовать с Нормом и Дейвом, ему нужно иметь три отдельных ключа – по одному на каждого друга. Это не является большой проблемой, пока Дену не потребуется взаимодействовать с сотней других людей за несколько месяцев и сохранять историю переписки. Ведь это потребует использования соответствующего ключа для переписки с каждым конкретным получателем. В таком случае это может стать сложнейшей задачей. Если десяти людям необходимо безопасно обмениваться данными друг с другом с использованием симметричной криптографии, им потребуется 45 ключей. Если же взаимодействовать нужно ста людям, им потребуется 4950 ключей. Формула для расчета необходимого количества симметричных ключей выглядит следующим образом:

$$\text{Число ключей} = N(N - 1)/2, \text{ где } N - \text{число абонентов}$$

При использовании симметричных алгоритмов отправитель и получатель используют один и тот же ключ для процессов зашифрования и расшифрования информации. Безопасность таких алгоритмов полностью зависит от того, насколько хорошо пользователи защищают ключи. В таком случае безопасность полностью зависит от персонала, который должен хранить свои ключи в секрете. Если ключ скомпрометирован, все сообщения, зашифрованные на этом ключе, могут быть расшифрованы и прочитаны злоумышленником. В действительности, это еще больше усложняется, поскольку ключи необходимо безопасно распространять и обновлять их при необходимости. Если Дену нужно взаимодействовать с Нормом впервые, Ден должен решить, каким образом безопасно передать Норму ключ. Если он сделает это небезопасно, например, просто отправив ключ по электронной почте, этот ключ может быть легко перехвачен и использован злоумышленником. Поэтому Ден должен передать ключ Норму нестандартным способом. К примеру, Ден может записать ключ на флеш-накопитель и положить его на стол Норму или отправить его Норму с доверенным курьером. Процесс распространения симметричных ключей может стать очень сложной и громоздкой задачей.

Поскольку оба пользователя используют один и тот же ключ для зашифрования и расшифрования сообщений, симметричные криптосистемы могут обеспечить конфиденциальность, но не аутентификацию или неотказуемость. Такой криптографический алгоритм не позволит доказать, кто реально отправил сообщение, т.к. оба пользователя используют один и тот же ключ.

Но если симметричные криптосистемы имеют столько недостатков и проблем, почему они используются почти повсеместно? Потому что они обеспечивают очень высокую скорость обработки данных и их очень трудно взломать. Симметричные алгоритмы гораздо быстрее асимметричных. Они могут сравнительно быстро зашифровывать и расшифровывать большие объемы данных. Кроме того, данные, зашифрованные симметричным алгоритмом с использованием длинного ключа, очень сложно вскрыть.

Следующий список описывает сильные и слабые стороны криптосистем с симметричными ключами:

Сильные стороны:

- Гораздо быстрее асимметричных систем
- При использовании длинного ключа сложно взломать

Слабые стороны:

- Требуется безопасного механизма передачи ключей
- Каждой паре пользователей нужен уникальный ключ; по мере увеличения количества

пользователей, возрастающее число ключей может сделать управление ими просто нереальной задачей

- Обеспечивает конфиденциальность, но не обеспечивает аутентификацию или неотказуемость

Ниже приведены некоторые примеры симметричных алгоритмов, которые будут подробно рассмотрены позднее в разделе «Блочные и поточные шифры».

- Data Encryption Standard (DES)
- Triple-DES (3DES)
- Blowfish
- IDEA
- RC4, RC5 и RC6
- Advanced Encryption Standard (AES)

Ссылки по теме:

- Security in Open Systems, Node 208, “Symmetric Key Cryptography,” by Paul Markovitz, NIST Special Publication 800-7 (July 1994)
- Understanding the Public Key Cryptography

Асимметричная криптография

В криптографии с симметричными ключами для зашифрования и расшифрования используется один и тот же секретный ключ, тогда как в системах с открытыми ключами для этих целей используются различные (*асимметричные*) ключи. При этом два отличающихся асимметричных ключа связаны между собой математически. Если сообщение зашифровано одним ключом, для его расшифрования требуется другой ключ.

В системах с открытыми ключами, создается пара ключей, один из которых является закрытым, другой – открытым. *Открытый ключ* (public key) может быть известен всем, а *закрытый ключ* (private key) должен знать только его владелец. Часто открытые ключи хранятся в каталогах и базах данных адресов электронной почты, общедоступных всем желающим использовать эти ключи для зашифрования и расшифрования данных при взаимодействии с отдельными людьми. Рисунок 6-9 иллюстрирует использование отличающихся асимметричных ключей.

Открытый и закрытый ключи асимметричной криптосистемы математически связаны, однако наличие у кого-то открытого ключа другого человека не позволяет узнать соответствующий ему закрытый ключ. Таким образом, если злоумышленник получит копию открытого ключа Боба, это вовсе не значит, что он с помощью какого-то математического волшебства сможет получить соответствующий ему закрытый ключ Боба. Однако, если кто-то получит закрытый ключ Боба, возникнет большая проблема. Поэтому никто кроме владельца не должен иметь доступа к закрытому ключу.

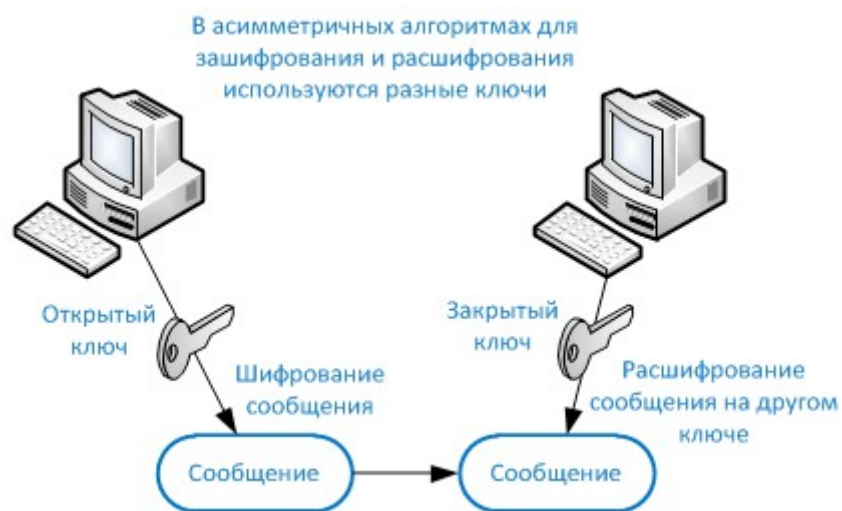


Рисунок 6-9. Асимметричная криптосистема

Если Боб зашифровал данные на своем закрытом ключе, получателю потребуется открытый ключ Боба, чтобы расшифровать их. Получатель может не только расшифровать сообщение Боба, но и ответить Бобу зашифрованным сообщением. Для этого ему нужно зашифровать свой ответ на открытом ключе Боба, тогда Боб сможет расшифровать этот ответ с помощью своего закрытого ключа. При использовании асимметричного алгоритма, невозможно зашифровывать и расшифровывать сообщение одним и тем же ключом, эти ключи, хотя и связаны математически, они не совпадают (в отличие от симметричных алгоритмов). Боб может зашифровать данные на своем закрытом ключе, тогда получатель сможет расшифровать их на открытом ключе Боба. Расшифровывая сообщение на открытом ключе Боба, получатель может быть уверен, что сообщение действительно исходит от Боба, ведь сообщение может быть расшифровано на открытом ключе Боба только в том случае, если оно было зашифровано на соответствующем закрытом ключе Боба. Это обеспечивает возможность аутентификации, т.к. Боб является (предположительно) единственным, кто имеет этот закрытый ключ. Если получатель хочет быть уверен, что единственным, кто сможет прочесть его ответ, будет Боб, он должен зашифровать свое сообщение Бобу на его открытом ключе. Тогда только Боб сможет расшифровать это сообщение, поскольку только у него есть необходимый для этого закрытый ключ.

Кроме того, получатель может решить зашифровать данные на своем закрытом ключе, а не на открытом ключе Боба. Что это ему даст? Аутентификацию. Боб будет знать, что сообщение пришло от него и не могло придти ни от кого другого. Если он зашифровывает данные на открытом ключе Боба, это не обеспечит аутентификацию, т.к. кто угодно может получить открытый ключ Боба. Если он использует свой закрытый ключ для зашифрования данных, тогда Боб может быть уверен, что сообщение исходит именно от него. Симметричные ключи не обеспечивают аутентификацию, т.к. обе стороны используют один и тот же ключ, что не может гарантировать, что сообщение исходит от конкретного человека.

Если отправителю в большей степени важна конфиденциальность передаваемой информации, ему следует зашифровать свое сообщение на открытом ключе получателя. Это называют **безопасным форматом сообщения** (secure message format), поскольку только человек, имеющий соответствующий закрытый ключ, сможет расшифровать это сообщение.

Если же отправителю в большей степени важна аутентификация, ему следует зашифровывать передаваемые данные на своем закрытом ключе. Это позволит получателю быть уверенным в том, что зашифровал данные именно тот человек, который имеет соответствующий закрытый ключ. Если отправитель шифрует данные на открытом ключе получателя, это не обеспечивает возможность аутентификации, т.к. открытый ключ доступен

всем.

Шифрование данных на закрытом ключе отправителя называют **открытым форматом сообщения** (open message format), т.к. любой человек может расшифровать эти данные с помощью общедоступного открытого ключа отправителя. Конфиденциальность при этом не обеспечивается.

Оба ключа, как закрытый, так и открытый, могут использоваться и для зашифрования, и для расшифрования данных. Не подумайте, что открытый ключ нужен только для зашифрования, а закрытый – только для расшифрования. При этом следует понимать, что если данные зашифрованы на закрытом ключе, они не могут быть расшифрованы на нем же.

Зашифрованные на закрытом ключе данные могут быть расшифрованы на соответствующем ему открытом ключе. И наоборот.

Асимметричный алгоритм работает медленнее, чем симметричный алгоритм, т.к. симметричные алгоритмы выполняют относительно простые математические функции над битами в процессах зашифрования и расшифрования. Они заменяют и перемешивают (перемещают) биты, что не очень сложно и не сильно загружает процессор. Причина их устойчивости к взлому заключается в том, что они выполняют эти функции много раз. Таким образом, в симметричных алгоритмах набор битов проходит более длинную серию замен и перестановок.

Асимметричные алгоритмы медленнее симметричных алгоритмов, т.к. они используют гораздо более сложную математику для выполнения своих функций, что требует больше процессорного времени. Однако асимметричные алгоритмы могут обеспечить аутентификацию и неотказуемость в зависимости от используемого алгоритма. Кроме того, асимметричные системы позволяют использовать более простой и управляемый процесс распространения ключей, по сравнению с симметричными системами и не имеют проблем с масштабируемостью, которые есть у симметричных систем. Причина этих различий в том, что при использовании асимметричных систем вы можете отправлять свой открытый ключ всем людям, с которыми вы хотите взаимодействовать, а не использовать для каждого из них отдельный секретный ключ. Далее, в разделе «Гибридные методы шифрования» в этом Домене мы рассмотрим, как эти две системы могут использоваться совместно для получения наилучшего результата.

ПРИМЕЧАНИЕ. Криптография с открытым ключом – это асимметричная криптография. Эти термины взаимозаменяемы.

Ниже указаны сильные и слабые стороны алгоритмов с асимметричными ключами:

Сильные стороны

- Лучше процесс распространения ключей, чем в симметричных системах
- Лучше масштабируемость, чем в симметричных системах
- Могут обеспечить аутентификацию и неотказуемость

Слабые стороны

- Работают гораздо медленнее симметричных систем
- Выполняют сложные математические преобразования

Ниже приведены примеры алгоритмов с асимметричными ключами.

- RSA
- Криптосистема на основе эллиптических кривых (ECC – Elliptic curve cryptosystem)
- Алгоритм Диффи-Хеллмана Diffie-Hellman

- Эль Гамаль (El Gamal)
- Алгоритм цифровой подписи (DSA – Digital Signature Algorithm)
- Knapsack

Эти алгоритмы мы рассмотрим далее в этом Домене, в разделе «Типы асимметричных систем».

В Таблице 6-1 приведено краткое резюме основных отличий между симметричными и асимметричными системами.

Атрибут	Симметричный	Асимметричный
Ключи	Один ключ используется совместно двумя или более субъектами	Один субъект имеет открытый ключ, другой субъект – соответствующий ему закрытый ключ
Обмен ключами	Нестандартный защищенный механизм	Открытый ключ делается общедоступным, а закрытый ключ хранит в секрете его владелец
Скорость	Алгоритм менее сложный и поэтому быстрый	Алгоритм более сложный и поэтому медленный
Использование	Комплексное шифрование (т.е. шифрование файлов и коммуникационных каналов)	Распространение ключей и цифровые подписи
Предоставляемые сервисы безопасности	Конфиденциальность	Аутентификация и неотказуемость

Таблица 6-1. Различия между симметричными и асимметричными системами

ПРИМЕЧАНИЕ. Цифровые подписи будут рассмотрены позднее в разделе «Цифровые подписи».

Ссылки по теме:

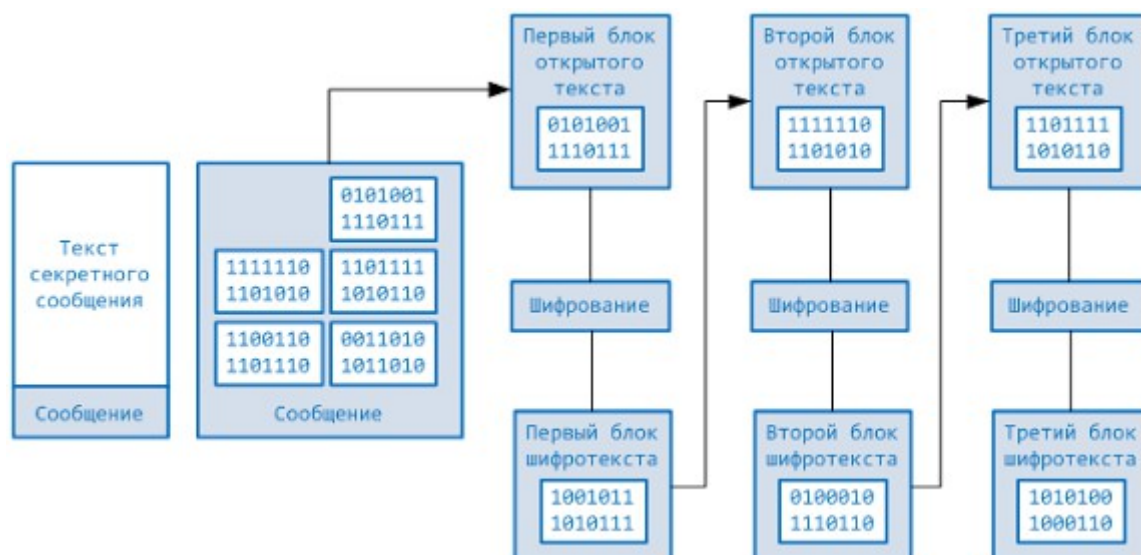
- Security in Open Systems, Node 210, “Asymmetric Key Cryptography,” by Paul Markovitz, NIST Special Publication 800-7 (July 1994)
- “Cryptography Defined/Brief History,” by Sarah Simpson (Spring 1997)
- Frequently Asked Questions About Today’s Cryptography, Version 4.1, Section 2.1.4.5, “What Is Output Feedback Mode?” by RSA Laboratories

4.2. Блочные и поточные шифры

Существует два основных типа симметричных алгоритмов: блочные шифры, которые работают с блоками битов, и потоковые шифры, которые обрабатывают по одному биту за раз.

Блочные шифры

Если для зашифрования и расшифрования данных используется **блочный шифр**, сообщение делится на блоки битов. Затем эти блоки передаются на обработку математическим функциям, по одному блоку за раз. Представьте, что вам нужно зашифровать сообщение для мамы с помощью блочного шифра, который работает с блоками по 64 бита. Длина вашего сообщения составляет 640 бит, поэтому оно делится на 10 отдельных блоков по 64 бита. Каждый блок последовательно передается на вход математической функции. Этот процесс продолжается до тех пор, пока каждый блок не будет преобразован в шифротекст. После этого вы отправляете зашифрованное сообщение вашей маме. Она использует такой же блочный шифр и тот же ключ. Эти 10 блоков шифротекста последовательно передаются в алгоритм в обратной последовательности до тех пор, пока не будет получен исходный открытый текст.



Для обеспечения стойкости шифра, в нем должны в достаточной степени использоваться два основных метода: перемешивание (confusion) и рассеивание (diffusion). **Перемешивание** обычно выполняется с помощью подстановки, тогда как **рассеивание** – с помощью перестановки. Чтобы шифр был действительно стойким, он должен использовать оба эти метода, чтобы сделать процесс обратного инжиниринга практически невозможным. На уровень перемешивания и рассеивания указывают случайность значения ключа и сложность применяемых математических функций.

В алгоритмах рассеивание может происходить как на уровне отдельных битов в блоках, так и на уровне самих блоков. Перемешивание выполняется с помощью сложных функций подстановки, чтобы злоумышленник не мог понять, каким образом заменялись исходные значения и получить оригинальный открытый текст. Представьте, что у меня есть 500 деревянных блоков, на каждый из которых нанесена буква. Я выстраиваю их в линию, чтобы составить из них сообщение (открытый текст). Затем я заменяю 300 из этих блоков блоками из другого набора (перемешивание путем подстановки). Затем я переставляю все эти блоки (рассеивание посредством перемешивания) и оставляю эту кучу. Чтобы вы смогли восстановить мое исходное предложение, вам нужно заменить блоки правильными и расставить их в правильной последовательности. Удачи!

Перемешивание выполняется для создания взаимосвязи между ключом и получаемым в результате шифротекстом. Эта взаимосвязь должна быть максимально сложной, чтобы невозможно было вскрыть ключ на основе анализа шифротекста. Каждое значение в шифротексте должно зависеть от нескольких частей ключа, но для наблюдателя эта связь между значениями ключа и значениями шифротекста должна выглядеть полностью случайной.

Рассеивание, с другой стороны, означает, что один бит открытого текста оказывает влияние на несколько бит шифротекста. Замена значения в открытом тексте должна приводить к замене нескольких значений в шифротексте, а не одного. Фактически, в действительно стойком блочном шифре, при замене одного бита в открытом тексте, должны изменяться около 50% битов в шифротексте. Т.е. при изменении всего одного бита в открытом тексте, изменится около половины шифротекста.

Блочные шифры в своих методах работы используют и перемешивание, и рассеивание. На Рисунке 6-10 показан концептуальный пример простого блочного шифра. Ему передано для обработки четыре блока длиной по четыре бита каждый. Рассматриваемый блочный алгоритм имеет два уровня четырехбитных боксов замещения, называемых S-боксами. Каждый S-бокс содержит таблицы подстановки, используемые алгоритмом в качестве инструкций по шифрованию битов.

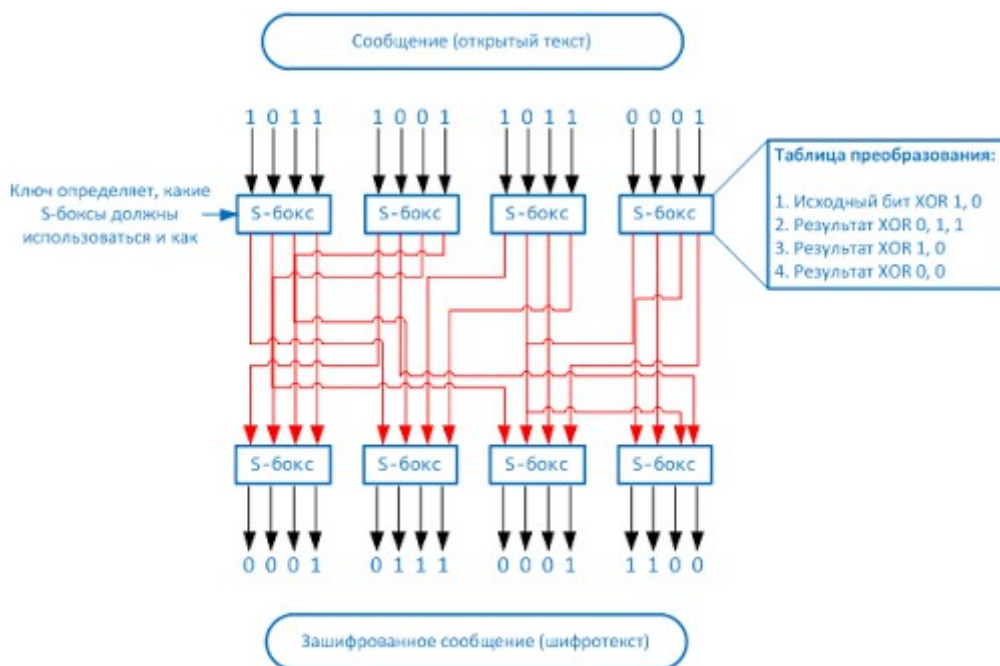


Рисунок 6-10. Сообщение разделяется на блоки битов, над которыми выполняются функции замещения и рассеивание

Ключ указывает (см. Рисунок 6-10), какие должны использоваться S-боксы в процессе перемешивания исходного сообщения из читаемого открытого текста в нечитаемый шифротекст. Каждый S-блок содержит различные методы подстановки и перестановки, которые могут быть выполнены над каждым блоком. Это очень простой пример. В реальности большинство блочных шифров работает с блоками размером по 32, 64 или 128 бит и может использовать гораздо больше S-блоков.

Поточные шифры

Как было сказано ранее, блочные шифры выполняют математические функции над блоками битов. В отличие от них, **поточные шифры** (stream cipher) не делят сообщение на блоки. Они обрабатывают сообщение, как поток битов, и выполняют математические функции над каждым битом отдельно.

При использовании поточного шифра, в процессе шифрования каждый бит открытого текста преобразуется в бит шифротекста. Поточные шифры используют генератор ключевого потока, который производит поток битов, объединяемых с помощью операции XOR с битами открытого текста, с целью получения шифротекста. Это показано на Рисунке 6-11.

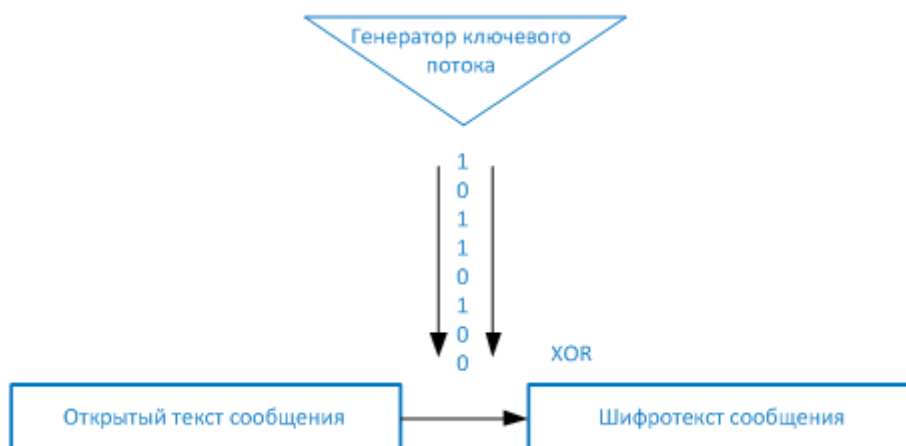


Рисунок 6-11. В поточном шифре биты, сгенерированные генератором ключевого потока, объединяются посредством XOR с битами открытого текста сообщения

ПРИМЕЧАНИЕ. Этот процесс очень похож на использование одноразовых шифровальных блокнотов, описанных ранее. Отдельные биты в одноразовом блокноте используются для шифрования отдельных битов сообщения с помощью операции XOR, а в поточном алгоритме отдельные биты создаются генератором ключевого потока, также используемым для шифрования битов сообщения с использованием операции XOR.

Если криптосистема зависит только от симметричного поточного алгоритма, атакующий может получить копию открытого текста и результирующий шифротекст, объединить их вместе с помощью операции XOR и получить в результате использованный ключевой поток, которым он может воспользоваться в дальнейшем для расшифрования других сообщений. Поэтому умные люди решили вставлять ключ в этот поток.

В блочных шифрах ключ определяет, какие функции применяются к открытому тексту и в каком порядке. Ключ обеспечивает случайность процесса шифрования. Как было сказано ранее, большинство алгоритмов шифрования является открытыми, поэтому люди знают, как они работают. Секретом является только ключ. В поточных шифрах случайность также обеспечивается с помощью ключа, делая максимально случайным поток битов, с которыми объединяется открытый текст. Эта концепция показана на Рисунке 6-12. Как вы можете увидеть на этом рисунке, и отправитель, и получатель должны иметь один и тот же ключ для генерации одинакового ключевого потока, чтобы иметь возможность правильно зашифровывать и расшифровывать информацию.



Рисунок 6-12. Отправитель и получатель должны иметь один и тот же ключ для генерации одинакового ключевого потока

Векторы инициализации

Векторы инициализации (IV – Initialization vectors) – это случайные значения, которые используются алгоритмом для обеспечения отсутствия шаблонов в процессе шифрования. Они используются совместно с ключами и их не нужно шифровать при отправке получателю. Если вектор инициализации не используется, два одинаковых открытых текста, зашифрованные на одном и том же ключе, дадут в результате один и тот же шифротекст. Такой шаблон существенно упростит задачу атакующего по взлому метода шифрования и вскрытию ключа. Если в вашем сообщении есть повторяющаяся часть (фраза или слово), вам нужно убедиться, что при шифровании каждой повторяющейся части открытого текста сообщения, создается различный шифротекст, т.е. не будет создаваться шаблон. Именно для обеспечения большей случайности в процессе шифрования и используется вектор инициализации совместно с ключом.

Стойкие и эффективные поточные шифры имеют следующие характеристики:

- **Длинные периоды неповторяющихся шаблонов в значениях ключевого потока.** Биты, генерируемые ключевым потоком должны быть случайны.
- **Статистически непредсказуемый ключевой поток.** Биты, получаемые на выходе генератора ключевого потока, не должны быть предсказуемы.
- **Ключевой поток не имеет линейной связи с ключом.** Если кто-то получил значения ключевого потока, это не должно привести к получению им значения ключа.
- **Статистически равномерный ключевой поток (примерно равное количество**

нулей и единиц). В ключевом потоке не должны преобладать нули или единицы.

Поточные шифры требуют обеспечения случайности и шифруют по одному биту за раз. Это требует больше ресурсов процессора, чем при использовании блочного шифра, поэтому поточные шифры больше подходят для реализации на аппаратном уровне. А блочные шифры, поскольку они не требуют столько ресурсов процессора, проще реализовывать на программном уровне.

ПРИМЕЧАНИЕ. Конечно, существуют и блочные шифры, реализованные на аппаратном уровне, и поточные шифры, работающие на программном уровне. Указанное выше утверждение просто является «лучшей практикой», рекомендациями по разработке и внедрению.

Поточные шифры и Одноразовые шифровальные блокноты. Поточные шифры обеспечивают тот же тип защиты, что и одноразовые шифровальные блокноты, поэтому они работают похожим образом. Поточные шифры в действительности не могут обеспечить такой же уровень защиты, как одноразовые шифровальные блокноты, т.к. они реализуются в виде программного обеспечения и автоматизированных средств. Однако за счет этого поточные шифры более практичны.

4.3. Гибридные методы шифрования

Ранее мы рассмотрели симметричные и асимметричные алгоритмы и отметили, что симметричные алгоритмы работают быстро, но имеют некоторые недостатки (плохая масштабируемость, сложное управление ключами, обеспечение только конфиденциальности), а асимметричные алгоритмы не имеют этих недостатков, но они очень медленные. Теперь рассмотрим гибридные системы, которые используют одновременно симметричные и асимметричные методы шифрования.

Совместное использование асимметричных и симметричных алгоритмов

Криптография с открытым ключом использует два ключа (открытый и закрытый), сгенерированные асимметричным алгоритмом, она применяется для защиты ключей шифрования и их распространения. Секретный ключ генерируется симметричным алгоритмом и используется для основного процесса шифрования. В этом и заключается гибридное использование двух различных алгоритмов: симметричного и асимметричного. Каждый алгоритм имеет свои преимущества и недостатки, а их совместное использование позволяет взять лучшее от каждого из них.

В гибридном подходе две эти технологии дополняют друг друга, каждая выполняет свои функции. Симметричный алгоритм создает ключи, используемые для шифрования основного объема данных, а асимметричный алгоритм создает ключи, используемые для автоматизированного распространения симметричных ключей.

Симметричный ключ используется для шифрования отправляемых вами сообщений. Когда ваш друг получает зашифрованное вами сообщение, ему нужно расшифровать его, для чего ему требуется симметричный ключ, на котором зашифровано ваше сообщение. Но вы не хотите отправлять этот ключ незащищенным образом, т.к. сообщение может быть перехвачено и незащищенный ключ может быть извлечен из него злоумышленником для последующего использования в целях расшифрования и ознакомления с вашими сообщениями. Не следует использовать для шифрования сообщений симметричный ключ, если для него не обеспечена надлежащая защита. Для обеспечения защиты симметричного ключа можно использовать асимметричный алгоритм, с помощью которого он может быть зашифрован (см. Рисунок 6-13). Но зачем нам использовать симметричный ключ для шифрования сообщений, а асимметричный ключ для шифрования симметричного ключа? Как было сказано ранее, асимметричный алгоритм работает медленно, т.к. он использует более сложную математику. А поскольку ваше сообщение, скорее всего, будет длиннее ключа, для его шифрования разумнее использовать более быстрый алгоритм (симметричный), а для шифрования ключа подойдет медленный (асимметричный), но обеспечивающий дополнительные сервисы безопасности.

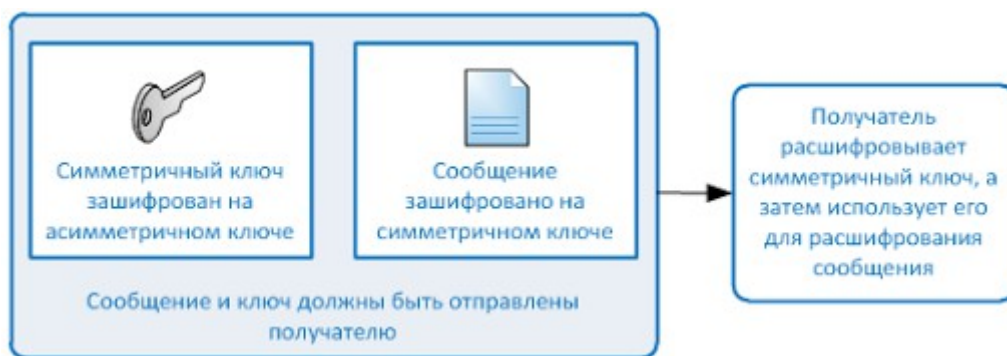
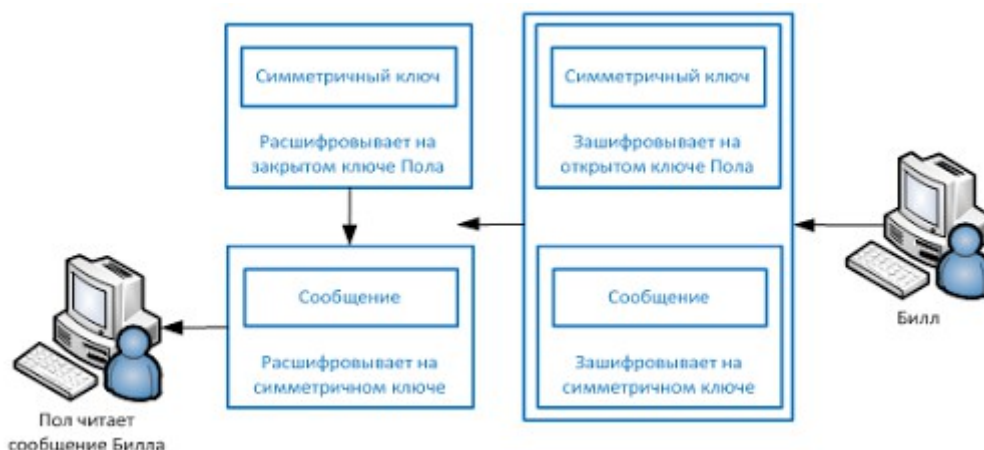


Рисунок 6-13. В гибридной системе асимметричный ключ используется для шифрования симметричного ключа, а симметричный ключ используется для шифрования сообщений

Как это работает в реальности? Предположим, что Билл отправляет Полу сообщение и хочет, чтобы только Пол мог прочитать его. Билл зашифровывает сообщение на секретном ключе, теперь он имеет шифротекст и симметричный ключ. Ключ должен быть защищен, поэтому Билл зашифровывает симметричный ключ на асимметричном ключе. Асимметричные алгоритмы используют закрытый и открытый ключи, поэтому Билл зашифровывает симметричный ключ на открытом ключе Пола. Теперь у Билла есть шифротекст сообщения и шифротекст симметричного ключа. Почему Билл зашифровал симметричный ключ на открытом ключе Пола, а не на своем закрытом ключе? Если бы Билл зашифровал его на собственном закрытом ключе, кто угодно мог бы расшифровать его на открытом ключе Билла и получить симметричный ключ. Однако Биллу не нужно, чтобы любой, имеющий его открытый ключ, мог читать его сообщения Полу. Биллу нужно, чтобы такая возможность была только у Пола. Итак, Билл зашифровал симметричный ключ на открытом ключе Пола. Если Пол хорошо защищал свой закрытый ключ, только он один сможет прочитать сообщение Билла.



Пол получает сообщение Билла и использует свой закрытый ключ, чтобы расшифровать симметричный ключ. Затем Пол использует симметричный ключ, чтобы расшифровать сообщение. Теперь Пол может прочитать важное и конфиденциальное сообщение от Билла.

Когда мы говорим, что Билл использует ключ для зашифрования сообщения, а Пол использует тот же ключ для расшифрования, это не значит, что они выполняют все эти операции вручную. Современное программное обеспечение делает все это за нас, не требуя от нас особых знаний для его использования.

Здесь все достаточно просто, вам нужно запомнить следующие аспекты:

- Асимметричный алгоритм выполняет зашифрование и расшифрование, используя закрытый и открытый ключи, которые математически связаны между собой.
- Симметричный алгоритм выполняет зашифрование и расшифрование с

использованием общего секретного ключа.

- Симметричный (секретный) ключ используется для шифрования реальных сообщений.
- Открытый ключ используется для зашифрования симметричного ключа с целью его безопасной передачи.
- Секретный ключ – это то же самое, что симметричный ключ.
- Асимметричный ключ может быть закрытым или открытым.

Итак, при использовании гибридной системы, симметричный алгоритм создает секретный ключ, используемый для шифрования данных или сообщений, а асимметричный ключ шифрует секретный ключ.

Сеансовые ключи

Сеансовый ключ (session key) – это симметричный ключ, используемый для шифрования сообщений, которыми обмениваются два пользователя. Сеансовый ключ ничем не отличается от симметричного ключа, описанного ранее, но он действителен только в рамках одного коммуникационного сеанса между пользователями.

Если у Тани есть симметричный ключ, который она постоянно использует для шифрования сообщений между ней и Лансом, этот симметричный ключ не нужно регенерировать или изменять. Они просто используют один и тот же ключ каждый раз при взаимодействии с использованием шифрования. Однако длительное повторное использование одного и того же ключа повышает вероятность его перехвата и компрометации безопасных коммуникаций. Чтобы избежать этого, следует генерировать новый симметричный ключ каждый раз, когда Тане и Лансу нужно взаимодействовать, и использовать его лишь на протяжении одного сеанса связи, а затем уничтожить (см. Рисунок 6-14). Даже если им потребуется снова взаимодействовать всего через час, будет сгенерирован новый сеансовый ключ.

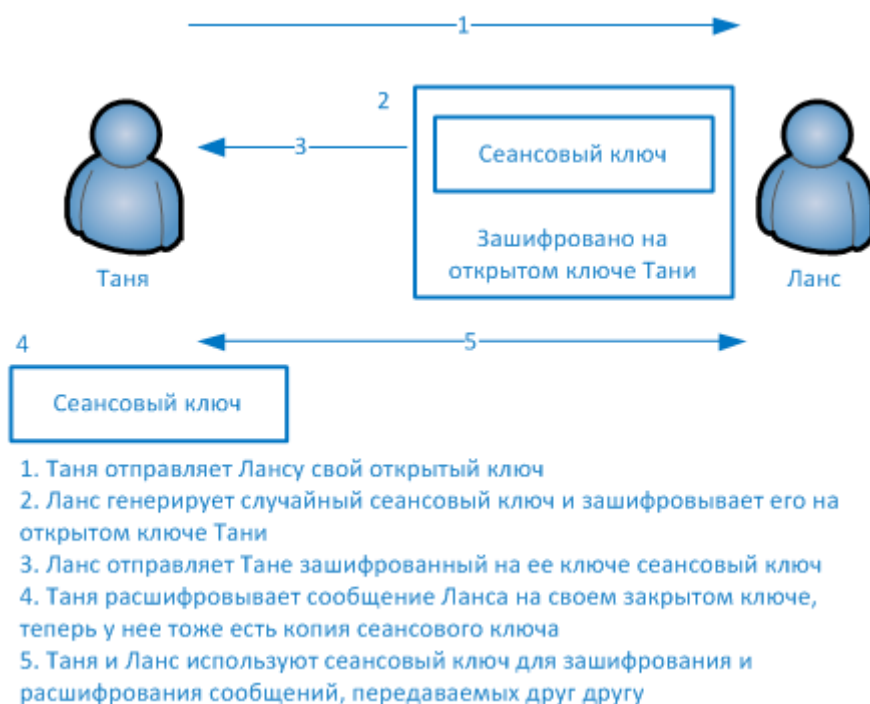
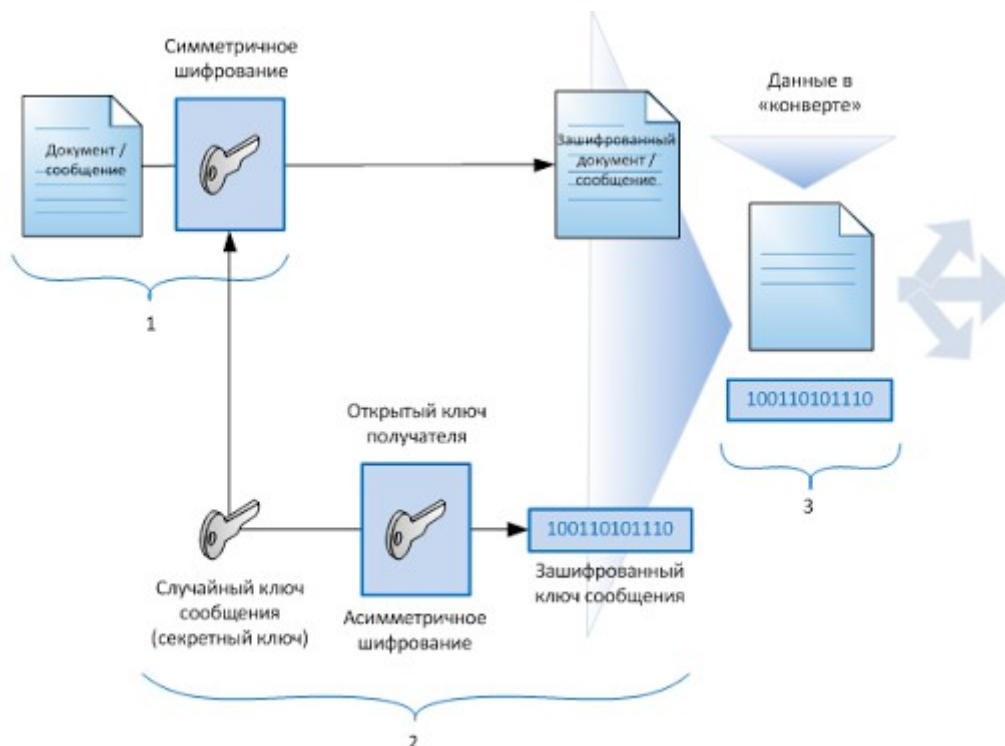


Рисунок 6-14. Сеансовый ключ генерируется для каждого сеанса взаимодействия пользователей и действует только в рамках этого сеанса

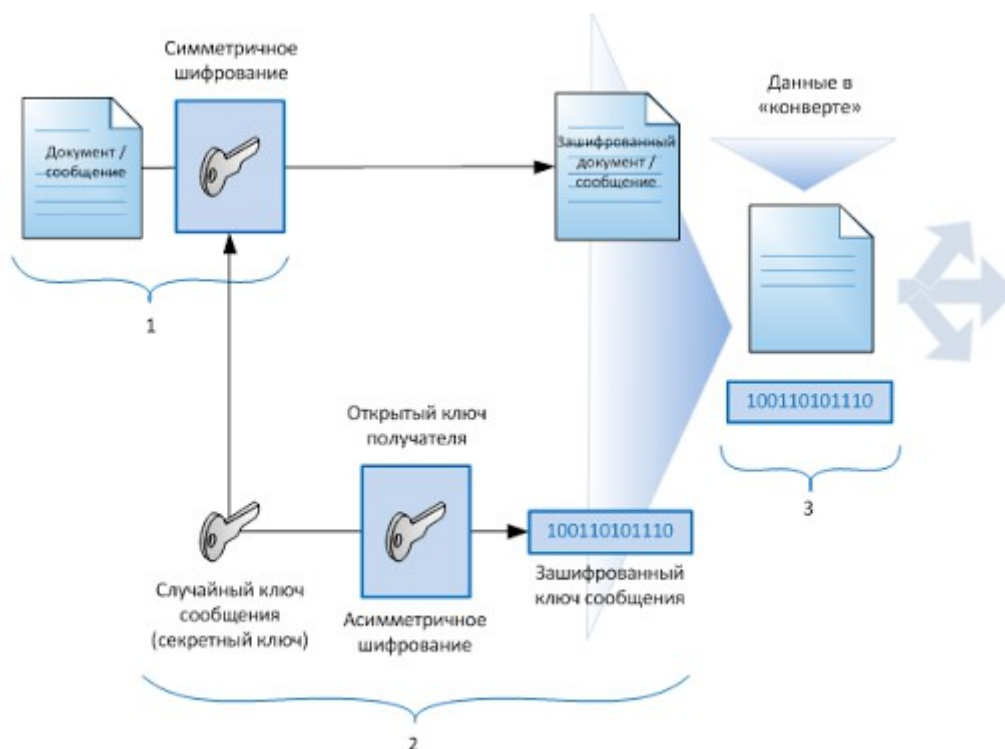
Цифровые конверты. При первом знакомстве людей с вопросами криптографии, совместное использование симметричных и асимметричных алгоритмов может вызвать непонимание. Однако эти концепции очень важно понять, поскольку они действительно являются ядром,

фундаментальными концепциями криптографии. Этот процесс используется не только в почтовом клиенте или в нескольких продуктах, он определяет весь порядок обработки данных и симметричных ключей при их передаче. Совместное использование этих двух технологий называется гибридным подходом, но у него есть и более общее название – **цифровой конверт** (digital envelope).



Сеансовый ключ обеспечивает более высокий уровень защиты, по сравнению со статичным симметричным ключом, т.к. он действителен только на один сеанс связи между двумя компьютерами. Если атакующий сможет перехватить сеансовый ключ, он сможет использовать его для несанкционированного доступа к передаваемой информации только в течение небольшого периода времени.

Если двум компьютерам нужно взаимодействовать с применением шифрования, сначала они должны пройти процесс «рукопожатия», в рамках которого они договариваются об алгоритме шифрования, который будет использоваться для передачи сеансового ключа, предназначенного для дальнейшего шифрования данных в процессе взаимодействия компьютеров. По сути, два компьютера устанавливают виртуальное соединение друг с другом, которое называют сеансом. После завершения сеанса, каждый компьютер уничтожает любые структуры данных, созданные для этого сеанса, освобождает ресурсы и, в том числе, уничтожает использованный сеансовый ключ. Эти вещи операционная система и приложения выполняют в фоновом режиме и пользователю не нужно заботиться об этом. Однако специалист по безопасности должен понимать различия между типами ключей и связанные с ними вопросы.



ПРИМЕЧАНИЕ. Закрытые и симметричные ключи не должны храниться и/или передаваться в виде открытого текста. Хотя это кажется очевидным, уже множество программных продуктов было скомпрометировано именно по этой причине.

Проблемы беспроводной безопасности. Мы рассматривали различные стандарты 802.11 и протокол WEP в Домене 05. Среди обширного списка проблем WEP, есть проблема, связанная с шифрованием данных. Если для шифрования беспроводного трафика используется только WEP, в таком случае в большинстве реализаций используется только один статистический симметричный ключ для шифрования пакетов. Одним из изменений и преимуществ стандарта 802.11i является то, что он обеспечивает шифрование каждого пакета уникальным сеансовым ключом.

5. Типы симметричных систем

В настоящее время используется несколько типов симметричных алгоритмов. Они по-разному выполняют функции зашифрования и расшифрования информации, но есть один аспект, который относится ко всем симметричным алгоритмам – отправитель и получатель должны использовать два экземпляра одного и того же ключа.

В этом разделе мы рассмотрим большинство из перечисленных ниже алгоритмов, в том числе их характеристики.

- Data Encryption Standard (DES)
- Triple DES (3DES)
- Blowfish
- Twofish
- International Data Encryption Algorithm (IDEA)
- RC4, RC5, RC6
- Advanced Encryption Standard (AES)
- SAFER
- Serpent

5.1. Data Encryption Standard

DES (Data Encryption Standard) имеет длинную и богатую историю в компьютерном мире. NIST (Национальный институт стандартов и технологий - National Institute of Standards and Technology) в течение 1960-х годов проводил исследования потребностей по защите критичных, но не классифицированных данных, и в начале 1970-х инициировал криптографическую программу. NIST пригласил производителей для оценки их криптографических алгоритмов с целью выбора наилучшего алгоритма в качестве криптографического стандарта. К тому моменту у IBM уже был разработан алгоритм шифрования, предназначенный для защиты финансовых транзакций. В 1974 году 128-битный алгоритм IBM, названный **Lucifer**, был представлен и принят. NSA (Агентство национальной безопасности - National Security Agency) модифицировало этот алгоритм для использования 64-битных ключей (при этом восемь бит использовалось для четности, что в итоге давало эффективную длину ключа 56 бит) вместо первоначальных 128-битных, и назвало его **DEA** (Data Encryption Algorithm). Возникли споры о том, с какой целью NSA ослабило Lucifer, многие склонялись к мнению, что это было сделано, чтобы NSA могло расшифровывать не предназначенные для них сообщения, но в конце концов модифицированный Lucifer стал национальным криптографическим стандартом в 1977 году и стандартом ANSI (Американский национальный институт стандартов - American National Standards Institute) в 1978 году.

ПРИМЕЧАНИЕ. DEA – это алгоритм, который и реализует DES, а DES на самом деле – просто стандарт, но обычно никто не выделяет именно алгоритм DEA, все называют его стандартом DES.

DES был реализован в большинстве коммерческих продуктов, использующих криптографию, а также в приложениях почти всех правительственных агентств. Он был протестирован и признан одним из самых стойких и эффективных среди доступных криптографических алгоритмов. Однако после ошеломительной поддержки алгоритма, его пользователей ждал большой конфуз, когда NSA в 1986 году сообщило, что с января 1988 года агентство больше не будет одобрять DES и разработанные на его основе продукты, и что он больше не соответствует федеральному стандарту 1027. NSA понимало, что поскольку DES был так популярен многие годы, он слишком долго был объектом атак и стал бесполезен в качестве официального стандарта. Многие исследователи были не согласны, но NSA хотело перейти на новый, более безопасный и менее популярный алгоритм в качестве нового стандарта.

Решение NSA прекратить поддержку DES вызвало большие проблемы и негативную реакцию. В то время DES продолжал обеспечивать необходимый уровень защиты – существующим тогда компьютерам требовались тысячи лет на взлом DES. DES уже был встроен в тысячи продуктов. Кроме того, на тот момент не было эквивалентной замены. В связи с этим, NSA пересмотрело свое решение, и NIST продлил сертификацию DES на следующие пять лет.

В 1998 году Electronic Frontier Foundation построила компьютерную систему за \$250 000, которая смогла взломать DES за три дня посредством брутфорс-атаки на все ключевое пространство. Эта система содержала 1536 процессоров, работающих на тактовой частоте 40 МГц, и выполняла 60 миллионов проверок расшифрования в секунду в расчете на один процессор. Хотя большинство людей пока не имело таких систем для выполнения подобной атаки, микропроцессоры увеличивали вычислительную мощность в соответствии с Законом Мура, и такая атака становилась все более реальной для обычного атакующего. Это обусловило появление 3DES, который обеспечивал более стойкую защиту. 3DES будет рассмотрен в следующем разделе.

Позднее DES был заменен NIST'ом на алгоритм Rijndael в качестве стандарта **AES**. Rijndael стал новым принятым методом шифрования критичной, но не классифицированной информации для правительства США, этот алгоритм широко используется в наше время.

Как работает DES?

DES – это симметричный блочный алгоритм шифрования. В него поступают 64-битные блоки открытого текста, которые преобразуются в 64-битные блоки шифротекста. Поскольку это симметричный алгоритм, в нем используется один и тот же ключ для зашифрования и расшифрования данных. Он использует 64-битные ключи: из 56 бит состоит реальный ключ, а 8 бит используются для контроля четности.

Алгоритм DES при работе с данными разделяет сообщение на блоки и работает с ними по отдельности. Блоки проходят 16 циклов применения функций перестановки и подстановки. Порядок и вид применяемых функций перестановки и подстановки зависит от значения ключа, переданного алгоритму. В результате получается 64-битный блок шифротекста.

Что значит "алгоритм взломан"?

Как уже было сказано в предыдущем разделе, DES в конце концов был взломан с помощью компьютера, названного DES Cracker. Но что в действительности означает «взломан»?

В большинстве случаев взлом алгоритма означает, что атакующий может вскрыть ключ, использованный в процессе шифрования. Предположим, что Кевин зашифровал сообщение и отправил Валери. Марк перехватил это зашифрованное сообщение и провел брутфорс-атаку на него, пытаясь расшифровать сообщение с использованием различных ключей до тех пор, пока не нашел подходящий ключ. Если такой ключ найден, алгоритм считается взломанным. Но значит ли это, что алгоритм стал бесполезен? Это зависит от того, кто ваш враг.

При взломе алгоритма с помощью брутфорс-атаки, атакующий просто пытается найти ключ, который использовался при зашифровании. Но при правильной реализации шифрования нам следует использовать сеансовый ключ, который действителен только в рамках одной сессии. В таком случае, даже если атакующий вскроет один сеансовый ключ, это может быть бесполезно для него, т.к. ему придется проделать ту же работу, чтобы вскрыть новый сеансовый ключ.

Если ваша информация имеет достаточную ценность для того, чтобы враг или вор был готов потратить много ресурсов, чтобы взломать шифр (например, в случае финансовых транзакций или военных секретов), вам не следует больше использовать взломанный алгоритм. Если вы шифруете сообщение подруге с приглашением на ужин, вы можете не волноваться о том, что алгоритм был взломан.

Для взлома алгоритма может использоваться брутфорс-атака, либо применяться анализ слабостей самого алгоритма. Эффективность брутфорс-атак повышается, т.к. растет производительность современных компьютеров. Алгоритм, использующий ключи длиной всего 40 бит, позволяет в качестве ключа использовать около триллиона различных значений. Если применяется ключ длиной 56 бит, возможных значений ключа становится уже около 72 квадриллионов. Может показаться, что это очень много, однако в связи с мощностью современных компьютеров, ключи такого размера вообще не обеспечивают какой-либо надежной защиты.

Алгоритмы строятся на основе современного понимания математики. Когда люди существенно продвинулись в математике, уровень защиты современных алгоритмов может быть сведен к нулю.

Режимы DES

Блочные шифры имеют несколько режимов работы. Каждый режим определяет способ обработки блоков. Один режим может лучше работать в одном типе среды для выполнения определенных функций, тогда как другой будет лучше работать в другой среде с абсолютно другими требованиями. Очень важно, чтобы производители, применяющие DES (или любой другой блочный шифр), понимали различные режимы его работы и знали, как правильно использовать эти режимы для разрабатываемого программного обеспечения.

DES и другие симметричные блочные шифры имеют несколько различных режимов работы,

которые могут использоваться в различных ситуациях для достижения различных результатов. Вам нужно понимать следующие пять из них:

- Электронная кодовая книга (ECB – Electronic Code Book)
- Сцепление блоков шифротекста (CBC – Cipher Block Chaining)
- Обратная связь по шифротексту (CFB – Cipher Feedback)
- Обратная связь по выводу (OFB – Output Feedback)
- Режим счетчика (CTR – Counter Mode)

Электронная кодовая книга

Режим **электронной кодовой книги** (ECB – Electronic Code Book) работает аналогично кодовой книге. 64-битные блоки данных вместе с ключом передаются в алгоритм, на выходе которого получается результирующий блок шифротекста. Для одного и того же блока открытого текста и ключа всегда производится один и тот же блок шифротекста. Длина сообщения не всегда кратна 64 битам, но ECB учитывает эту проблему. ECB – это простейший и самый быстрый режим работы, однако он имеет свои слабости.

Обычно ключ является инструкциями по использованию кодовой книги, которые обычно указывают, как именно блок текста должен шифроваться и расшифровываться. Кодовая книга обеспечивает набор команд подстановки и перестановки, которые должны быть выполнены над блоком открытого текста. Проблемой безопасности режима ECB является то, что каждый блок шифруется в точности тем же ключом, т.е. той же кодовой книгой. Это может привести к двум плохим вещам: атакующий может взломать ключ и получить возможность расшифровать все блоки данных, либо атакующий может собрать шифротекст и открытый текст для каждого блока и на основании этого создать кодовую книгу, использование которой не требует наличия ключа.

Основная проблема заключается в том, что процесс шифрования независимых блоков недостаточно случаен, поэтому этот режим проще взломать, чем другие. Тогда возникает вопрос, а зачем вообще использовать этот режим? Дело в том, что это самый быстрый и самый простой режим, его применяют для шифрования небольших объемов данных, таких как PIN-коды, ключи, значения запрос/ответ в процессах аутентификации.

Поскольку этот режим работает с блоками данных независимо, данные в файле не обязательно должны шифроваться строго в определенном порядке. Это очень полезно при шифровании баз данных. В базе данных много различных частей данных (таблиц, записей и т.п.), доступ к которым осуществляется в случайном порядке. Если база данных зашифрована в режиме ECB, любые записи или таблицы могут быть добавлены, зашифрованы, удалены или расшифрованы независимо от любых других таблиц и записей. В других режимах DES существует зависимость от текста, зашифрованного ранее. Такая зависимость создает лишние сложности при шифровании небольших объемов текста, т.к. расшифрование должно производиться в том же порядке, что и шифрование.

Поскольку ECB не использует сцепление (chaining) с предыдущим блоком, не следует использовать его для шифрования больших объемов данных, т.к. в шифротексте появится периодический шаблон.

Сцепление блоков шифротекста

В режиме ECB один и тот же блок открытого текста при шифровании на одинаковом ключе всегда будет давать в результате один и тот же шифротекст. Это позволяет атакующему выявить шаблон и приблизиться на шаг к компрометации процесса шифрования.

При использовании режима **сцепления блоков шифротекста** (CBC – Cipher Block Chaining) шаблоны не создаются, т.к. алгоритмом при шифровании каждого блока текста используется

не только ключ, но и значение, полученное при шифровании предыдущего блока, как это показано на Рисунке 6-15. В результате получается более случайный шифротекст. Это подразумевает наличие зависимости между блоками, их «сцепление», что и обуславливает название данного режима. Эффект «сцепления» каждого следующего блока с предыдущим скрывает любые шаблоны.

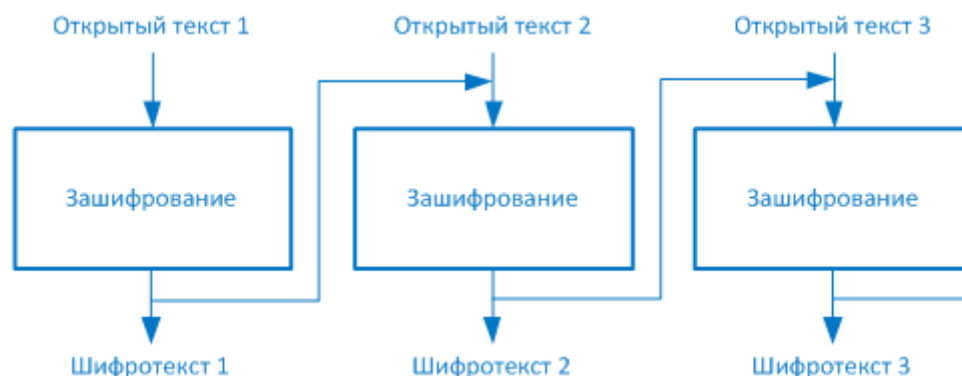


Рисунок 6-15. В режиме CBC шифротекст предыдущего блока данных используется при шифровании следующего блока данных

Результирующее значение, полученное при шифровании одного блока, объединяется с помощью операции XOR со следующим блоком перед его шифрованием, т.е. каждый предыдущий блок используется для изменения следующего блока. Таким образом, эффект «сцепления» обуславливает зависимость каждого отдельного блока шифротекста от всех блоков, обработанных перед ним, а не только от предыдущего блока.

Когда мы зашифровываем самый первый блок открытого текста в режиме CBC, у нас нет предыдущего блока шифротекста, позволяющего добавить случайность в процесс шифрования. Но если мы не добавляем случайный элемент при шифровании первого блока, плохие парни могут найти шаблон, пройти в обратном порядке и вскрыть ключ. Поэтому нам нужно использовать вектор инициализации. 64-битный вектор инициализации объединяется с помощью операции XOR с первым блоком открытого текста, а затем все это передается в процесс шифрования. Полученный результат (шифротекст) объединяется со следующим блоком открытого текста, затем зашифровывается следующий блок. Это продолжается, пока все сообщение не будет зашифровано. Это и называется сцеплением блоков шифра, обеспечивающим необходимую случайность процесса шифрования, что позволяет использовать CBC для шифрования больших файлов. В результате ни отдельный блок шифротекста, ни все зашифрованное сообщение целиком не будут содержать шаблонов, что не позволит атакующему провести обратный инжиниринг и вскрыть ключ.

Если каждый раз при зашифровании сообщения мы будем выбирать различные (неповторяющиеся) векторы инициализации, то даже при зашифровании одного и того же сообщения, получаемый в результате шифротекст будет всегда уникальным. Например, вы можете передать одно сообщение 50 людям, зашифровав его для каждого из них с использованием различных векторов инициализации. В результате передаваемый каждому из этих людей шифротекст будет различным, несмотря на то, что им передается одинаковое сообщение.

Обратная связь по шифротексту

Иногда блочный шифр может эмулировать поточный шифр. Но зачем это нужно? Например, если вы собираетесь отправить зашифрованное сообщение по электронной почте своему начальнику, ваш почтовый клиент будет использовать симметричный блочный шифр, работающий в режиме CBC. Почтовый клиент не будет использовать режим ECB, поскольку большинство сообщений являются достаточно длинными, чтобы в них в таком режиме зашифрования появился шаблон, который может использоваться для обратного

инжиниринга процесса и вскрытия ключа шифрования. Режим CBC целесообразно использовать, когда вам нужно за раз отправить большой объем данных. Но что делать, если вам нужно отправить устойчивый поток данных получателю? К примеру, если вы работаете на терминале, взаимодействующем с терминальным сервером, при каждом нажатии клавиши и движении мыши на терминальный сервер отправляется всего несколько байт для обработки этого действия. Это необходимо, чтобы происходящее выглядело так, как будто ваш компьютер (терминал) выполняет команды и обработку данных по вашим запросам, хотя в действительности все это происходит на терминальном сервере. Если вам нужно шифровать данные, передаваемые от терминала на терминальный сервер, вам не следует использовать режим CBC, т.к. он шифрует только блоки данных длиной в 64 бита. А ваши блоки данных имеют длину всего 8 бит. Именно для такой ситуации и предназначен режим *обратной связи по шифротексту* (CFB – Cipher Feedback).

Рисунок 6-16 показывает работу режима CFB, который в действительности является комбинацией блочного и поточного шифра. Для зашифрования первого восьмибитного блока мы делаем то же самое, что и в режиме CBC, использующем вектор инициализации. Вспомним, как работает поточный шифр: ключ и вектор инициализации используются алгоритмом для создания ключевого потока, который является просто множеством случайных битов. Это множество битов объединяется с помощью операции XOR с блоком открытого текста, и создает в результате блок шифротекста такого же размера. Итак, первый восьмибитовый блок объединен с набором битов, созданных генератором ключевого потока. После этого одна копия этого восьмибитового результирующего блока шифротекста отправляется получателю (в нашем примере – терминальному серверу), а другая копия используется для шифрования следующего восьмибитового блока открытого текста. Использование этой второй копии шифротекста в процесс зашифрования следующего блока, обеспечивает большую случайность процесса шифрования.



Рисунок 6-16. Блочный шифр, работающий в режиме CFB

Мы рассмотрели пример, в котором шифруются восьмибитные блоки, но в действительности режим CFB может использоваться для шифрования блоков любого размера, даже однобитовых. Однако в большинстве случаев этот режим применяется именно для шифрования восьмибитовых блоков, поскольку восемь бит представляют собой один символ.

ПРИМЕЧАНИЕ. При использовании режима CBC, хорошей идеей является использование уникального для каждого сообщения значения вектора инициализации, но это не является необходимым при шифровании очень больших сообщений. При использовании режима CFB, мы шифруем небольшие объемы данных, поэтому следует использовать новое значение вектора инициализации для зашифрования каждого нового потока данных.

Обратная связь по выводу

Как было сказано выше, вы можете использовать режим CBC для шифрования маленьких объемов данных, таких как ключи или значения PIN-кодов. Режим CBC вы можете использовать для шифрования больших объемов данных блоками по 64 бита. В случаях, когда вам нужно последовательно шифровать небольшие объемы данных, вам нужен для работы шифр, похожий на поточный шифр, шифрующий отдельные биты блоков, такой как режим CFB. Однако возникают ситуации, когда вам по-прежнему нужно шифровать небольшие объемы данных за раз (от одного до восьми бит), но вам нужно обеспечить,

чтобы возможные ошибки не оказывали влияния на процесс зашифрования и расшифрования данных.

Если вы снова взгляните на Рисунок 6-16, вы увидите, что шифротекст из предыдущего блока используется для зашифрования следующего блока открытого текста. Что произойдет, если бит в первом блоке шифротекста будет поврежден? Тогда мы получим неверное значение при зашифровании следующего блока открытого текста, и эта проблема распространится на все последующие блоки, поскольку используется режим сцепления. Теперь посмотрите на Рисунок 6-17. Он очень похож на рисунок 6-16, однако в нем значение, используемое при зашифровании следующего блока открытого текста, берется напрямую из ключевого потока, а не из предыдущего блока шифротекста. В этом и заключается разница между этими двумя режимами.



Рисунок 6-17. Блочный шифр, работающий в режиме OFB

Если вам нужно шифровать что-то очень чувствительное к ошибкам такого типа, например, цифровое видео или звук, вам не следует использовать режим CFB. Вместо него вам следует использовать режим **обратной связи по выводу** (OFB – Output Feedback), который уменьшает влияние повреждений отдельных битов.

Таким образом, OFB – это режим работы блочного шифра, применяемый при необходимости эмуляции потока (т.к. шифруются небольшие объемы данных за раз), и обеспечивающий низкую вероятность создания ошибок и их распространения на весь последующий процесс шифрования.

Для обеспечения максимально возможной защиты в режимах OFB и CFB, размер шифротекста (в CFB) или значения ключевого потока (в OFB) должны быть того же размера, что и размер блока зашифровываемого блока открытого текста. Таким образом, если вы используете CFB и зашифровываете по восемь бит за раз, созданный перед этим блок шифротекста должен также иметь длину 8 бит. Иначе начнут появляться повторяющиеся значения, создавая шаблон (по этой же самой причине одноразовый шифровальный блокнот следует использовать только один раз и хранить его ровно столько времени, сколько и само сообщение).

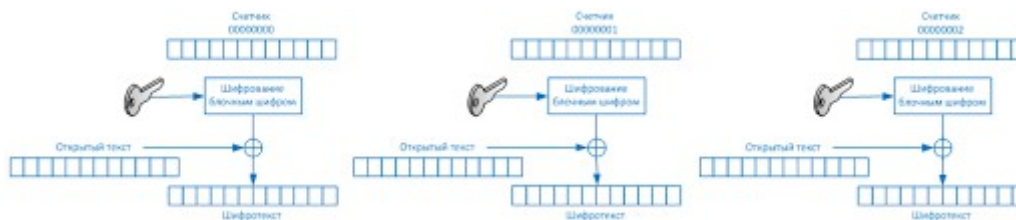
Режим счетчика

Режим счетчика (CTR – Counter Mode) очень похож на режим OFB, но вместо использования случайных уникальных значений вектора инициализации для генерации значений ключевого потока, этот режим использует счетчик, значение которого добавляется к какому-либо блоку открытого текста, который нужно зашифровать. Уникальное значение счетчика гарантирует, что каждый блок объединяется с уникальным значением ключевого потока.

Другое отличие заключается в том, что не применяется сцепление, т.е. при шифровании следующего блока не используется шифротекст предыдущего блока. Поскольку нет сцепления, шифрование отдельных блоков может происходить параллельно, что повышает производительность. Основной причиной использования режима CTR вместо других режимов является производительность.

Этот режим существует уже довольно долго и применяется при шифровании ячеек АТМ для виртуальных соединений, а также в IPSec. В настоящее время он также интегрирован в

новый беспроводной стандарт 802.11i. Разработчики выбирают этот режим в указанных ситуациях, поскольку отдельные ячейки АТМ или пакеты, проходящие через туннель IPSec или через радиоканал, могут прийти к получателю в ином порядке. Поскольку в этом режиме не используется сцепление, получатель может расшифровать и начать обрабатывать отдельные пакеты, не ожидая получения всего сообщения.



Ссылки по теме:

- “Cryptography,” by Bill Unruh
- Cryptography Research, Inc. home page

5.2. Triple-DES

Мы сразу переходим от DES к 3DES, пропуская Double-DES. Дело в том, что хотя Double-DES использует ключ длиной 112 бит, существует специальная атака против Double-DES, которая снижает фактор трудозатрат на примерно тот же уровень, что и для DES. Таким образом, он не более безопасен, чем DES.

Многие успешные атаки против DES и различных его реализаций применимы и к 3DES. NIST знал, что необходим новый стандарт (которым в итоге стал AES), однако требовалось срочное решение, обеспечивающее более высокую защиту критичных данных. Для этого был разработан 3DES.

3DES в своей работе использует 48 циклов, что существенно повышает его сопротивляемость дифференциальному криптоанализу. Однако это требует выполнения существенно большего объема работ, нанося удар по производительности. Процесс зашифрования и расшифрования данных может занимать до 3 раз больше времени, чем при использовании DES.

Однако несмотря на потери производительности и тот факт, что NIST уже выбрал алгоритм Rijndael для замены DES в качестве AES, и NIST, и многим другим был нужен 3DES в качестве временного решения.

3DES может работать в различных режимах, выбранный режим определяет количество используемых ключей, а также выполняемые функции:

- **DES-EEE3.** Использует три различных ключа в процессе шифрования. Данные последовательно зашифровываются на каждом из них (т.е. операция зашифрования выполняется трижды).
- **DES-EDE3.** Использует три различных ключа в процессе шифрования. Данные зашифровываются на первом, затем расшифровываются на втором, а потом зашифровываются на третьем.
- **DES-EEE2.** Аналогичен DES-EEE3, но использует только два ключа. На первом и третьем этапах шифрования используется один и тот же ключ.
- **DES-EDE2.** Аналогичен DES-EDE3, но использует только два ключа. На первом и третьем этапах шифрования используется один и тот же ключ.

Режим EDE на первый взгляд может выглядеть избыточным. Насколько хорошую защиту можно обеспечить путем зашифрования, расшифрования на другом ключе и еще одного зашифрования? Когда данные зашифровываются на одном симметричном ключе и

расшифровываются на другом симметричном ключе, это еще больше перемешивает данные. В действительности, данные при этом не расшифровываются на втором этапе.

5.3. Advanced Encryption Standard

После 20-летнего использования DES в качестве стандарта шифрования, он был взломан, и в срочном порядке потребовалась новая технология. NIST решил, что нужен новый стандарт – AES (Advanced Encryption Standard), чтобы заменить DES. В январе 1997 года NIST объявил конкурс для кандидатов на AES и описал требования к нему в FIPS PUB 197. В качестве кандидатов на AES рассматривались симметричные блочные шифры, поддерживающие ключи длиной 128, 192 и 256 бит. Финалистами стали следующие пять алгоритмов:

- **MARS.** Разработан командой IBM, ранее создавшей Lucifer.
- **RC6.** Разработан RSA Laboratories.
- **Serpent.** Разработан Россом Андерсоном, Эли Бихэмом и Ларсом Кнудсенем.
- **Twofish.** Разработан Counterpane Systems.
- **Rijndael.** Разработан Джоанной Деймен и Венсентом Риджменом.

Из них был выбран Rijndael. Rijndael может работать с блоками 128, 192 и 256 бит. Число циклов зависит от размера блока и длины ключа:

- Если длина ключа и блока составляет 128 бит, используется 10 циклов.
- Если длина ключа и блока составляет 192 бита, используется 12 циклов.
- Если длина ключа и блока составляет 256 бит, используется 14 циклов.

Rijndael хорошо работает как при аппаратной, так и при программной реализации во многих различных продуктах и средах. Он использует небольшой объем памяти и спроектирован таким образом, чтобы его было легко защитить от атак по времени (timing attack). NIST заменил DES на Rijndael. Сейчас этот алгоритм используется американским правительством для защиты критичной, но не классифицированной информации.

ПРИМЕЧАНИЕ. DEA – это алгоритм, использовавшийся в DES, а Rijndael – это алгоритм, который используется в AES. Но мы знаем эти алгоритмы как DES и AES, а не по их действительным названиям.

5.4. International Data Encryption Algorithm

IDEA (International Data Encryption Algorithm) – это блочный шифр, работающий с 64-битными блоками данных. 64-битный блок данных делится на 16 маленьких блоков, каждый из которых проходит через восемь циклов математических функций. Ключ имеет длину 128 бит. При программной реализации IDEA быстрее DES.

Алгоритм IDEA может работать в различных режимах, похожих на режимы, описанные ранее в разделе DES, однако он более устойчив к взлому, по сравнению с DES, поскольку использует ключи большей длины. IDEA используется в PGP, а также другом криптографическом программном обеспечении. Он вполне мог заменить DES, но он запатентован, что требует определенных лицензионных отчислений при его использовании.

На момент написания этой книги, не известно о фактах успешных атак против этого алгоритма, хотя предпринимались неоднократные попытки.

5.5. Blowfish

Blowfish – это блочный шифр, который работает с 64-битными блоками данных. Длина ключа может быть любой от 32 до 448 бит, а блоки данных проходят через 16 циклов криптографических функций. Этот алгоритм разработал Брюс Шнайер. Предполагалось, что именно Blowfish заменит устаревающий DES. Blowfish общедоступен и не запатентован, чем

выгодно отличался от других алгоритмов, защищенных различными патентами, либо предназначенных для хранения государственных секретов.

5.6. RC4

RC4 является одним из самых популярных среди реализованных поточных шифров. RC4 использует ключи различной длины. Он применяется в протоколе SSL, был реализован (неудачно) в протоколе WEP. RC4 был разработан в 1987 году Роном Ривестом и являлся собственностью компании RSA Data Security, Inc., пока кто-то не опубликовал его исходный код. После этого алгоритм несколько раз внедрялся под именем ArcFour или ARC4, т.к. название RC4 является зарегистрированной торговой маркой.

Алгоритм RC4 очень прост, быстр и эффективен, из-за чего он и стал так популярен.

5.7. RC5

RC5 – это блочный шифр, использующий различные параметры, которые могут определять размер блока, длину ключа, число используемых циклов. Он был разработан Роном Ривестом и исследован RSA Data Security, Inc. В этом алгоритме могут использоваться блоки, размером 32, 64 или 128 бит, а размер ключа может достигать 2048 бит. Число циклов, выполняемое в процессе зашифрования и расшифрования, также может меняться. Оно может достигать до 255.

5.8. RC6

RC6 – это блочный шифр, созданный на основе RC5. Поэтому он имеет те же атрибуты, что и RC5. Алгоритм RC6 был разработан в качестве претендента на AES, но вместо него был выбран Rijndael. Было реализовано несколько модификаций алгоритма RC5, целью которых было повышение его скорости. Одной из таких модификаций и является RC6.

Условные обозначения в криптографии. В некоторых источниках вы можете встретить обозначения вида *rc5-w/r/b* (например, *RC5-32/12/16*). Это условное обозначение, описывающее конфигурацию алгоритма.

- *w* – половина длины блока (в битах), может быть равен 16, 32 или 64 бита
- *r* – число циклов от 0 до 255
- *b* – длина ключа (в байтах)

Таким образом, RC5-32/12/16 означает:

- Шифруются 64-битные блоки данных
- Используется 12 циклов
- Длина ключа составляет 16 байт (128 бит)

Разработчик указывает эти параметры в конкретной реализации алгоритма. Использование этих параметров дает разработчикам больше гибкости.

6. Типы асимметричных систем

Как было описано ранее в этом Домене, использование симметричной криптографии в чистом виде имеет три недостатка:

- **Сервисы безопасности.** В чистом виде криптография с симметричными ключами обеспечивает только конфиденциальность, но не аутентификацию или неотказуемость.
- **Масштабируемость.** Поскольку потребности человечества в коммуникациях растут, растет и количество необходимых симметричных ключей, которыми нужно управлять.
- **Безопасное распространение ключей.** Симметричный ключ должен быть доставлен

получателю безопасным способом.

Несмотря на эти недостатки, криптография с симметричными ключами использовалась всем компьютерным сообществом достаточно долгое время. Асимметричная криптография появились значительно позже симметричной. Мы испытывали на себе проблемы симметричной криптографии достаточно долгое время, ожидая, пока кто-то умный спасет нас от них.

6.1. Алгоритм Диффи-Хеллмана

Первым недостатком симметричной криптографии, который было решено исправить, стало безопасное распространение симметричного ключа. Над этой проблемой работали Диффи и Хеллман, которые в итоге разработали первый алгоритм с асимметричными ключами, названный их именем.

Чтобы понять, как работает алгоритм Диффи-Хеллмана, представим себе следующий пример. Таня и Эрика хотят передавать данные по зашифрованному каналу с использованием алгоритма Диффи-Хеллмана. Они обе генерируют свою пару ключей (открытый и закрытый) и обмениваются открытыми ключами. Программное обеспечение Тани берет ее закрытый ключ (являющийся просто числовым значением) и открытый ключ Эрики (другое числовое значение) и передает их в алгоритм Диффи-Хеллмана. Программное обеспечение Эрики берет ее закрытый ключ и открытый ключ Тани и также передает их в алгоритм Диффи-Хеллмана на своем компьютере. В результате на выходе из алгоритма Таня и Эрика получают одно и то же общее значение, которое используется для создания экземпляров симметричных ключей.

Таким образом, Таня и Эрика обменялись через недоверенную сеть информацией, которую не требуется защищать (их открытые ключи), а затем сгенерировали одинаковый симметричный ключ на своих системах. Теперь они обе имеют симметричный ключ для зашифрования и расшифрования передаваемой между ними информации.

ПРИМЕЧАНИЕ. Приведенный пример описывает процедуру *соглашения о ключах* (key agreement), которая может выполняться способами, отличными от *обмена ключами* (key exchange). Соответствующие функции других асимметричных алгоритмов будут обсуждаться далее в этом Доме. При выполнении функции обмена ключами отправитель зашифровывает симметричный ключ на предварительно полученном открытом ключе получателя.

Алгоритм Диффи-Хеллмана позволяет двум системам безопасно получить симметричный ключ без установления предварительных взаимоотношений или соглашений. Этот алгоритм позволяет распространять ключи, но он не обеспечивает функций шифрования и цифровой подписи. Алгоритм основан на сложности расчета дискретных логарифмов в конечном поле.

Оригинальный алгоритм Диффи-Хеллмана уязвим к атаке «человек посередине» (Man in the middle - MitM-атака), поскольку он не производит аутентификации перед обменом открытыми ключами. Вернемся к нашему примеру. При получении Эрикой открытого ключа, она не может быть уверена, что это открытый ключ Тани. Что если Ланс отправил Эрике свой открытый ключ от имени Тани? Эрика примет его ключ, думая, что это ключ Тани. Рассмотрим последовательность выполнения такой атаки, показанной на Рисунке 6-18:

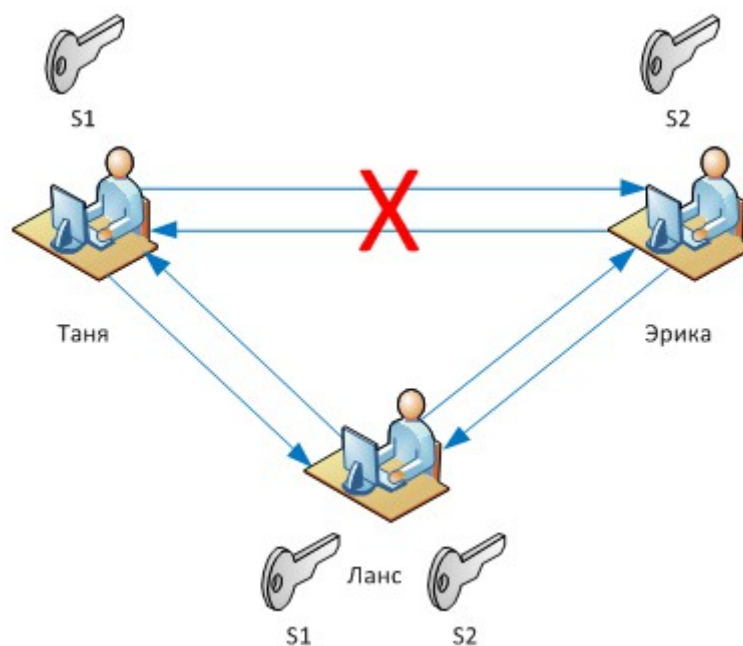


Рисунок 6-18. Атака «человек посередине»

1. Таня отправляет свой открытый ключ Эрике, но Ланс перехватывает ключ в процессе передачи и не позволяет ему дойти до Эрики.
2. Ланс отправляет свой открытый ключ Эрике, представляясь Таней. Эрика думает, что это открытый ключ Тани.
3. Эрика отправляет свой открытый ключ Тане, но Ланс также перехватывает ее ключ в процессе передачи и не позволяет ему дойти до Тани.
4. Ланс отправляет свой открытый ключ Тане, представляясь Эрикой. Таня думает, что это открытый ключ Эрики.
5. Таня объединяет с помощью алгоритма Диффи-Хеллмана свой закрытый ключ с открытым ключом Ланса и создает симметричный ключ S1.
6. Ланс объединяет свой закрытый ключ с открытым ключом Тани и создает симметричный ключ S1.
7. Эрика объединяет свой закрытый ключ с открытым ключом Ланса и создает симметричный ключ S2.
8. Ланс объединяет свой закрытый ключ с открытым ключом Эрики и создает симметричный ключ S2.
9. Теперь Таня и Ланс имеют общий симметричный ключ S1, а Эрика и Ланс имеют другой общий симметричный ключ S2. Таня и Эрика думают, что у них есть общий симметричный ключ, они не знают о вмешательстве Ланса.
10. Таня пишет сообщение Эрике, используя свой симметричный ключ S1 для зашифрования, и отправляет его.
11. Ланс перехватывает сообщение и расшифровывает его на симметричном ключе S1, читает или изменяет сообщение и перешифровывает его на симметричном ключе S2, а затем отправляет Эрике.
12. Эрика берет свой симметричный ключ S2, расшифровывает на нем сообщение и читает его, не догадываясь, что его изменил Ланс.

Контрмерой против такой атаки является проведение аутентификации перед принятием открытого ключа, что обычно обеспечивается с помощью цифровой подписи и цифровых

сертификатов.

ПРИМЕЧАНИЕ. Хотя алгоритм Диффи-Хеллмана уязвим к атаке «человек посередине», это не означает, что такой вариант компрометации возможен всегда при использовании этого алгоритма. Большинство реализаций включают в себя другие программные компоненты или протоколы, которые компенсируют эту уязвимость. Однако как специалист по безопасности, вы должны знать об этой проблеме.

ПРИМЕЧАНИЕ. MQV (Menezes-Qu-Vanstone) – это криптографическая функция соглашения о ключах аутентификации, очень похожая на алгоритм Диффи-Хеллмана. Для создания сеансовых ключей выполняется обмен открытыми ключами пользователей. Это обеспечивает защиту от получения сеансового ключа атакующим, поскольку для успешной атаки ему нужны закрытые ключи обоих пользователей.

6.2. RSA

Название **RSA** состоит из имен его создателей (Рон Ривест, Ади Шамир и Леонард Адлеман). Алгоритм RSA является алгоритмом с открытыми ключами, он стал самым популярным асимметричным алгоритмом с момента их появления. Фактически RSA является признанным во всем мире стандартом, он может использоваться для цифровой подписи, обмена ключами и шифрования. Он был разработан в 1978 году в Массачусетском технологическом институте (MIT - Massachusetts Institute of Technology) и обеспечил как аутентификацию, так и шифрование ключей.

Безопасность этого алгоритма основана на сложности разложения на множители произведения двух простых чисел. Открытый и закрытый ключи являются функцией пары больших простых чисел. Для расшифрования сообщения из шифротекста в открытый текст с использованием закрытого ключа выполняются действия, сравнимые с разложением на множители произведения двух простых чисел.

ПРИМЕЧАНИЕ. Простое число – это положительное целое число, не имеющее собственных делителей, что означает, что это число можно разделить без остатка только на единицу и само это число.

В чем заключается отличие между криптографией с открытыми ключами и инфраструктурой открытых ключей (PKI)?

Криптография с открытыми ключами использует асимметричный алгоритм. Понятия асимметричного алгоритма и криптографии с открытыми ключами взаимозаменяемы и, по сути, означают одно и то же. Примером асимметричного алгоритма является RSA, криптосистема на основе эллиптических кривых (ECC- Elliptic Curve Cryptosystem), алгоритм Диффи-Хеллмана, Эль Гамаль (El Gamal), LUC и Knapsack. Эти алгоритмы используются для создания ключевых пар (открытый/закрытый ключ), выполнения обмена ключами или соглашения о ключах, установки и проверки цифровых подписей.

Отметим, что алгоритм Диффи-Хеллмана может выполнять только соглашение о ключах и не может устанавливать и проверять цифровую подпись.

Инфраструктура открытых ключей (PKI - Public key infrastructure) – это не алгоритм, протокол или приложение – это инфраструктура, основанная на криптографии с открытыми ключами.

Одним из преимуществ RSA является то, что он может использоваться и для шифрования, и для цифровой подписи. Используя свои односторонние функции, RSA выполняет зашифрование и проверку подписи, при выполнении этих функций в обратном направлении – расшифрование и установку подписи.

В настоящее время RSA реализован во множестве приложений, его используют операционные системы Microsoft, Apple, Sun, Novell, он применяется на аппаратном уровне в сетевых картах, системах защищенной телефонии, смарт-картах. Он может использоваться в качестве *протокола обмена ключами*, т.е. для шифрования симметричного ключа с целью его безопасной передачи получателю. RSA чаще всего используется совместно с симметричным алгоритмом DES, который был заменен алгоритмом AES. Таким образом, если в качестве протокола обмена ключами используется RSA, криптосистема сначала создает

симметричный ключ для алгоритма DES или AES. Затем криптосистема зашифровывает симметричный ключ на открытом ключе получателя и отправляет его получателю. При этом симметричный ключ защищен, поскольку только человек, имеющий соответствующий закрытый ключ сможет расшифровать это сообщение и извлечь симметричный ключ.

Ныряем в математику

Криптография полностью основана на математике, используемой для преобразования данных в нечитаемый вид, а затем обратного преобразования в исходную форму, понятную человеку или компьютеру. Математика RSA основана на сложности разложения больших целых чисел на два простых сомножителя. Теперь рассмотрим, как этот алгоритм работает.

Алгоритм создает открытый ключ и закрытый ключ с помощью функции, выполняющейся над большими простыми числами. Если данные зашифрованы на открытом ключе, только с помощью соответствующего закрытого ключа можно расшифровать их. Процесс расшифрования обычно является тем же самым разложением на множители произведения двух простых чисел. Например, у меня есть секрет (зашифрованное сообщение), а вам, чтобы вскрыть этот секрет, нужно взять определенное большое число и разложить его на два сомножителя, получив в результате два числа, которые запишете у меня на листке бумаги. Кажется, что это совсем просто, но если это числа порядка 2^{300} , задача усложняется.

Следующая последовательность действий описывает работу алгоритма RSA с ключами.

1. Выбираем два случайных больших простых числа p и q .
2. Вычисляем их произведение $n = pq$.
3. Выбираем случайное целое число e ($e < 1 < (p-1)(q-1)$) в качестве ключа зашифрования. Убеждаемся, что e и $(p-1)(q-1)$ являются взаимно простыми.
4. Рассчитываем ключ расшифрования d . $ed=1 \bmod (p-1)(q-1)$ или $d=e^{-1} \bmod ((p-1)[q-1])$.
5. Открытый ключ = (n,e) .
6. Закрытый ключ = (n,d) .
7. Исходные простые числа p и q уничтожаются безопасным образом.

Теперь у нас есть открытый и закрытый ключи, но как они работают вместе?

Если вам нужно зашифровать сообщение m на вашем открытом ключе (e, n) , применяется формула $C = m^e \bmod n$. Затем вам нужно расшифровать сообщение на вашем закрытом ключе (d, n) , для этого применяется формула $M = c^d \bmod n$.

Вы можете подумать: «Чудесно. Хотя я не понял этих формул, они выглядят достаточно простыми. Интересно, почему никто не может взломать эти маленькие формулы и вскрыть зашифрованную информацию?». Может кто-то однажды и сможет. Когда люди создадут новые математические инструменты и увеличат вычислительную мощность, воспользуются достижениями криптоанализа, алгоритм RSA однажды может быть взломан. Если мы научимся быстро и легко раскладывать большие числа на их простые сомножители, этот алгоритм не сможет больше обеспечивать безопасность. Но на данный момент ничего подобного нет, и мы можем продолжать использовать RSA в нашей деятельности.

Односторонние функции

Односторонние функции (one-way function) – это математические функции, которые легче рассчитать в прямом направлении, чем в обратном. Например, бросить на пол стакан очень просто, а вот собрать потом с пола все осколки и восстановить стакан практически нереально. Эта аналогия похожа на использование односторонних функций в криптографии, в частности в алгоритме RSA и других асимметричных алгоритмах, основанных на них.

В качестве простого направления расчета односторонней функции в алгоритме RSA

используется перемножение двух больших простых чисел. Перемножить эти числа гораздо проще, чем потом разложить произведение на сомножители, чтобы получить исходные два числа. RSA основан на сложности разложения на сомножители больших чисел, являющихся произведением двух больших простых чисел. Атаки на такие криптосистемы не обязательно пытаются проверить каждое возможное значение ключа, чаще они предпринимают попытки разложить большое число на сомножители, что позволит атакующему получить закрытый ключ.

Если пользователь зашифровывает сообщение на открытом ключе, это сообщение кодируется с помощью односторонней функции (разбивающийся стакан). Эта функция имеет **черный ход** (trapdoor) (знание о том, как собрать стакан обратно), но воспользоваться черным ходом можно только если знать о нем и использовать правильный код. Закрытый ключ и является тем самым черным ходом. Закрытый ключ знает об этом черном ходе, знает как получить исходные простые числа и имеет необходимый программный код, позволяющий использовать этот секретный черный ход с целью восстановления исходного сообщения на основе шифротекста (пересборки разбитого стакана). Именно знание о черном ходе и обладание правильным функционалом для его использования и делает закрытый ключ закрытым.

При выполнении односторонней функции в прямом направлении, доступна функциональность зашифрования и проверки цифровой подписи. При выполнении односторонней функции в обратном направлении, доступна функциональность расшифрования и установки цифровой подписи. Таким образом, только открытый ключ может использоваться для зашифрования и проверки подписи, и только закрытый ключ – для расшифрования и установки подписи.

Как было сказано ранее, *фактор трудозатрат* – это объем времени и ресурсов, которые потребуются для взлома метода шифрования. В асимметричных алгоритмах фактором трудозатрат является разница во времени и усилиях, необходимых для выполнения односторонней функции в прямом направлении и обратном. В большинстве случаев увеличение длины ключа затрудняет для атакующего выполнение односторонней функции в обратном направлении (дешифрование сообщения).

Основная мысль этого раздела – это то, что все асимметричные алгоритмы обеспечивают безопасность путем использования математических уравнений, которые просто выполнять в одном направлении и практически невозможно в другом. Это основано на математической сложности этой задачи. Математическая сложность алгоритма RSA основана на сложности разложения больших чисел на исходные простые сомножители. Алгоритм Диффи-Хеллмана и Эль Гамаль основаны на сложности расчета логарифмов в конечном поле.

6.3. Эль Гамаль

Эль Гамаль (El Gamal) – это алгоритм с открытыми ключами, который может использоваться для цифровой подписи, шифрования и обмена ключами. Он основан не на сложности разложения на сомножители больших чисел, а на расчете дискретных логарифмов в конечном поле. В действительности Эль Гамаль является расширением алгоритма Диффи-Хеллмана.

Эль Гамаль обеспечивает ту же функциональность, что и другие асимметричные алгоритмы, однако его основной недостаток – это низкая производительность. По сравнению с другими алгоритмами он самый медленный.

6.4. Криптосистемы с эллиптическими кривыми

Эллиптические кривые – это богатые математические структуры, с пользой применяемые во множестве различных приложений. **Криптосистема с эллиптическими кривыми** (ECC – Elliptic Curve Cryptosystem) реализует большинство функций RSA: цифровая подпись,

безопасное распространение ключей и шифрование. Единственным отличительным фактором ECC является ее эффективность. ECC более эффективна, чем RSA и другие асимметричные алгоритмы.

На Рисунке 6-19 показан пример эллиптической кривой. В этом математическом поле, точки на кривой объединяются в структуры, называемые группами. Эти точки являются значениями, используемыми в математических формулах процессов шифрования и расшифрования в ECC. Алгоритм рассчитывает дискретные логарифмы эллиптических кривых, которые отличаются от расчета дискретных логарифмов в конечном поле (который используют Диффи-Хеллман и Эль Гамаль).

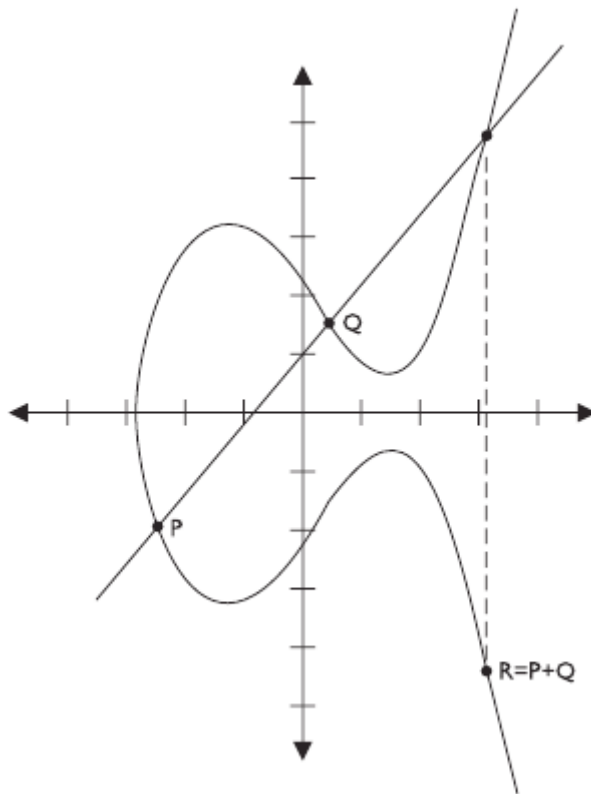


Рисунок 6-19. Эллиптические кривые

Некоторые устройства имеют ограниченные возможности для обработки и хранения данных, ограниченные источники питания и полосу пропускания, например, беспроводные устройства и сотовые телефоны. Для таких устройств крайне важна эффективность использования ресурсов. ECC выполняет функции шифрования, требуя для этого наименьшего количества ресурсов, по сравнению с RSA и другими алгоритмами. Поэтому именно ECC используется в таких устройствах.

В большинстве случаев, более длинный ключ обеспечивает лучшую защиту, но ECC может обеспечивать одинаковый с RSA уровень защиты при использовании более коротких ключей. Поскольку более длинные ключи требуют больше ресурсов для выполнения математических задач, ключи меньшей длины, используемые в ECC, требуют меньше ресурсов устройства.

6.5. LUC

Этот алгоритм основан на «последовательностях Люка» (Lucas sequences). Он выполняет расчет дискретных логарифмов в конечном поле, но использует последовательности Люка, позволяющие ускорить выполнение расчетов.

6.6. Knapsack

Со временем появились различные версии алгоритмов knapsack. Первый был разработан

Меркле и Хеллманом, он мог использоваться только для шифрования, но позднее был доработан для реализации возможностей цифровой подписи. Эти алгоритмы основаны на "задаче рюкзака" (knapsack problem) – математической дилемме, которая иллюстрирует следующий вопрос: если у вас есть несколько различных предметов, каждый из них имеет свой вес, каким образом можно уложить их максимальное количество в рюкзак, чтобы рюкзак имел определенный вес?

Этот алгоритм небезопасен, в настоящее время в криптосистемах не используется.

6.7. Доказательство с нулевым разглашением

Когда военные представляют новостным изданиям обзоры некоторых мировых событий, они преследуют только одну цель: рассказать такую историю, которую общественность ожидает услышать и ничего больше. Не предоставляйте больше информации, чем это нужно, чтобы сделать выводы, т.е. больше информации, чем они должны знать. Военные делают это, поскольку понимают, что не только хорошие парни смотрят CNN. Это пример **доказательства с нулевым разглашением** (zero knowledge proof). Вы сообщаете кому-то только ту информацию, которую он должен знать, и ничего более.

Доказательство с нулевым разглашением также используется в криптографии. Если я зашифровываю что-то на своем закрытом ключе, вы можете проверить мой закрытый ключ, расшифровав данные на моем открытом ключе. Шифруя что-то на своем закрытом ключе, я доказываю, что обладаю им, но при этом я не передаю и не показываю никому свой закрытый ключ. Только владелец закрытого ключа может таким образом доказать, что он владеет им.

Ссылки по теме:

- “Elliptic Curve Cryptography FAQ v1.12,” by George Barwood (Dec. 12, 1997)

7. Целостность сообщения

Биты четности и функции CRC (Cyclic Redundancy Check - Циклический избыточный код) используются в протоколах для выявления изменений в потоке битов, проходящем от одного компьютера к другому, но обычно они могут выявить только неумышленные изменения. Такие изменения могут произойти из-за перепадов напряжения, помех, затухания сигнала в проводах или других физических причин, вызывающих повреждение битов при их передаче между компьютерами. Биты четности не могут выявить факт перехвата сообщения злоумышленником, его изменения и последующей отправки получателю, т.к. злоумышленник может просто рассчитать новое значение четности и указать его в новом сообщении, и получатель никогда не заметит разницы. Для защиты от такой атаки необходимы алгоритмы хэширования, позволяющие успешно выявлять факты и умышленного, и неумышленного изменения данных. Сейчас мы рассмотрим некоторые алгоритмы хэширования и их характеристики.

7.1. Односторонний хэш

Односторонний хэш (one-way hash) – это функция, которая создает строки и сообщения переменной длины, а также значения фиксированной длины, называемые значениями хэша. Например, если Кевину нужно отправить сообщение Марии, и он хочет быть уверенным, что в его сообщение не будут внесены несанкционированные изменения в процессе передачи, он должен рассчитать хэш-значение своего сообщения и добавить его к самому сообщению. Когда Мария получит сообщение, она выполнит такую же функцию хэширования, что и Кевин, и сравнит полученный результат со значением, указанным в сообщении. Если два значения совпадают, Мария может быть уверена, что сообщение не было изменено в процессе передачи. Если два значения отличаются, Мария узнает, что сообщение было изменено умышленно или неумышленно, и уничтожит это сообщение.

Алгоритм хэширования не является секретом – он общеизвестен. Секретность односторонней функции хэширования обеспечивается ее «односторонностью». Эта функция работает только в прямом направлении, но не в обратном. В этом состоит ее отличие от односторонней функции в криптографии с открытыми ключами, в которой секретность обеспечивается за счет того, что никто не знает «черного хода», а выполнить одностороннюю функцию в обратном направлении, чтобы привести зашифрованное сообщение к читаемому виду, очень сложно. Однако односторонние функции хэширования никогда не выполняются в обратном направлении – получатель не пытается провести обратный процесс, он просто запускает ту же самую функцию хэширования в прямом направлении над тем же самым сообщением, чтобы сравнить результаты.

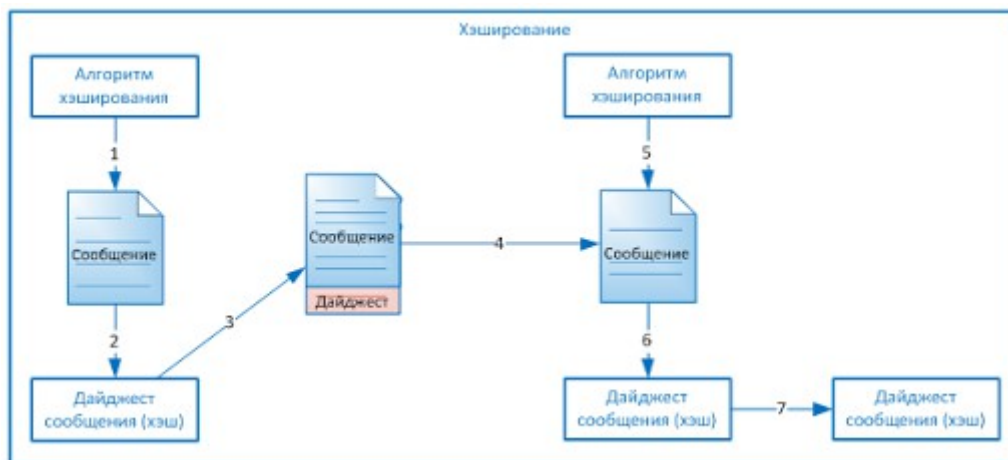
Односторонняя функция хэширования выполняется без использования каких-либо ключей. Это означает, например, что Черил пишет сообщение, рассчитывает дайджест сообщения, добавляет его к самому сообщению и отправляет сообщение вместе с дайджестом Скотту. Брюс может перехватить это сообщение и изменить его, а затем рассчитать новый дайджест, присоединить новый дайджест к измененному сообщению и отправить Скотту. Когда Скотт получит его, он проверит дайджест сообщения, но так и не узнает, что в действительности сообщение было изменено Брюсом. Скотт будет думать, что сообщение пришло от Черила в неизменном виде, т.к. он сравнил два значения дайджеста (рассчитанное самостоятельно и прикрепленное к сообщению) и они были равны. Если Черилу нужен более высокий уровень защиты, ему нужно использовать **код аутентификации сообщения** (MAC – Message Authentication Code).

Функция MAC – это схема аутентификации, полученная в результате применения секретного ключа к сообщению. Но это не означает, что используется симметричный ключ для шифрования сообщения.

Вам нужно знать два основных типа MAC: MAC-хэш (HMAC – hash-MAC) и CBC-MAC.

НМАС

Выше мы рассматривали пример, когда Черил передавал сообщение Скотту. Посмотрим, что изменится, если Черил воспользуется функцией HMAC вместо простого алгоритма хэширования. Перед передачей в алгоритм хэширования к его сообщению прикрепляется симметричный ключ. Затем к исходному сообщению, уже без симметричного ключа, прикрепляется полученное значение MAC и результат отправляется Скотту. Если Брюс перехватит это сообщение и изменит его, он не сможет рассчитать правильное новое значение MAC, поскольку у него нет необходимого симметричного ключа. На Рисунке 6-20 показаны эти шаги.



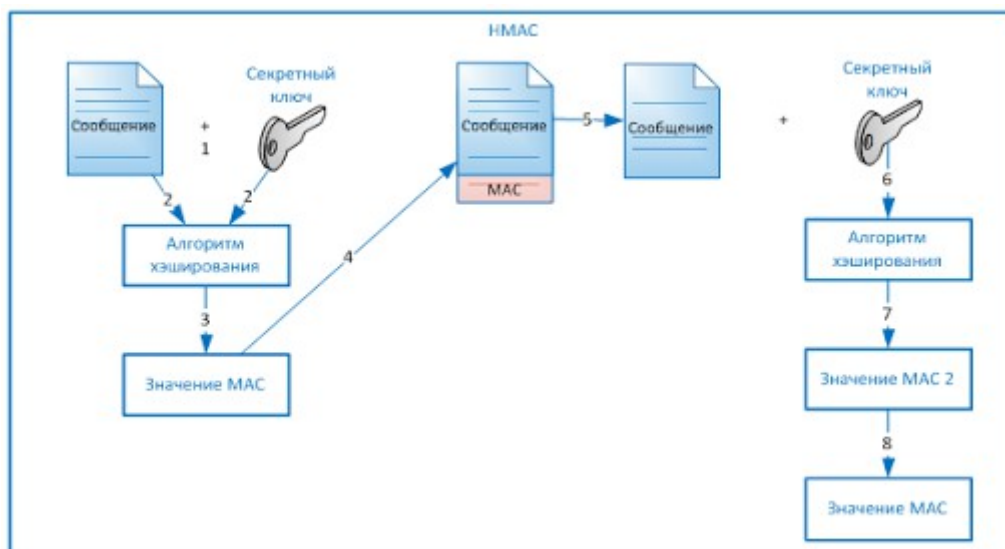


Рисунок 6-20. Шаги выполнения алгоритма хэширования и функции HMAC

Используемая терминология. Идея функций хэширования очень проста. Вы пропускаете сообщение через хэширующий алгоритм, который вырабатывает хэш-значение. Здесь используется совсем немного терминов.

- Хэш-значение может также называться *дайджестом* (digest) или *отпечатком* (fingerprint) сообщения.
- Алгоритмы хэширования также называют *бесключевыми дайджестами сообщений* (nonkeyed message digest).
- Существует два типа MAC: HMAC (hashed MAC) и CBC-MAC.
- MAC также иногда называют *кодом целостности сообщения* (MIC – Message Integrity Code) или *кодом выявления изменений* (MDC – Modification Detection Code).

Ниже перечислены основные шаги процесса хэширования, показанного на Рисунке 6-20:

1. Отправитель пропускает сообщение через функцию хэширования.
2. Генерируется значение дайджеста сообщения.
3. Дайджест сообщения добавляется к сообщению.
4. Отправитель отправляет сообщение получателю.
5. Получатель пропускает сообщение через функцию хэширования.
6. Получатель генерирует свое собственное значение дайджеста сообщения.
7. Получатель сверяет два значения дайджеста сообщения. Если они совпадают, сообщение не было изменено.

В нижней части Рисунка 6-20 показаны шаги HMAC:

1. Отправитель добавляет симметричный ключ к сообщению.
2. Результат помещается в алгоритм хэширования.
3. Генерируется значение MAC.
4. Значение MAC добавляется к сообщению.
5. Отправитель отправляет сообщение получателю (только сообщение с присоединенным к нему значением MAC, симметричный ключ вместе с сообщением не передается).
6. Получатель добавляет свою копию симметричного ключа к полученному сообщению.
7. Получатель пропускает результат через хэширующий алгоритм и генерирует свое

собственное значение MAC.

8. Получатель сравнивает два значения MAC. Если они совпадают, сообщение не было изменено.

Когда мы говорим здесь, что к сообщению добавляется симметричный ключ, мы не подразумеваем, что этот ключ применяется для шифрования сообщения. Функция HMAC не шифрует сообщение, поэтому она не обеспечивает конфиденциальность.

Эта технология требует, чтобы у отправителя и получателя был одинаковый симметричный ключ. Функция HMAC не реализует безопасную передачу симметричного ключа получателю. Для этого применяются другие технологии, которые мы уже обсуждали ранее (алгоритм Диффи-Хеллмана и соглашение о ключах, либо RSA и обмен ключами).

CBC-MAC

При использовании CBC-MAC, сообщение шифруется симметричным блочным шифром в режиме CBC, а последний блок шифротекста используется в качестве MAC. При этом отправитель не отправляет зашифрованную версию сообщения, он отправляет открытый текст с прикрепленным к нему MAC. Получатель зашифровывает полученный открытый текст таким же симметричным блочным шифром в режиме CBC и независимо рассчитывает свое значение MAC. Затем получатель сравнивает рассчитанное значение MAC со значением в полученном сообщении. Этот метод, в отличие от HMAC, не использует алгоритмов хэширования.

Использование симметричного ключа гарантирует, что целостность сообщения может проверить только человек, обладающий копией этого ключа. Больше проверить это не может никто, и если кто-то перехватит и изменит данные, он не сможет сгенерировать новое значение MAC (HMAC или CBC-MAC) так, чтобы получатель не смог заметить подмены. Любые изменения будут заметны получателю.

Таким образом, получатель знает, что полученное сообщение пришло от отправителя, который имеет другую копию того же симметричного ключа, что обеспечивает возможность для **аутентификации источника данных** (data origin authentication), иногда называемой **аутентификацией системы** (system authentication). Это отличается от аутентификации пользователя, которая требует использования закрытого ключа. Закрытый ключ связан с конкретным человеком, а симметричный ключ – нет. Таким образом, MAC-аутентификация обеспечивает слабый вид аутентификации, т.к. аутентифицируется не сам пользователь, а только компьютер или устройство.

ПРИМЕЧАНИЕ. Один и тот же ключ не следует использовать и для аутентификации, и для шифрования.

Как и в большинстве других алгоритмов, в CBC-MAC были найдены некоторые проблемы безопасности, для решения которых был создан **CMAC** (Cipher-Based Message Authentication Code - Код аутентификации сообщения, основанный на шифровании). CMAC обеспечивает такой же вариант аутентификации источника данных и контроля целостности, как и CBC-MAC, но он более защищен с математической точки зрения. CMAC является одной из разновидностей CBC-MAC, он был одобрен для работы с алгоритмами AES и 3DES. Для выявления изменения данных используется CRC, но контроль целостности выполняется обычно на более низком уровне сетевого стека. Поскольку эти функции работают на более низком уровне сетевого стека, они используются для выявления изменений (повреждений) при передаче сетевых пакетов от одного компьютера другому. HMAC, CBC-MAC и CMAC работают на более высоких уровнях сетевого стека, поэтому могут выявлять не только ошибки передачи (случайные), но и намеренные изменения сообщений злоумышленником для получения собственной выгоды. Таким образом, все эти технологии (за исключением CRC) могут выявить намеренные, несанкционированные изменения, а также случайные, неумышленные изменения.

CMAC работает следующим образом. Симметричный алгоритм (AES или 3DES) создает симметричный ключ. Этот ключ используется для создания суб-ключей. Суб-ключи используются по-отдельности для зашифрования отдельных блоков сообщения, как показано на Рисунке 6-21. Это в точности совпадает с тем, как работает CBC-MAC, но с более сильной магией. Этой магией является сложная математика, однако столь глубоких знаний не требуется для прохождения экзамена CISSP. Чтобы лучше понять эту математическую магию, ознакомьтесь с документом по адресу: http://csrc.nist.gov/publications/nistpubs/800-38B/SP_800-38B.pdf.

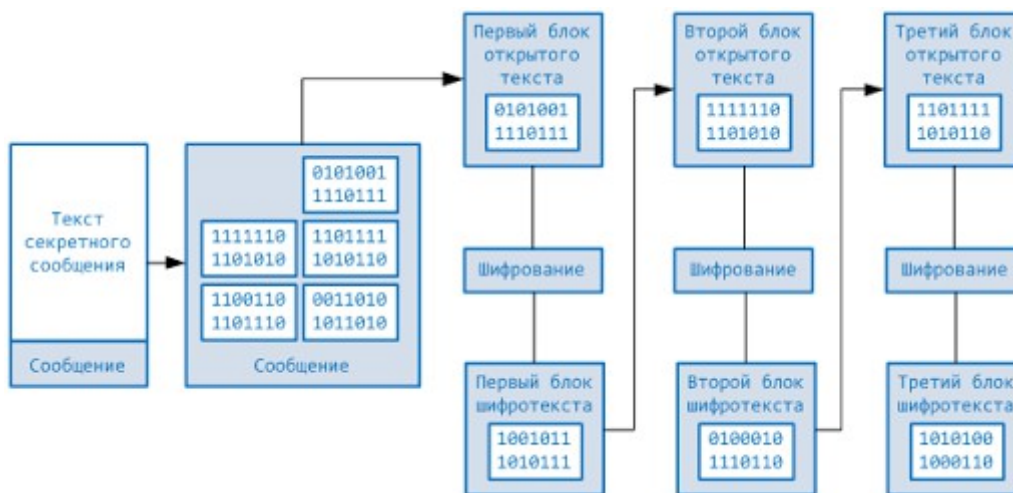


Рисунок 6-21. Процесс шифрования блочным шифром в режиме сцепления блоков шифротекста

Хотя углубление в математику CMAC не обязательно для экзамена CISSP, вам нужно знать, что это алгоритм расчета кода аутентификации сообщения, основанный на блочном шифре. Это означает, что он также может обеспечить только аутентификацию источника данных (например, направивший их компьютер), но не человека (реально отправившего их).

Хэши, HMAC, CBC-MAC, CMAC. MAC и процессы хэширования могут показаться сложными. Следующая таблица упрощает понимание различий между ними.

Функция	Шаги	Предоставляемые сервисы безопасности
Хэш	<ol style="list-style-type: none"> 1. Отправитель пропускает сообщение через хэширующий алгоритм и генерирует значение дайджеста сообщения (MD – Message Digest). 2. Отправитель отправляет сообщение и значение MD получателю. 3. Получатель пропускает сообщение через такой же хэширующий алгоритм и независимо создает свое собственное значение MD. 4. Получатель сравнивает значения MD. Если они совпадают, сообщение не было изменено. 	Целостность, но не конфиденциальность или аутентификация. Может выявлять только случайные, неумышленные изменения.
HMAC	<ol style="list-style-type: none"> 1. Отправитель добавляет к сообщению симметричный ключ и пропускает результат через хэширующий алгоритм. В результате создается значение MAC. 2. Отправитель добавляет значение MAC к исходному сообщению и отправляет результат получателю. 3. Получатель добавляет к сообщению свою собственную копию того же симметричного ключа. В результате получает свое независимое значение MAC. 4. Если они совпадают, получатель знает, что сообщение не было изменено и знает, от какой системы оно пришло. 	Целостность и аутентификация источника данных. Конфиденциальность не обеспечивается.
CBC-MAC	<ol style="list-style-type: none"> 1. Отправитель шифрует сообщение симметричным блочным алгоритмом в режиме CBC. 2. Последний блок шифротекста используется в качестве MAC. 3. К открытому тексту сообщения добавляется MAC и результат отправляется получателю. 4. Получатель аналогичным образом шифрует сообщение, создает собственное значение MAC и сравнивает эти значения. Если они совпадают, получатель знает, что сообщение не было изменено, и знает с какой системы оно пришло. 	Аутентификация источника данных и целостность.
CMAC	CMAC работает аналогично CBC-MAC, но он основан на более сложных математических и логических функциях.	

7.2. Различные алгоритмы хэширования

Как было сказано ранее, целью использования односторонних функций хэширования является создание "отпечатка" (хэша) сообщения. Если два различных сообщения дают в результате одинаковое значение хэша, это может упростить атакующему задачу взлома механизма безопасности, поскольку такой шаблон может быть воспроизведен.

Стойкие односторонние функции хэширования не должны давать одинаковое значение хэша для двух и более разных сообщений. Если хэширующий алгоритм дает гарантии создания различных хэшей для двух и более различных сообщений, говорят, что он *не подвержен коллизиям* (collision free).

Хорошие криптографические функции хэширования должны иметь следующие характеристики:

- Хэш должен вычисляться на основе всего сообщения.
- Хэширование должно быть односторонним, чтобы сообщение нельзя было восстановить по значению хэша
- Не должно существовать двух различных сообщений, при хэшировании которых получаются одинаковые значения хэша.
- Функция должна быть устойчива к «атаке дня рождения» (birthday attack) (эта атака описана в разделе "Атаки на односторонние хэширующие функции").

В Таблице 6-2 и следующих разделах вкратце описаны некоторые из доступных алгоритмов хэширования, используемые в криптографии в настоящее время.

Алгоритм	Описание
MD2	Односторонняя функция. Создает 128-битные значения хэша. Гораздо медленнее MD4 и MD5.
MD4	Односторонняя функция. Создает 128-битные значения хэша.
MD5	Односторонняя функция. Создает 128-битные значения хэша. Более сложная, чем MD4.
HAVAL	Односторонняя функция. Создает значения хэша различной длины. Это модификация алгоритма MD5, обеспечивающая более высокий уровень защиты от атак, которым подвержен MD5.
SHA	Односторонняя функция. Создает 160-битные значения хэша. Используется в DSA.
SHA-1, SHA-256, SHA-384, SHA-512	Обновленная версия SHA. SHA-1 создает 160-битные значения хэша, SHA-256 – 256-битные и т.д.

Таблица 6-2. Различные алгоритмы хэширования

MD2

MD2 – это односторонняя функция хэширования, разработанная Роном Ривестом, она создает 128-битное значение дайджеста сообщения. Она не обязательно слабее всех остальных алгоритмов семейства MD, но она очень медленная.

MD4

MD4 – это односторонняя функция хэширования, разработанная Роном Ривестом. Она также создает 128-битные значения дайджеста сообщения. Она используется для высокоскоростных вычислений в программных реализациях и оптимизирована для микропроцессоров.

MD5

MD5 также создана Роном Ривестом и является новой версией MD4. Она также создает 128-битные хэши, но ее алгоритм более сложен и более устойчив к взлому.

MD5 добавляет четвертый цикл операций, выполняемых в процессе работы функции хэширования, он увеличивает количество выполняемых математических операций и повышает сложность, обеспечивая более высокий уровень безопасности.

SHA

Когда американскому правительству потребовался более безопасный алгоритм хэширования Агентством Национальной Безопасности был разработан алгоритм **SHA** для использования в стандарте DSS (Data Signature Standard). SHA предназначен для формирования цифровых подписей.

SHA создает 160-битное значение хэша (или дайджеста сообщения). Затем это значение хэша передается в асимметричный алгоритм, который рассчитывает значение подписи для сообщения.

SHA похож на MD4. Он использует некоторые дополнительные математические функции и создает 160-битные значения хэша (а не 128-битные). Он более устойчив к брутфорс-атакам, включая атаку «дня рождения».

SHA был усовершенствован и переименован в SHA-1. Недавно были разработаны и выпущены новые версии этого алгоритма (вместе называемые семейством алгоритмов SHA-2): SHA-256, SHA-384 и SHA-512.

HAVAL

HAVAL – это односторонняя хэширующая функция, создающая хэши, длиной от 128 до 256 бит. HAVAL является модификацией MD5. Она обрабатывает сообщение, разбивая его на блоки по 1024 бита (вдвое больше длины блоков в MD5).

Tiger

Росс Андерсон и Эли Бихэм разработали в 1995 году алгоритм хэширования, названный Tiger. Он был разработан для выполнения функций хэширования в 64-битных системах,

скорость его работы превышает скорость MD5 и SHA-1. Он создает хэш-значения длиной 192 бита. Большинство алгоритмов хэширования (например, MD5, RIPEMD, SHA0, SHA1) построены на основе архитектуры MD4. Tiger основан на другой архитектуре, чтобы он не был уязвим для тех же видов атак, которые успешно применялись против других алгоритмов хэширования. Чтобы лучше понять, как проводились атаки в процессе тестирования алгоритма Tiger, ознакомьтесь с этим документом http://th.informatik.uni-mannheim.de/people/lucks/papers/Tiger_FSE_v10.pdf.

ПРИМЕЧАНИЕ. В рамках европейского проекта, названного RIPE (RACE Integrity Primitives Evaluation), был разработан алгоритм хэширования на замену MD4. Этот алгоритм был назван RIPEMD. Он очень похож на MD4, но он не привлек такого же внимания.

Ссылки по теме:

- Counterpane Internet Security, Inc. home page
- The Search Directory: Cryptography, 3DES, Hash algorithm, MD5, SHA-1 links
- RFC 1321 – The MD5 Message-Digest Algorithm
- “Asymmetric Cryptography,” sample chapter from .NET Security and Cryptography, by Peter Thorsteinson and G. Ganesh (Prentice Hall PTR, August 2003)

7.3. Атаки на односторонние функции хэширования

Хороший алгоритм хэширования не должен создавать одинаковое значение хэша для двух различных сообщений. Одинаковое значение хэша для двух (или более) различных сообщений называется **коллизией** (collision). Атакующий может попытаться создать коллизию специально, что называется **атакой «дня рождения»** (birthday attack). Эта атака основана на математическом парадоксе «день рождения», существующем в обычной статистике. Например, сколько людей нужно собрать в одной комнате, чтобы среди них нашелся человек, родившийся в тот же день, что и вы? Ответ: 253. Сколько людей нужно собрать в одной комнате, чтобы среди них нашлись два или более человека, родившийся в один день? Ответ: 23.

Обратите внимание, сколько нужно людей, чтобы у одного из них день рождения совпал с вашим. А теперь посмотрите, сколько людей нужно для того, чтобы день рождения совпал у кого-то среди них. Вероятность найти двух людей, родившихся в один день, выше, чем вероятность найти человека, родившегося в тот же день, что и вы. Другими словами, проще найти два совпадающих значения в море значений, чем найти в нем значение, равное определенному значению.

Но зачем это нам? Парадокс «день рождения» может применяться в криптографии. Для любой случайной группы из 23 человек высока вероятность (не менее 50%), что в ней найдутся два человека с одинаковым днем рождения. Переводя это в криптографию, можно сказать, что если алгоритм хэширования создает 60-битные значения хэшей, существует высокая вероятность коллизии при сравнении хэшей всего 2^{30} сообщений.

Основная задача атакующего – попытаться с помощью брутфорс-атаки найти сообщения, значения хэшей которых совпадают со значением хэша определенного сообщения. Если это ему удастся, это аналогично нахождению человека с таким же днем рождения. Если он находит два сообщения с одним и тем же значением хэша, это аналогично нахождению двух людей с одинаковым днем рождения.

Если длина результирующего значения на выходе из алгоритма хэширования является n , то чтобы с помощью брутфорс-атаки найти сообщение с определенным значением хэша, потребуется рассчитать хэши для 2^n случайных сообщений. А чтобы просто найти два сообщения с одинаковым значением хэша, потребуется рассчитать хэши для $2^{n/2}$ случайных сообщений.

Как может произойти атака дня рождения в криптографии?

Предположим, что Сью и Джо собираются пожениться, но перед этим они заключают брачный контракт, в котором указано, что в случае развода Сью получит свою первоначальную собственность, а Джо – свою. Чтобы гарантировать, что контракт не изменен, используется функция хэширования и создается дайджест сообщения.

Через месяц после свадьбы Сью делает копию значения дайджеста сообщения и пишет новый брачный контракт, в котором указывает, что в случае развода она получает не только свою собственность, но и собственность Джо. Сью создает хэш нового контракта и сравнивает его с хэшем первоначального варианта. Они не совпадают. Тогда Сью немного изменяет новый контракт, рассчитывает новое значение хэша и снова сравнивает его с первоначальным. Она продолжает вносить незначительные (почти незаметные) изменения в новый контракт, пока не находит коллизию, в результате которой выполнение той же функции хэширования над ее новым вариантом контракта дает такое же значение дайджеста сообщения, что и первоначальный контракт. Затем Сью заменяет первоначальный контракт на новый и по-быстрому разводится с Джо. Забирая собственность Джо, она показывает ему новый контракт, и доказывает его неизменность, сравнивая значения хэша.

Алгоритм хэширования обычно создает значения хэша достаточно большого размера (значение n), затрудняя нахождение коллизий, но они остаются реальными. Например, алгоритм, создающий 160-битные хэши (например, SHA-1), требует перебрать порядка 2^{80} вариантов сообщения для нахождения коллизии. Таким образом, существует менее одного шанса на 2^{80} , что кому-то удастся выполнить успешную атаку «дня рождения».

Парадокс «день рождения» показывает важность использования длинных значений хэша. Чем более длинный результат выдает алгоритм хэширования, тем меньше он уязвим к брутфорс-атаке, такой как атака «дня рождения». Именно по этой причине новая версия SHA выдает такие длинные значения дайджеста сообщения.

Ссылки по теме:

- The PGP Attack FAQ, Part 3, “The One-Way Hash”

7.4. Цифровая подпись

Цифровая подпись – это значение хэша, зашифрованное на закрытом ключе отправителя. Процесс подписи значения хэша сообщения закрытым ключом показан на Рисунке 6-22.

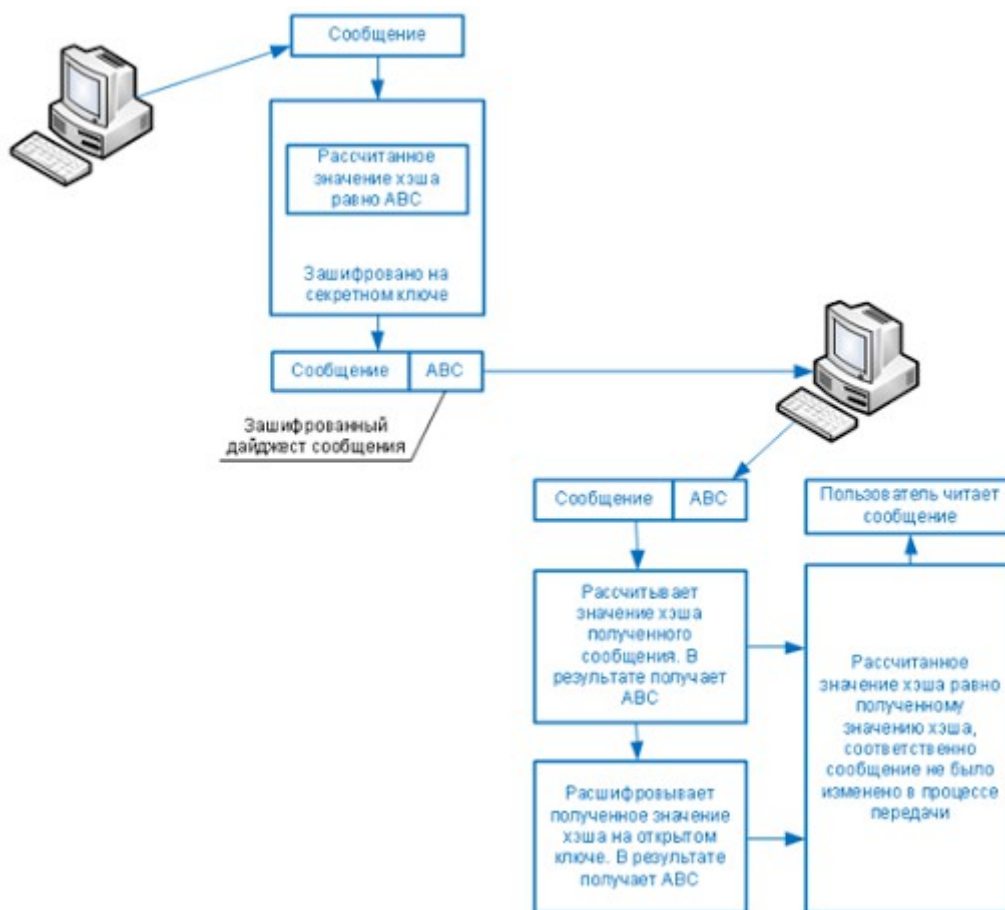


Рисунок 6-22. Создание цифровой подписи сообщения

Вспомним наш пример из раздела «Односторонние хэши». Если Кевину нужно гарантировать отсутствие изменений в сообщении, которое он отправляет Марии, *и* обеспечить, чтобы она была уверена в том, что сообщение исходит именно от Кевина, он может подписать сообщение с помощью цифровой подписи. Для этого сообщение сначала обрабатывается с помощью односторонней функции хэширования, а затем Кевин зашифровывает полученное значение хэша на своем закрытом ключе.

Мария, получив сообщение, выполнит функцию хэширования над сообщением и получит свое собственное значение хэша. Затем она расшифрует полученное в сообщении значение хэша (цифровую подпись) на открытом ключе Кевина. Теперь она может сравнить два значения, и если они совпадут, она будет уверена, что сообщение не было изменено в процессе передачи. Она также будет уверена, что сообщение исходит именно от Кевина, поскольку значение хэша было зашифровано на его закрытом ключе.

Функция хэширования гарантирует целостность сообщения, а подписание значения хэша обеспечивает аутентификацию и неотказуемость. Подписание сообщения цифровой подписью в действительности просто означает, что значение хэша сообщения зашифровывается на закрытом ключе.

Вам должны быть понятны все доступные механизмы криптографии, т.к. различные алгоритмы и отдельные шаги предоставляют различные сервисы безопасности:

- Сообщение может быть зашифровано, что обеспечивает конфиденциальность.
- Для сообщения может быть рассчитано значение хэша, что обеспечивает целостность.
- Сообщение может быть подписано цифровой подписью, что обеспечивает аутентификацию, неотказуемость и целостность.

- Сообщение может быть зашифровано и подписано цифровой подписью, что обеспечит конфиденциальность, аутентификацию, неотказуемость и целостность.

Некоторые алгоритмы могут выполнять только шифрование, тогда как другие поддерживают и цифровую подпись, и шифрование. Для хэширования используется алгоритм хэширования, а не алгоритм шифрования.

Важно понимать, что не все алгоритмы реализуют одновременно все сервисы безопасности. Большинство алгоритмов используются в комбинации с другими, для обеспечения необходимого набора сервисов безопасности. В Таблице 6-3 показаны сервисы, обеспечиваемые различными алгоритмами.

Алгоритм	Шифрование	Цифровая подпись	Хэширование	Распространение ключей
Алгоритмы с асимметричными ключами				
RSA	X	X		X
ECC	X	X		X
Диффи-Хеллман				X
Эль Гамаль	X	X		X
DSA		X		
LUC	X	X		X
Knapsack	X	X		X
Алгоритмы с симметричными ключами				
DES	X			
3DES	X			
Blowfish	X			
IDEA	X			
RC4	X			
SAFER	X			
Алгоритмы хэширования				
MD2, MD4 и MD5			X	
SHA			X	
HAVAL			X	

Таблица 6-3. Функциональность различных алгоритмов



7.5. Стандарт цифровой подписи

Поскольку применение цифровой подписи является крайне важным компонентом при определении того, кто отправил сообщение, правительство США решило издать стандарт, относящийся к функциональности и приемлемому использованию цифровой подписи. В 1991 году NIST предложил федеральный стандарт, названный **Стандартом цифровой подписи** (DSS – Digital Signature Standard). Он был разработан для федеральных управлений и агентств, но большинство производителей также воспользовались этим стандартом при разработке своей продукции. Федеральное правительство требует, чтобы его департаменты

использовали DSA, RSA или ECDSA (Elliptic Curve Digital Signature Algorithm). Реализовано это следующим образом: SHA создает дайджесты сообщений длиной 160 бит, которые затем передаются в один из трех вышеуказанных алгоритмов для формирования цифровой подписи. При этом SHA используется для обеспечения целостности сообщений. Это пример совместного использования двух различных алгоритмов для получения нужного сочетания сервисов безопасности.

RSA и DSA являются хорошо известными и наиболее широко используемыми алгоритмами цифровой подписи. DSA был разработан Агентством национальной безопасности США. В отличие от RSA, DSA может применяться только для формирования цифровой подписи, кроме того, DSA медленнее RSA при проверке подписи. А RSA может применяться не только для цифровой подписи, но и для шифрования и безопасного распространения симметричных ключей.

8. Инфраструктура открытых ключей

PKI (Public key infrastructure – Инфраструктура открытых ключей) состоит из программ, форматов данных, процедур, коммуникационных протоколов, политик безопасности и криптографических механизмов с открытыми ключами, работающих совместно, с целью предоставления широкого спектра возможностей для безопасных и предсказуемых коммуникаций. Другими словами, PKI устанавливает уровень доверия в пределах окружения. PKI является аутентификационной платформой ISO, использующей криптографию с открытыми ключами и стандарт X.509. Эта платформа позволяет выполнять аутентификацию между различными сетями, а также сетью Интернет. Отдельные протоколы и алгоритмы не имеют спецификации, поэтому PKI называется платформой, а не конкретной технологией.

PKI обеспечивает аутентификацию, конфиденциальность, неотказуемость и целостность при обмене сообщениями. PKI – это *гибридная* система, в которой применяются алгоритмы и методы с симметричными и асимметричными ключами, обсуждавшимися ранее.

Существуют различия между криптографией с открытыми ключами и PKI. Криптография с открытыми ключами – это просто другое название асимметричных алгоритмов, в то время как название PKI говорит о том, что это инфраструктура. Эта инфраструктура отвечает за однозначное подтверждение личности отправителя с помощью сертификатов, а также выполнение автоматического процесса обмена ключами с помощью асимметричного алгоритма. Для этого данная инфраструктура содержит компоненты, выполняющие идентификацию пользователей, создание и распространение сертификатов, поддержку и аннулирование сертификатов, распространение и поддержку ключей шифрования, а также позволяющие всем технологиям, из которых состоит данная инфраструктура, взаимодействовать и совместно работать, обеспечивая возможности для осуществления зашифрованных коммуникаций и аутентификации.

Криптография с открытыми ключами – это одна из частей PKI, но в эту инфраструктуру входит множество других частей. Здесь можно провести аналогию с протоколом SMTP. SMTP – это технология, применяемая для передачи сообщений электронной почты, но для того, чтобы она работала, необходимо множество других вещей: клиентское программное обеспечение для электронной почты, почтовые серверы и сообщения электронной почты, которые в совокупности создают инфраструктуру – инфраструктуру электронной почты. PKI также состоит из множества различных частей: центров сертификации, центров регистрации, сертификатов, ключей и пользователей. В следующих разделах рассмотрены эти отдельные части, а также их совместная работа.

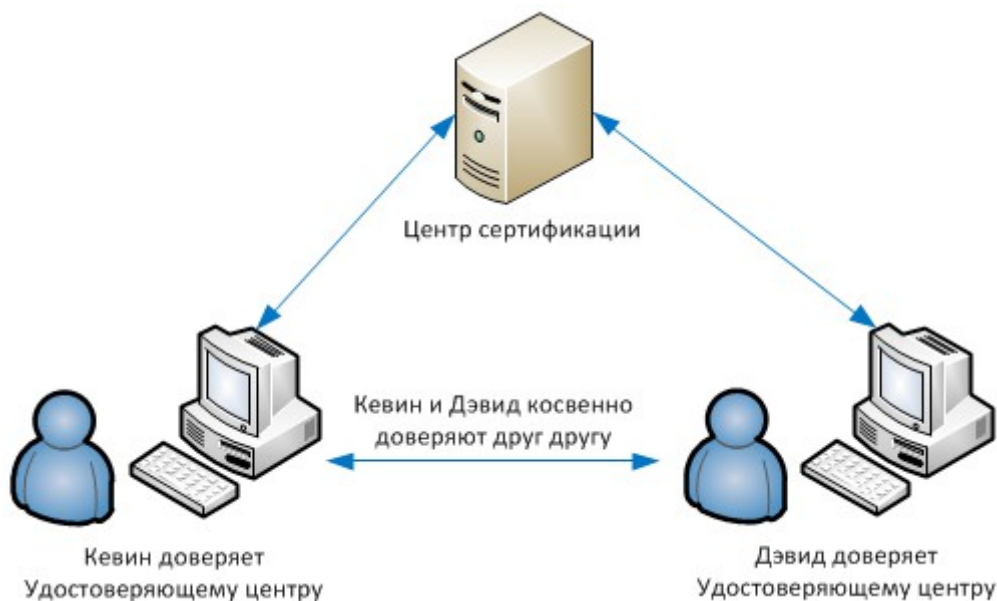
Каждый человек, который хочет участвовать в PKI, должен получить цифровой сертификат, являющийся его удостоверением личности. Цифровой сертификат содержит открытый ключ этого человека, а также другую идентификационную информацию. Сертификат

создается и подписывается (цифровой подписью) доверенной третьей стороны, которой является Удостоверяющий центр (Центр сертификации). При подписании Центром сертификации сертификата, открытый ключ человека связывается с его идентификационными данными, а Удостоверяющий центр берет на себя ответственность за проверку личности этого человека. Эта доверенная третья сторона (Удостоверяющий центр) позволяет людям, которые никогда не встречались друг с другом, проводить взаимную аутентификацию и взаимодействовать с помощью безопасных механизмов. К примеру, если Кевин никогда не встречался с Дэвидом, но ему нужно взаимодействовать с ним безопасным образом, и при этом они оба доверяют одному и тому же Удостоверяющему центру, Кевин может получить в нем цифровой сертификат Дэвида и они могут начать процесс безопасного взаимодействия.

8.1. Центр сертификации

Центр сертификации (CA – Certificate Authority, Удостоверяющий центр) – это доверенная организация (или сервер), которая выпускает и осуществляет поддержку цифровых сертификатов. Если человек запрашивает сертификат, Центр регистрации проверяет личность этого человека, после чего передает запрос на сертификат в Центр сертификации. Центр сертификации выпускает сертификат, подписывает его, отправляет запросившему его человеку и в дальнейшем поддерживает этот сертификат на протяжении всего времени его жизни. Если с этим человеком нужно взаимодействовать другому человеку, Центр сертификации обеспечивает подтверждение его личности. После того, как Дэвид получит цифровой сертификат от Кевина, Дэвид выполнит определенные шаги, чтобы проверить его. Предоставляя свой цифровой сертификат Дэвиду, Кэвин говорит что-то вроде: «Я знаю, что ты не знаешь меня и не доверяешь мне, но вот документ, созданный тем, кого ты знаешь и кому ты доверяешь. Документ говорит о том, что я хороший парень, и ты можешь доверять мне».

Проверив цифровой сертификат, Дэвид извлекает из него открытый ключ Кевина. При этом Дэвид уверен, что этот открытый ключ принадлежит Кевину. Также Дэвид знает, что если он получит сообщение, цифровую подпись которого он сможет расшифровать на открытом ключе Кевина, он может быть уверен, что это сообщение пришло от Кевина, поскольку цифровая подпись была зашифрована на его (Кевина) закрытом ключе.

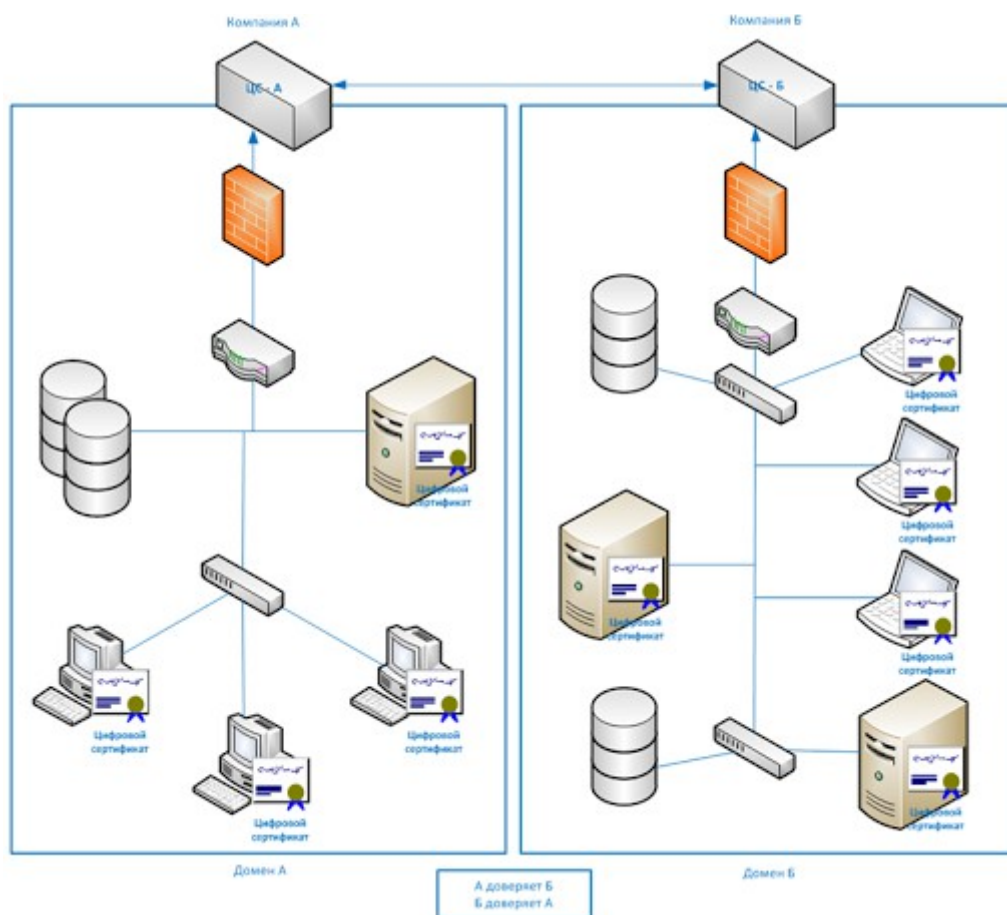


ПРИМЕЧАНИЕ. Помните атаку «человек посередине», которую мы рассматривали ранее в разделе «Алгоритм Диффи-Хеллмана»? Эта атака невозможна в среде PKI, поскольку пользователи в ней достоверно знают личность владельцев открытых ключей.

Удостоверяющий центр может быть внутренним подразделением компании. Это позволит

компания самостоятельно управляет сервером Центра сертификации, настраивать на нем процесс аутентификации, осуществлять поддержку сертификатов, выполнять отзыв отдельных сертификатов в случае необходимости. Другие Удостоверяющие центры являются специализированными организациями, предоставляющие соответствующие услуги. Их клиенты – различные люди и компании – платят им за поддержку. Одними из самых известных Удостоверяющих центров являются Entrust и VeriSign. В стандартных настройках многих браузеров производителем указывается несколько хорошо известных Удостоверяющих центров.

ПРИМЕЧАНИЕ. Все больше и больше компаний организуют свои собственные внутренние PKI. Если такие независимые Удостоверяющие центры должны обеспечивать безопасные коммуникации между различными компаниями, необходим способ обеспечения взаимного доверия корневых Центров сертификации. Два Центра сертификации, не имеющие других вышестоящих Центров сертификации, могут доверять друг другу, если они провели кросс-сертификацию. *Кросс-сертификация* (cross certification) – это процесс, выполняемый Центрами сертификации для установления доверительных отношений, в результате которого они считают цифровые сертификаты и открытые ключи друг друга доверенными и доверяют им также, как своим собственным. Когда это сделано, Удостоверяющий центр одной компании может проверять цифровые сертификаты другой компании и наоборот.



В обязанности Центра сертификации входит создание, выдача сертификатов, их поддержка и, при необходимости, отзыв. Отзыв сертификатов производится Центром сертификации, информация об отозванных сертификатах сохраняется в *списке отозванных сертификатов* (CRL – Certificate revocation list). В этом списке указываются все сертификаты, которые были отозваны на момент формирования списка. Список отозванных сертификатов поддерживается и регулярно обновляется. Сертификат может быть отозван по причине компрометации закрытого ключа владельца или если Удостоверяющему центру стало известно, что сертификат был выпущен не тому человеку. Если сертификат стал по каким-либо причинам недействительным, именно список отозванных сертификатов является тем механизмом Удостоверяющего центра, который позволяет другим узнать об этом.

ПРИМЕЧАНИЕ. Во многих реализациях PKI работа списка отозванных сертификатов имеет недостатки. Интересно отметить тот факт, что по умолчанию веб-браузеры не проверяют список отозванных сертификатов, чтобы убедиться, что сертификат не был отозван. Таким образом, когда вы устанавливаете SSL-соединение при использовании услуги электронной коммерции через Интернет, вы можете довериться сертификату, который в действительности был отозван. Это не очень хорошо.

Протокол OSCP (Online Certificate Status Protocol) используется все чаще по сравнению с громоздким подходом CRL. При использовании CRL, браузер пользователя должен обратиться к центральному CRL, чтобы проверить, не был ли отозван сертификат, либо браузер должен постоянно обновлять CRL на стороне клиента, чтобы поддерживать его актуальность. При использовании OSCP, эта работа выполняется автоматически в фоновом режиме. OSCP выполняет проверку сертификатов в режиме реального времени и сообщает пользователю, является ли сертификат действительным, недействительным или неизвестным. OSCP проверяет CRL, поддерживаемый Центром сертификации. Фактически CRL продолжает использоваться, но OSCP существенно упрощает проверку CRL в процессе проверки сертификата.

8.2. Сертификаты

Одним из наиболее важных элементов PKI является цифровой сертификат. **Сертификат** является механизмом, который используется для связи открытого ключа с набором реквизитов способом, достаточным для уникальной идентификации заявленного владельца. Центр сертификации создает сертификаты на основе стандарта X.509, определяющего, какие поля должны быть предусмотрены в сертификате, а также правильные значения, которые могут быть указаны в этих полях. Сейчас используется четвертая версия этого стандарта, которую иногда записывают как X.509v4. Многие криптографические протоколы (включая SSL) используют такие сертификаты.

В сертификате указывается его серийный номер, номер версии стандарта, идентификационная информация, информация об алгоритме, даты срока действия и подпись выпустившего его Удостоверяющего центра, как показано на Рисунке 6-23.



Рисунок 6-23. Каждый сертификат содержит структуру со всей необходимой идентификационной информацией

8.3. Центр регистрации

Центр регистрации (RA – Registration Authority) выполняет задачи регистрации сертификатов. Центр регистрации устанавливает и подтверждает личность человека (будущего владельца сертификата), инициирует процесс сертификации в Центре сертификации от лица конечного пользователя и выполняет функции управления жизненным циклом сертификатов. Центр регистрации не может выпускать сертификаты, он выполняет функции посредника между пользователем и Центром сертификации. Если пользователю нужны новые сертификаты, он делает запрос в Центр регистрации, Центр регистрации проверяет всю необходимую идентификационную информацию, после чего передает запрос в Центр сертификации.

8.4. Шаги PKI

Теперь мы знаем некоторые основные компоненты PKI и то, как они совместно работают. Давайте еще раз рассмотрим это на примере. Предположим, что Джону нужно получить цифровой сертификат, чтобы принять участие в PKI. Для этого он должен выполнить следующие шаги:

1. Джон делает запрос в Центр регистрации.
2. Запрос в Центр регистрации содержит идентификационную информацию Джона, такую как, копия его водительских прав, номер его телефона, адрес и другую информацию.
3. После получения всей необходимой информации от Джона, Центр регистрации проверяет ее и пересылает его запрос на сертификат в Центр сертификации.
4. Центр сертификации создает сертификат открытого ключа Джона и включает в него идентификационную информацию Джона. (Пара ключей (закрытый/открытый) может генерироваться Центром сертификации или компьютером Джона, в зависимости от настроек системы. Если ключи создаются Центром сертификации, необходимо организовать безопасную доставку пользователю закрытого ключа. В большинстве случаев, пользователь самостоятельно генерирует пару ключей и отправляет свой открытый ключ процессе регистрации).

Теперь Джон зарегистрирован и стал участником PKI. Джону нужно взаимодействовать с Дианой. Для этого им нужно выполнить последовательность шагов, показанную на Рисунке 6-24.



Рисунок 6-24. Взаимодействие пользователя с Центром сертификации

1. Джон запрашивает открытый ключ Дианы из общего каталога.
2. Каталог (часто его называют репозиторием) отправляет Джону цифровой сертификат Дианы.
3. Джон проверяет этот сертификат и извлекает из него открытый ключ Дианы. Джон использует этот открытый ключ для зашифрования сеансового ключа, который будет использоваться для шифрования их сообщений. Джон отправляет Диане зашифрованный сеансовый ключ и свой сертификат, содержащий его открытый ключ.
4. При получении Дианой сертификата Джона, ее браузер проверяет, является ли доверенным Центр сертификации, подписавший этот сертификат своей цифровой подписью. Браузер Дианы доверяет этому Центру сертификации, поэтому после проверки сертификата Джон и Диана могут взаимодействовать с использованием

шифрования.

PKI может состоять из следующих компонентов и функций:

- Центр сертификации
- Центр регистрации
- Репозиторий сертификатов
- Система отзыва сертификатов
- Система резервного копирования и восстановления ключей
- Автоматическое обновление ключей
- Управление историей ключей
- Установка меток времени
- Клиентское программное обеспечение

PKI обеспечивает следующие сервисы безопасности:

- Конфиденциальность
- Управление доступом
- Целостность
- Аутентификация
- Неотказуемость

PKI должен хранить историю ключей, чтобы отслеживать использование людьми как старых, так и актуальных открытых ключей. Например, если Кевин зашифровал симметричный ключ на старом открытом ключе Дэвида, у Дэвида должен существовать способ получения доступа к этим данным. Это может быть возможно только в том случае, если Центр сертификации хранит историю старых сертификатов и ключей Дэвида.

ПРИМЕЧАНИЕ. Другим важным компонентом, который должен быть интегрирован в PKI, является надежный источник данных о времени, который реализует способ безопасного получения штампов времени. Это играет большую роль, когда необходимо обеспечить неотказуемость.

Ссылки по теме:

- Network World Cryptography page
- Introduction to Public Key Technology and the Federal PKI Infrastructure, by D. Richard Kuhn et al., NIST Special Publication 800-32 (Feb. 26, 2001)
- The Open Group Secure Messaging Toolkit

9. Управление ключами

Криптография может использоваться в качестве механизма безопасности, обеспечивающего конфиденциальность, целостность и аутентификацию, но только если используемые ключи не были скомпрометированы. Злоумышленник может перехватить, изменить, повредить или взломать ключи. Криптография основана на модели доверия. Люди должны доверять тому, что другие участники обеспечивают надлежащую защиту своих ключей, доверять администратору, поддерживающему эти ключи, доверять серверу, на котором хранятся, сопровождаются ключи, с которого осуществляется их распространение.

Многие администраторы знают, что управление ключами – самая большая головная боль при внедрении криптографии. Объем работ по поддержке и сопровождению ключей значительно превышает объем работ по их использованию для шифрования сообщений. Ключи должны

безопасно распространяться уполномоченным пользователям и постоянно обновляться. Они должны надежно защищаться при их передаче и хранении на рабочих станциях и серверах. Ключи должны генерироваться, уничтожаться и восстанавливаться безопасным образом. Управление ключами может быть организовано вручную или с помощью автоматического процесса.

Хранение ключей выполняется и до, и после их передачи. Пользователь, при получении ключа, должен не просто бросить его на свой рабочий стол – он обязан разместить ключ в безопасном месте своей файловой системы для его надежного хранения и исключения возможности бесконтрольного использования. Ключ, алгоритм, использующий этот ключ, настройки и параметры сохраняются в модуле, который также должен быть хорошо защищен. Если атакующий сможет получить доступ к этим компонентам, он сможет выдавать себя за другого пользователя, чтобы получить возможности расшифровывать, читать или перешифровывать предназначенные для этого пользователя сообщения.

Криптографические ключи могут физически храниться в защищенных контейнерах и доставляться специальными курьерами под охраной. Когда такой уровень защиты не требуется, ключи могут передаваться на основной сервер, а затем распространяться администратором, чтобы курьеру не нужно было посещать каждого пользователя в отдельности. В некоторых реализациях мастер-ключ передается на все площадки (например, во все офисы компании), а затем используется для генерации уникальных секретных ключей пользователей, работающих на этих площадках (офисах). В настоящее время в большинстве реализаций распространение ключей выполняется автоматически и является одной из функций специализированных протоколов. Чтобы принять решение о том или ином варианте реализации процесса управления ключами, компания должна оценить трудозатраты, требуемые этим процессом, необходимый уровень безопасности, а также вопросы соотношения затрат и преимуществ. Но в общем случае, автоматизация процесса управления ключами обеспечивает более высокую эффективность и безопасность.

При использовании протокола Kerberos (описанного в Домене 02), для хранения, распространения и сопровождения криптографических сеансов и секретных ключей применяется Центр распространения ключей (KDC). Это пример автоматизированного метода распространения ключей. Компьютер, которому нужен доступ к определенному сервису на другом компьютере, запрашивает этот доступ через KDC. KDC генерирует сеансовый ключ, который используется при взаимодействии запрашивающего компьютера с компьютером, предоставляющим запрашиваемый сервис или ресурс. Автоматизация этого процесса снижает вероятность ошибок, которые не редко случаются при ручной работе, однако если Служба предоставления билетов (TGS, являющаяся частью KDC) скомпрометирована, все компьютеры и их сервисы становятся подвержены возможной компрометации.

В некоторых случаях, управление ключами по-прежнему выполняется вручную. Многие компании используют криптографические ключи, но, к сожалению, они редко меняют их, что может быть вызвано как раз трудностями в процессе управления ключами, либо чрезмерной загруженностью сетевого администратора другими задачами, или его непониманием реальной важности выполнения регулярной смены ключей. Частота, с которой следует менять криптографические ключи, напрямую связана с частотой их использования. Чем чаще используется ключ, тем выше вероятность его перехвата и компрометации. Если ключ используется очень редко, этот риск существенно снижается. Кроме частоты использования ключей, на необходимую частоту их смены также указывает требуемый уровень безопасности. Криптографические ключи, используемые для передачи информации, типа приглашения на ужин в ресторане, могут меняться, например, раз в месяц, тогда как ключи, используемые для защиты военной информации, должны меняться ежедневно или еженедельно. Также, следует обратить внимание на безопасность самих методов смены ключей.

Управление ключами – это наиболее сложная часть криптографии и, в то же время, наиболее критичная. Это нужно учитывать при разработке сложных алгоритмов и методов работы с ключами, но если ключи хранятся или передаются небезопасным образом, стойкость используемого алгоритма не имеет никакого значения.

9.1. Принципы управления ключами

Ключ не должен находиться в виде открытого текста вне криптографического устройства. Как было сказано ранее, многие криптографические алгоритмы общедоступны, что усложняет защиту секретности ключа. Если атакующий знает, как работает алгоритм, в большинстве случаев для компрометации системы ему нужен только ключ. Поэтому ключи не должны быть доступны в открытом виде, ведь именно ключи обеспечивают секретность при шифровании.

Эти шаги, а также весь процесс распространения и сопровождения ключей следует автоматизировать и скрыть от пользователя. Их следует интегрировать в программное обеспечение или операционную систему. Выполнение этого процесса вручную только добавит сложностей и откроет дверь для большего количества ошибок. Кроме того, это приведет к зависимости от конечных пользователей при выполнении ими определенных функций.

Ключи подвержены риску утраты, уничтожения или повреждения. Следует организовать их резервное копирование, резервные копии должны быть легко доступны при необходимости. Если пользователь зашифровал данные, а затем потерял ключ, необходимый для их расшифрования, эта информация может быть навсегда потеряна, если не была создана резервная копия ключа. Криптографическое приложение может иметь специальные функции для восстановления ключей, либо требовать, чтобы пользователь организовал безопасное хранение копии ключей.

Восстановление ключей с резервных копий может потребоваться во множестве различных случаев. Например, в зашифрованной папке на компьютере Боба хранятся все важные расчеты цен, информация по биржевым индексам, а также презентация по анализу корпоративных трендов, завтра с утра необходимая высшему руководству. А Боб, так некстати, упал и сломал руку... Но кто-то же должен получить доступ к этим данным вместо него. Другой пример – из компании увольняется сотрудник, на компьютере которого хранятся важные зашифрованные документы. Вероятно, компании позднее потребуется получить доступ к этим данным. Или вице-президент положил дискету со своим закрытым ключом в ящик стола, вместе с мощным магнитом. Он не намерен выслушивать лекцию об электромагнитных полях и их воздействии на носители информации, ему срочно нужен доступ к важной информации.

Конечно, наличие дополнительных экземпляров ключа увеличивает вероятность его компрометации, поэтому компания должна решить, действительно ли нужны резервные копии, и, если они все-таки необходимы, внедрить соответствующие меры для их надежной защиты. Компания может внедрить систему коллегиального аварийного восстановления ключей, в которой при необходимости восстановления ключа, требуется участие нескольких человек одновременно (2-3 или более). Они должны ввести свои закрытые ключи или аутентифицироваться иным способом для восстановления утраченного ключа. При этом эти люди не должны быть сотрудниками одного подразделения (например, Департамента ИТ). Желательно, чтобы среди них были руководители, представители безопасности и кто-то из Департамента ИТ. Коллегиальное восстановление ключей снижает вероятность злоупотреблений, поскольку для этого необходим сговор двух или более людей.

9.2. Правила использования ключей и управления ключами

Крайне важно обеспечить надлежащую защиту процесса управления ключами. К процессу управления ключами предъявляются следующие требования:

- Длина ключа должна быть достаточно большой для обеспечения необходимого уровня защиты.
- Ключи должны храниться и передаваться безопасными способами.
- Ключи должны быть абсолютно случайными, а алгоритмы их генерации должны полностью использовать все доступное ключевое пространство.
- Срок действия ключа должен соответствовать критичности защищаемых им данных. Менее критичные данные можно защищать ключом с большим сроком действия, тогда как критичные данные требуют использования ключей с коротким сроком действия.
- Чем чаще используется ключ, тем короче должен быть срок его действия.
- Должна быть создана резервная копия ключа, либо дубликат ключа должен быть передан на хранение независимой третьей стороне (escrowed) на случай чрезвычайной ситуации.
- Ключи должны надлежащим образом уничтожаться, когда завершается срок их действия.

10. Канальное и сквозное шифрование

Шифрование может выполняться различными способами и на различных коммуникационных уровнях. Существует два основных режима реализации шифрования – канальное шифрование и сквозное шифрование. При **канальном шифровании** (link encryption) зашифровываются все данные, передаваемые по определенному коммуникационному маршруту, например, по спутниковой линии связи, линии ТЗ, телефонной линии. При этом шифруется не только пользовательская информация, но и заголовки, окончания, адреса, данные маршрутизации – шифрованию подлежит все содержимое передаваемых пакетов. В этой технологии не шифруется только трафик управляющих сообщений канального уровня, который включает в себя команды и параметры, используемые различными канальными устройствами для синхронизации процесса коммуникаций. Канальное шифрование обеспечивает защиту от перехвата пакетов и прослушивания сети. При **сквозном шифровании** (end-to-end encryption) информация в заголовках, адреса, данные маршрутизации и окончания не шифруются, что позволяет атакующим получить значительно больше информации из перехваченных пакетов и их заголовков.

Канальное шифрование, которое иногда называют *шифрованием в режиме реального времени* (online encryption), обычно реализуется провайдерами услуг и является частью сетевых протоколов. Поскольку вся информация зашифрована, пакеты должны расшифровываться на каждом узле (таком, как маршрутизатор) или ином промежуточном устройстве, чтобы этот узел мог принять решение, куда дальше отправлять этот пакет. Маршрутизатор должен расшифровать заголовок пакета, считать из него информацию маршрутизации и адреса, а затем снова зашифровать его и отправить дальше.

При сквозном шифровании, пакет не нужно расшифровывать и снова зашифровывать на каждом узле, т.к. заголовки и окончания пакетов при этом не шифруются. Узлы между отправителем и получателем могут просто считать всю необходимую им информацию маршрутизации и передать пакет дальше.

Сквозное шифрование обычно инициируется компьютером отправителя. Сквозное шифрование предоставляет больше гибкости для пользователей, позволяя им самим решать, нужно ли шифровать то или иное сообщение. Такое шифрование называется сквозным, поскольку сообщение остается в зашифрованном виде на всем своем пути от отправителя до получателя.

Канальное шифрование выполняется на канальном и физическом уровнях, как показано на Рисунке 6-25. Аппаратные устройства шифрования работают на физическом уровне, шифруя все проходящие через них данные. В этом случае атакующему не доступны никакие части данных, поэтому атакующий не может получить важную информацию о потоках данных в сети. Это называют *безопасностью потока трафика* (traffic-flow security).

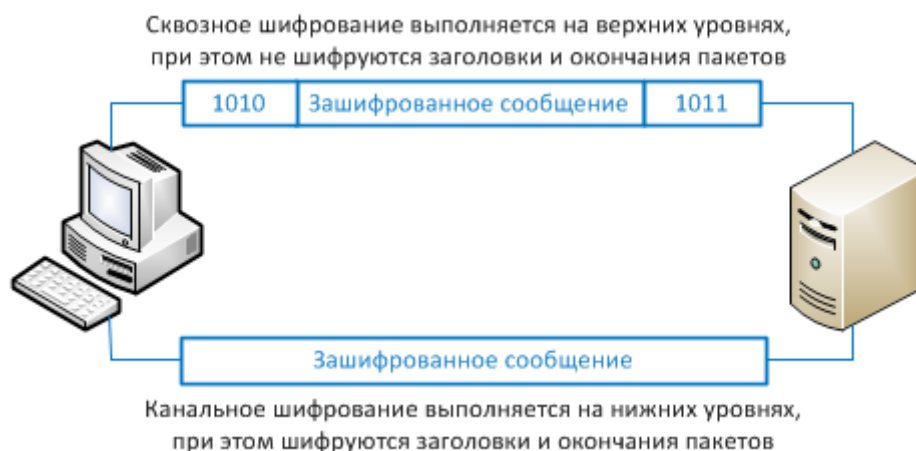


Рисунок 6-25. Канальное и сквозное шифрование выполняются на различных уровнях модели OSI

ПРИМЕЧАНИЕ. Узел (hop) – это устройство, которое помогает пакету достичь пункта своего назначения. Обычно это маршрутизатор, который анализирует адреса пакета, чтобы решить, куда его следует отправить далее. Обычно пакет проходит через множество узлов между компьютерами отправителя и получателя.

Шифрование на различных уровнях. В действительности, шифрование может выполняться на разных уровнях операционной системы и сетевого стека. Ниже приведено несколько примеров:

- Сквозное шифрование выполняется приложением.
- Шифрование SSL выполняется на транспортном уровне.
- Шифрование PPTP выполняется на канальном уровне.
- Канальное шифрование выполняется на канальном и физическом уровнях.

Следующий список описывает преимущества и недостатки методов сквозного и канального шифрования.

Преимущества сквозного шифрования:

- Пользователям предоставляется больше гибкости в выборе того, что шифровать и как.
- Обеспечивается высокая детализация функциональности, т.к. каждое приложение или пользователь могут выбрать специфические настройки.
- Не требуется, чтобы у каждого компьютера в сети был ключ для расшифрования каждого пакета.

Недостатки сквозного шифрования:

- Не зашифровывается информация заголовков, адреса, информация маршрутизации - эта информация не защищена.

Преимущества канального шифрования:

- Все данные зашифрованы, включая заголовки, адреса и информацию маршрутизации.
- Пользователям ничего не нужно делать, чтобы это выполнялось. Шифрование выполняется на нижнем уровне модели OSI.

Недостатки канального шифрования:

- Распространение ключей и управление ими становится сложной задачей, т.к. каждое устройство должно получить ключ. При смене ключа он должен обновляться на каждом устройстве.
- Пакеты расшифровываются на каждом узле, поэтому существует много точек потенциальных уязвимостей.
- Канальное шифрование препятствует выполнению анализа трафика.

Программные и аппаратные криптографические системы. Шифрование может выполняться с помощью программного или аппаратного обеспечения, у каждого варианта есть свои преимущества и недостатки. Обычно, программные реализации дешевле, но медленнее аппаратных. Программные криптографические методы могут быть существенно проще изменять и отключать по сравнению с аппаратными системами, но это зависит от конкретного программного приложения и аппаратного устройства. Если компании нужно выполнять функции шифрования на высокой скорости, для нее более предпочтительны аппаратные решения.

Ссылки по теме:

- On Distributed Communications: IX Security, Secrecy, and Tamper-Free Considerations, Section 3, “Some Fundamentals of Cryptography,” by Paul Baran (Rand Corp., August 1964)

11. Стандарты электронной почты

Криптография, как и другие технологии, имеет отраслевые стандарты и стандарты де-факто. Стандарты крайне важны, поскольку они помогают обеспечить совместимость продуктов различных производителей. Наличие стандарта обычно означает, что соответствующая технология прошла детальный анализ, надлежащим образом протестирована и принята в сообщество подобных технологий. Компании по-прежнему нужно решить, каким стандартам следовать и какие технологии внедрять.

Для принятия решения, компании следует оценить функциональность технологии, провести анализ затрат и выгод в отношении конкурирующих продуктов в рамках выбранных стандартов. Перед выбором технологий криптографии, компания должна определить, что должно быть защищено с помощью шифрования, требуется ли цифровая подпись, каким образом должно осуществляться управление ключами, какие доступны ресурсы для внедрения и поддержки выбранной технологии, какова общая стоимость всего этого.

Если компании просто нужно шифровать некоторые сообщения электронной почты, наилучшим выбором может быть PGP. Если компания хочет шифровать все данные, передаваемые по каналам связи в свои филиалы, лучшим выбором может быть внедрение канального шифрования. Если компания хочет внедрить технологию SSO для аутентификации пользователей при использовании ими различных сервисов и ресурсов сети, лучшим выбором может быть внедрение PKI или Kerberos. Сетевой администратор должен разбираться в каждой из таких технологий и стандартов, что позволит ему принимать взвешенные решения, анализировать и тестировать конкурирующие продукты в рамках выбранной технологии перед покупкой. Криптография сложна сама по себе, но также сложным является процесс ее внедрения и последующего сопровождения. Проведение качественного и всестороннего анализа присутствующих на рынке продуктов, вместо покупки самого красивого и блестящего продукта, может существенно снизить головную боль компании в дальнейшем.

В следующих разделах рассмотрены некоторые наиболее популярные стандарты электронной почты.

11.1. MIME

MIME (Multipurpose Internet Mail Extension – многоцелевое расширение функций электронной почты в Интернете) – это техническая спецификация, определяющая способы передачи мультимедийных данных и вложений в сообщениях электронной почты. В Интернете существуют почтовые стандарты, определяющие форматирование сообщений электронной почты, их инкапсуляцию, передачу и открытие. Если сообщение или документ содержит двоичное вложение, MIME указывает, как должна обрабатываться эта часть сообщения.

Если вложение содержит аудио-файл, графическое изображение или иной мультимедийный компонент, почтовый клиент отправляет этот файл с заголовком, описывающим тип файла. Например, заголовок может указывать, что MIME-типом файла является «изображение», а подтипом – «jpeg». Хотя это должно быть в заголовке, системы часто используют расширение файла для указания MIME-типа. Так, в предыдущем примере имя файла может быть `stuff.jpeg`. Система пользователя увидит расширение `jpeg` или поле заголовка, после чего начнет поиск ассоциированной с этим расширением программы, которой следует открывать этот конкретный файл. Если в системе пользователя файл JPEG ассоциирован с MS Paint, именно MS Paint должен открыть и показать изображение пользователю.

Иногда могут возникать ситуации, когда у системы отсутствует ассоциация для полученного типа файла или не установлена программа, необходимая для его просмотра и использования. В этом случае пользователю предоставляется возможность сделать выбор программы самостоятельно (либо установить необходимую программу).

MIME – это спецификация, которая определяет, как файлы определенных типов должны передаваться и обрабатываться. Эта спецификация содержит множество типов и подтипов, позволяющих различным компьютерам обмениваться данными в различных форматах, обеспечивая при этом стандартизированный подход к представлению данных.

S/MIME (Secure MIME - безопасный MIME) – это стандарт шифрования и подписи электронной почты, предназначенный для безопасной передачи данных. S/MIME расширяет стандарт MIME, позволяя шифровать электронную почту и вложения. Алгоритмы шифрования и хэширования могут быть указаны в почтовом сообщении пользователя, они не диктуются стандартом. S/MIME следует Стандартам криптографии с открытыми ключами (PKCS – Public Key Cryptography Standard). S/MIME обеспечивает конфиденциальность с помощью алгоритмов шифрования, целостность – с помощью алгоритмов хэширования, аутентификацию – с помощью сертификатов открытых ключей X.509, а неотказуемость – с помощью подписанных дайджестов сообщений.

11.2. PEM

PEM (Privacy-Enhanced Mail – электронная почта повышенной секретности) – это интернет-стандарт, обеспечивающий безопасную передачу электронной почты через Интернет и внутреннюю коммуникационную инфраструктуру. Протоколы, входящие в состав PEM, обеспечивают аутентификацию, целостность сообщений, шифрование и управление ключами. Этот стандарт совместим со многими типами процессов управления ключами, а также с симметричными и асимметричными методами шифрования. Кроме того, стандарт совместим с PKCS.

PEM – это набор технологий аутентификации и шифрования сообщений, разработанный несколькими правительственными группами. PEM может использовать AES для шифрования и RSA для аутентификации отправителя и управления ключами. Кроме того, он обеспечивает неотказуемость. В составе PEM могут использоваться следующие компоненты:

- Шифрование сообщений с помощью AES в режиме CBC
- Управление ключами с помощью RSA

- Структура и формат сертификатов на основе стандарта X.509

PEM не привлек большого внимания разработчиков. Основной проблемой PEM является то, что он предоставляет слишком много структур для различных сред, что требует большой гибкости в инфраструктуре защищенных коммуникаций.

11.3. MSP

MSP (Message Security Protocol - Протокол безопасности сообщений) – это военный вариант PEM, разработанный Агентством национальной безопасности США. Он является протоколом прикладного уровня, совместимым с X.400, и используется для безопасного обмена сообщениями электронной почты. MSP может использоваться для подписи и шифрования сообщений, выполнения функций хэширования. Аналогично PEM, приложения, в которых реализован MSP, позволяют использовать различные алгоритмы и параметры для обеспечения большой гибкости.

Ссылки по теме:

- Encryption and Security Tutorial, by Peter Gutmann
- Defective Sign & Encrypt in S/MIME, PKCS#7, MOSS, PEM, PGP, and XML, by Don Davis

11.4. PGP

PGP (Pretty Good Privacy) – бесплатная программа, разработанная Филом Циммерманом и предназначенная для обеспечения безопасности электронной почты. Первая версия PGP была выпущена в 1991 году. Эта была первая программа шифрования с открытыми ключами, получившая широкое распространение. PGP – это комплексная криптосистема, обеспечивающая защиту сообщений электронной почты и файлов. Стандартно в PGP для управления ключами используется асимметричный алгоритм RSA, а для основной массы операций шифрования – симметричный шифр IDEA, хотя пользователю предоставляется возможность для выбора иных алгоритмов для этих функций. PGP обеспечивает конфиденциальность с помощью алгоритма шифрования IDEA, целостность – с помощью алгоритма хэширования MD5, аутентификацию – с помощью сертификатов открытого ключа, и неотказуемость – с помощью цифровой подписи сообщений. В PGP применяется свой собственный тип цифровых сертификатов, отличный от используемых в PKI, но оба они имеют схожее назначение.

Для генерации и шифрования закрытого ключа, пользователю предлагается некоторое время нажимать на клавиатуре случайные клавиши. Вместо паролей в PGP используются парольные фразы. Парольные фразы применяются для шифрования закрытого ключа пользователя, хранящегося на его жестком диске.

PGP не использует иерархию Удостоверяющих центров или другие типы формально доверенных сертификатов, вместо этого в PGP для управления ключами применяется подход «сеть доверия» (web of trust). Каждый пользователь создает и распространяет свой открытый ключ, пользователи подписывают открытые ключи друг друга, создавая сообщество, в рамках которого все пользователи доверяют друг другу. Это отличается от подхода с Удостоверяющим центром, при котором никто не доверяет друг другу – все доверяют только Удостоверяющему центру. Например, Марку и Джо нужно взаимодействовать с помощью PGP. Марк передает свой открытый ключ Джо. Джо подписывает ключ Марка и сохраняет у себя копию, после чего передает Марку копию своего ключа, чтобы начать процесс безопасного взаимодействия. Предположим, что позднее у Марка возникает потребность взаимодействовать с Салли, но Салли не знает Марка и не знает, можно ли доверять ему. Марк отправляет Салли свой открытый ключ, подписанный Джо. У Салли есть открытый ключ Джо, т.к. они взаимодействовали раньше, и она доверяет Джо. Поскольку Джо подписал открытый ключ Марка, Салли теперь также может доверять Марку и отправляет ему свой

открытый ключ, чтобы начать процесс безопасного взаимодействия с ним. Таким образом, PGP как бы говорит: «Я не знаю тебя, но мой друг Джо говорит, что ты хороший парень и тебе можно доверять».

Каждый пользователь хранит в файле, называемом «*кольцом для ключей*» (key ring), набор открытых ключей, которые он получил от других пользователей. Каждый ключ в этом файле имеет параметры, определяющие его достоверность (validity) и уровень доверия к соответствующему пользователю. Например, если Стив знает Лизу многие годы и доверяет ей, он может указать более высокий уровень доверия к ее открытому ключу, чем к ключу Тома, которому больше никто не доверяет. Кроме того, предусмотрено поле, определяющее, кто может подписывать другие ключи в области доверия Стива. Если Стив получает ключ от кого-то, кого он не знает, например, от Кевина, и этот ключ подписан Лизой, программа проверяет вышеуказанное поле, в котором перечислены те, чьим подписям на других ключах он доверяет. Если в этом поле указано, что Стив достаточно доверяет Лизе, чтобы принимать подписанные ей ключи других людей, Стив может принять ключ Кевина и взаимодействовать с ним, поскольку Лиза поручилась за него. Однако если Стив получит ключ Кевина, подписанный недоверенным Томом, Стив может решить не доверять Кевину и не взаимодействовать с ним.

Эти поля доступны для обновления и изменения. Если однажды Стив лучше узнает Тома и поймет, что ему можно доверять, он может изменить соответствующие параметры в PGP и предоставить Тому больше доверия.

Поскольку сеть доверия не имеет центра, такого как Удостоверяющий центр, реализовать на базе нее стандартизованную функциональность сложнее. Если Стив потеряет свой закрытый ключ, он должен будет уведомить об этом каждого, т.к. его открытому ключу больше нельзя доверять. В среде PKI Стиву нужно было бы уведомить об этом только Удостоверяющий центр, который разместит сертификат открытого ключа Стива в списке отозванных сертификатов (CRL), с помощью которого любой, кто будет проверять действительность открытого ключа Стива сразу узнает, что ему нельзя доверять. В мире PGP это не централизовано и не организовано. Стив может отправить сертификат отзыва ключа (key revocation certificate), но нет гарантий выполнения соответствующих изменений в файле с ключами на компьютере каждого пользователя.

PGP – это общедоступное программное обеспечение, использующее криптографию с открытыми ключами. Хотя PGP не получил одобрения Агентства национальной безопасности США, этот прекрасный и к тому же бесплатный продукт получил широкое распространение для личного пользования, став фактически стандартом шифрования в Интернете.

ПРИМЕЧАНИЕ. PGP считается криптосистемой, поскольку он имеет все необходимые компоненты: алгоритмы с симметричными ключами, алгоритмы с асимметричными ключами, алгоритмы хэширования, ключи, протоколы и необходимые программные компоненты.

Ссылки по теме:

- Introduction to Cryptography, Chapter 1, “How PGP Works”
- The International PGP home page

11.5. Квантовая криптография

Сегодня мы обладаем очень сложными и стойкими алгоритмами, стойкости которых более чем достаточно для большинства современных вариантов использования, включая финансовые транзакции и обмен секретной информацией. Однако некоторые типы передаваемых данных все же имеют столь высокую критичность и настолько сильно востребованы отдельными субъектами, обладающими значительными ресурсами, что стойкости современных алгоритмов может быть недостаточно для них. Такие данные могут

быть перехвачены посредством шпионажа, информационных войн и т.п. Когда целая страна хочет взломать зашифрованные другой страной данные, она может привлечь к этому огромные ресурсы, что может привести к взлому современных алгоритмов.

Для удовлетворения потребностей в более стойких криптографических алгоритмах, умные люди смешали криптографию с квантовой физикой, что позволило получить системы, которые при правильной реализации являются невзламываемыми, а действия злоумышленников в них могут быть выявлены. В традиционной криптографии, мы пытаемся создать очень сложные алгоритмы, создавая максимум трудностей для злоумышленника, пытающегося взломать алгоритм шифрования или вскрыть ключ, но эти алгоритмы не позволяют выявить факты действий злоумышленника. В квантовой криптографии реализуется не только экзастойкое шифрование, но и *возможности* для выявления злоумышленников.

Квантовая криптография может быть реализована несколькими различными способами. Мы очень упрощенно рассмотрим один из вариантов, чтобы вы смогли понять, как она работает.

Предположим, что Том и Кэти – шпионы, и им нужно передавать друг другу данные, будучи уверенными, что они не будут перехвачены. Для этого на обеих сторонах им нужно использовать симметричный ключ шифрования: одна копия ключа для Тома и одна – для Кэти.

В **квантовой криптографии** для представления бита (1 или 0) используется поляризация фотона. *Поляризация* – это ориентация электромагнитных волн, которыми являются фотоны. Электромагнитные волны могут иметь ортогональную поляризацию (т.е. быть ориентированы в горизонтальной или вертикальной плоскости) или диагональную (т.е. быть наклонены влево или вправо).

Теперь представим, что у Кэти и Тома есть свои фотонные пушки, которые они будут использовать для обмена фотонами (информацией) друг с другом. Должна быть установлена связь между поляризацией фотона и двоичным значением. Поляризация может быть вертикальной (|), горизонтальной (–), левой (\\) или правой (/), а поскольку у нас только два двоичных значения, возникает некоторое перекрытие.

В рамках этого примера будем считать, что фотон с вертикальной поляризацией соответствует двоичному значению 0, левая поляризация – 1, правая – 0, а горизонтальная – 1. Эта «привязка» (кодирование) к двоичным значениям и является ключом шифрования. Том должен знать эту схему «привязки» поляризации фотонов к двоичным значениям, чтобы иметь возможность правильно интерпретировать информацию, которую ему отправляет Кэти. Если Том получит фотон с левой поляризацией, он декодирует его как «1», если он получит фотон с вертикальной поляризацией – то как «0». Производя декодирование сообщения с помощью такого ключа, он получит информацию, которую ему отправила Кэти.

Таким образом, для обмена информацией им необходимо договориться о ключе, которым будет являться соответствие между состояниями поляризации фотонов и двоичным представлением передаваемой информации. Это необходимо сделать заранее, до начала сеанса передачи информации по выделенной оптической линии. После того, как соглашение о симметричном ключе достигнуто сторонами, этот ключ может использоваться ими для зашифрования и расшифрования данных, передаваемых по открытым коммуникационным каналам. Случайность поляризации и сложность создания симметричного ключа способствует обеспечению того, что злоумышленник не может вскрыть ключ шифрования.

Поскольку этот тип криптографии основан на квантовой физике, а не на точной математике, отправитель и получатель могут быть уверены, что никакой злоумышленник не прослушивает коммуникационный канал и не выполняет атаку «человек посередине». Это связано с тем, что квантовый уровень – это уровень характеристик атомных и субатомных частиц. Если злоумышленник будет выполнять пассивную атаку, прослушивая

коммуникационный канал (сниффинг), получатель сразу узнает об этом, т.к. даже это простое действие изменит характеристики (поляризацию) фотонов.

Некоторые эксперты считают, что квантовая криптография уже реально используется в американском Белом доме и Пентагоне, а также при передаче данных между военными базами. Информация об этом засекречена американским правительством.

Ссылки по теме:

- Quantum Cryptography
- Quantum Cryptography

12. Безопасность в сети Интернет

Веб (web) – это не то же самое, что Интернет. Можно сказать, что веб работает над Интернетом. Веб – это множество HTTP-серверов, на которых хранятся и работают используемые нами веб-сайты. А Интернет – это множество физических устройств и коммуникационных протоколов, используемых для работы этих сайтов и взаимодействия с ними. Внешний вид и поведение веб-сайтов зависит от использованного разработчиками языка, который определяет порядок взаимодействия с сайтом и его функциональность. Веб-браузеры позволяют пользователям просматривать веб-страницы, преобразуя этот язык (HTML, DHTML, XML и т.д.) в понятный и удобный для человека вид. Браузер – это окно пользователя во «всемирную паутину» (World Wide Web).

Браузеры понимают и могут обрабатывать множество различных протоколов и команд, но не все. Если браузер пользователя не может работать с определенным протоколом или набором команд, пользователь должен скачать и установить специальную программу просмотра или плагин (модульный компонент, который интегрируется в систему или браузер). Это быстрый и простой способ расширения функциональности браузера. Однако установка плагинов может привести к нарушению безопасности, поскольку такие модули могут содержать вирусы и другое нежелательное программное обеспечение, которое пользователь не заметит, пока не будет уже слишком поздно.

12.1. Начнем с основ

Зачем мы подключаемся к Интернету? Чтобы скачать музыку, проверить электронную почту, заказать книги по безопасности, посмотреть веб-сайты, пообщаться с друзьями, выполнить другие задачи. Но что мы в действительности делаем? Мы используем различные сервисы, предоставляемые компьютерными протоколами и программами. Такими сервисами может быть передача файлов через FTP, удаленные подключения через Telnet, интернет-соединения через HTTP, безопасные соединения через SSL и многое другое. Без этих протоколов мы не можем использовать ресурсы Интернет.

Руководство компании должно принять решение, какие функции сети Интернет должны быть доступны сотрудникам, а администратор должен реализовать это решение на практике и организовать контроль использования сервисов снаружи и внутри сети. Использование сервисов может быть ограничено различными способами, например: разрешение запуска в системе только определенных сервисов и ограничение доступа к системе; использование только безопасных версий сервисов; фильтрация запросов к сервисам; блокирование нежелательных сервисов. Выбор необходимых сервисов и способов ограничения доступа к остальным сервисам определяет безопасность и указывает технологии, которые нужны для обеспечения защиты.

Рассмотрим основные технологии и протоколы, из которых состоит «всемирная паутина».

HTTP

TCP/IP – это набор протоколов сети Интернет, а HTTP – это протокол веб. HTTP находится

наверху TCP/IP. Когда пользователь нажимает на ссылку на веб-странице, его браузер использует HTTP для отправки запроса на веб-сервер, на котором размещен веб-сайт. Веб-сервер находит соответствующий файл и отправляет его пользователю также с помощью HTTP. Но где же здесь TCP/IP? Протокол TCP управляет процедурой «рукопожатия» и поддерживает соединение между пользователем и сервером, а протокол IP гарантирует правильную маршрутизацию и доставку файлов через Интернет пользователю. Таким образом, протокол IP находит способ передачи данных между веб-сервером и пользователем, TCP обеспечивает корректность отправителя и получателя, а HTTP представляет содержимое, которым является веб-страница.

Протокол HTTP не создает постоянных соединений – клиент и сервер создают и разрывают временное соединение для каждой операции. Когда пользователь запрашивает определенную веб-страницу, веб-сервер находит эту страницу, предоставляет ее пользователю и сразу разрывает соединение. Если пользователь нажимает на ссылку на полученной веб-странице, устанавливается новое соединение, на веб-сервер отправляется новый запрос, веб-сервер в ответ отправляет запрошенную страницу и снова разрывает соединение. Веб-сервер не запоминает пользователей, запрашивающих у него веб-страницы, поскольку это потребовало бы гораздо больше ресурсов.

HTTPS

HTTPS (HTTP Secure) – это HTTP, работающий через SSL (HTTP работает на прикладном уровне, а SSL – на транспортном). SSL (Secure Socket Layer) использует криптографию с открытым ключом и обеспечивает шифрование данных, аутентификацию сервера, целостность сообщений и (опционально) аутентификацию клиента. Веб-сайт может одновременно иметь как безопасную, так и открытую часть. Для доступа в безопасную часть может требоваться аутентификация пользователя. При переходе пользователя с открытой страницы веб-сайта на безопасную страницу, веб-сервер обращается к SSL для защиты передаваемых данных.

Сервер возвращает сообщение обратно клиенту, указывая, что необходимо установить безопасное соединение, а клиент в своем ответе направляет параметры безопасности. Сервер выполняет сверку этих параметров безопасности со своими собственными, пока не найдет совпадение. Это и есть фаза «рукопожатия» (handshaking phase). Аутентификация сервера клиентом производится с помощью цифрового сертификата сервера. Если клиент принимает решение о доверии серверу, процесс продолжается. Сервер также может потребовать цифровой сертификат клиента для выполнения аутентификации клиента (взаимная аутентификация), но это делается довольно редко.

Клиент генерирует сеансовый ключ и зашифровывает его на открытом ключе сервера. Зашифрованный ключ направляется веб-серверу, который расшифровывает его на своем закрытом ключе. Теперь они оба имеют симметричный сеансовый ключ, который в дальнейшем используют для шифрования данных, передаваемых в обоих направлениях. Таким образом создается безопасный канал.

SSL поддерживает созданный коммуникационный канал в открытом состоянии, пока одна из сторон не пришлет запрос на окончание сеанса. Обычно окончание сеанса инициируется клиентом, который отправляет на сервер пакет FIN, указывающий на необходимость закрытия соединения.

Для использования SSL требуется его поддержка как сервером, так и браузером пользователя. SSL обеспечивает безопасность соединений, но не безопасность полученных данных, т.е. данные зашифрованы только в процессе их передачи, но не после их доставки на компьютер получателя. Таким образом, если пользователь отправляет в банк свою финансовую информацию через защищенное SSL-соединение, он может быть уверен в

защите этих данных в процессе передачи, но он должен доверять банку, получившему эту информацию.

Пользователь может убедиться, что используется безопасное соединение, взглянув на адрес сайта и проверив, что в начале строки адреса присутствует `https://`. Также пользователю следует проверить наличие в нижнем углу окна браузера изображения с закрытым замком или ключом (зависит от конкретного браузера).

В стеке протоколов, SSL расположен ниже прикладного уровня, но выше сетевого уровня. Это обеспечивает независимость SSL от конкретных прикладных протоколов и возможность использования им транспортных коммуникационных стандартов Интернета. В различных источниках SSL может быть расположен на различных уровнях модели OSI, что может вызвать непонимание. Нужно учитывать, что модель OSI – это концептуальная конструкция, которая пытается описать сетевую реальность. В действительности SSL состоит из двух протоколов: один работает внизу сеансового уровня, а другой работает наверху транспортного уровня. Именно поэтому одни источники ставят SSL на сеансовый уровень, а другие – на транспортный. Для сдачи экзамена CISSP запомните, что SSL работает на транспортном уровне.

Хотя SSL практически всегда используется в паре с HTTP, он также может использоваться и с другими протоколами. Если вы видите название обычного протокола, к которому прибавлено окончание «s», это безопасная версия протокола, использующая SSL для шифрования передаваемых данных.

Текущей версией SSL является версия 3.0. Поскольку SSL был разработан Netscape, он не является открытым протоколом. В связи с этим расширение функциональности SSL не является простой задачей. В спецификации и функции закрытого, защищенного авторскими правами протокола, независимые разработчики не могут вносить изменений. Открытой версией протокола SSL является протокол TLS (Transport Layer Security). Различия между SSL 3.0 и TLS незначительны, однако TLS имеет значительно больше возможностей для расширения функциональности и при этом он обратно совместим с SSL.

Secure HTTP

Между **S-HTTP** (Secure HTTP) и HTTPS (HTTP Secure) есть существенные различия, хотя их названия очень похожи. S-HTTP – это технология, которая защищает каждое сообщение, передаваемое между двумя компьютерами, тогда как HTTPS защищает коммуникационный канал между двумя компьютерами, в том числе сообщения и все остальное. HTTPS использует SSL и HTTP для организации защищенного канала между клиентом и сервером. S-HTTP следует использовать при необходимости шифрования отдельных сообщений, но если нужно шифровать всю информацию, передаваемую между двумя компьютерами, следует использовать HTTPS.

Ссылки по теме:

- “SSL/TLS Strong Encryption: An Introduction,” Apache Software Foundation, with permission from Frederick J. Hirsch

SET

SET (Secure Electronic Transaction – защищенные электронные транзакции) – это технология безопасности, предложенная Visa и MasterCard для повышения безопасности транзакций с банковскими картами. Ожидалось, что SET одержит быструю победу и вскоре будет принят в качестве стандарта и повсеместно внедрен. Хотя SET и обеспечивает эффективный способ передачи информации банковских карт, компании и пользователи не восприняли его, поскольку требовалось координировать работу различных сторон, каждому участнику нужно было устанавливать и настраивать много дополнительного (обновленного) программного

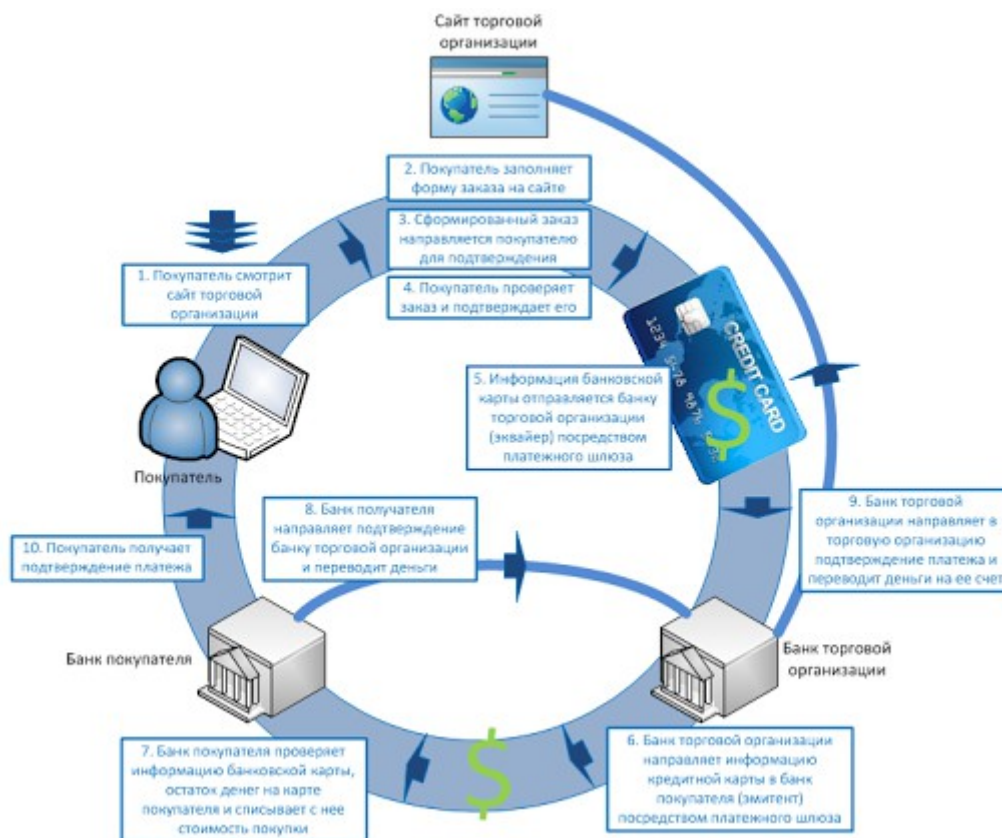
обеспечения, к тому же это требовало значительно больших усилий и затрат, чем широко используемый метод, основанный на SSL.

SET – это криптографический протокол и инфраструктура, он разработан для передачи через Интернет зашифрованных номеров кредитных карт. В транзакциях SET принимают участие следующие субъекты, которые должны обновить свое программное обеспечение и, вероятно, аппаратные устройства:

- **Эмитент (Issuer, банк держателя карты).** Финансовая организация, предоставляющая банковские карты своим клиентам.
- **Держатель карты (Cardholder).** Человек, уполномоченный на использование банковской карты.
- **Торговая организация (Merchant).** Субъект, осуществляющий продажу товаров или услуг.
- **Эквайер (Acquirer, банк, в котором обслуживается торговая организация).** Финансовая организация, обрабатывающая платежи по банковским картам.
- **Платежный шлюз (Payment gateway).** Платежный шлюз принимает и обрабатывает платежи торговой организации. Эту функцию может выполнять эквайер.

Для использования SET, пользователь должен ввести номер своей банковской карты в электронный "бумажник". Эта информация сохраняется на жестком диске пользователя или на смарт-карте. Программное обеспечение электронного "бумажника" создает пару ключей – открытый и закрытый, которые используются для шифрования финансовой информации перед ее отправкой.

Предположим, что Таня хочет с помощью своей кредитной карты купить на интернет-сайте подарок маме. Для покупки найденного ей подарка она должна отправить зашифрованные данные своей банковской карты на веб-сервер торговой организации. Торговая организация не расшифровывает эти данные, она просто подписывает их своей цифровой подписью и пересылает в банк. Платежный сервер банка расшифровывает полученные данные, проверяет, что на карте Тани есть достаточно денег для выполнения транзакции, после чего переводит деньги со счета Тани на счет торговой организации. Затем платежный сервер отправляет торговой организации сообщение, которое подтверждает успешное выполнение транзакции, а также квитанцию Тани и торговой организации. На каждом этапе очередной участник проверяет цифровую подпись отправителя (от которого он получил данные), и ставит свою цифровую подпись перед отправкой данных следующему участнику, вовлеченному в процесс. Это требует наличия у всех участников цифровых сертификатов и работы в PKI.



Это значительно более безопасный способ для выполнения транзакций через Интернет, но на сегодняшний день достаточно того уровня безопасности, который обеспечивает SSL. Ни держатели кредитных карт, ни финансовые организации не чувствуют достаточных мотивов для перехода на новую, более сложную, технологию. Также это связано с большим объемом изменений, которые нужно произвести в современных технологических процессах, и финансовых затрат, необходимых для этого.

Куки

Куки (cookie) – это текстовые файлы, которые браузер хранит на жестком диске пользователя. Куки имеют множество различных вариантов использования, например, в некоторых случаях они применяются для сбора демографической информации или в рекламных целях. Когда пользователь переходит с сайта на сайт в Интернете, эти сайты могут записывать данные в куки, сохраняемые в системе пользователя. С помощью этих данных сайты могут отслеживать перемещения пользователя по Интернету, его привычки и предпочтения в отношении определенных сайтов. Например, Эмили посещает в Интернете в основном сайты по садоводству, и эти сайты (что весьма вероятно) записывают информацию о ее посещениях, а также об элементах сайтов, которые вызывают у нее наибольший интерес. Когда Эмили возвращается на тот же или открывает похожий сайт, он запрашивает куки с ее компьютера, и находит в них информацию о том, что раньше ее интересовали книги по садоводству. На основании этой информации сайт показывает ей ссылку на новую серию книг по садоводству. Это существенно увеличивает вероятность того, что Эмили выберет и купит понравившуюся книгу. Это одна из современных маркетинговых тактик.

Порядок использования куки определяют серверы веб-сайтов. Если пользователь добавляет товар в свою корзину покупок на сайте, данные об этом обычно записываются в куки. Затем, когда пользователь закончил выбирать товары и готов оплатить покупку, данные о выбранных им товарах извлекаются из соответствующего куки и рассчитывается итоговая сумма покупки.

Как было сказано ранее, протокол HTTP не устанавливает соединений, поэтому веб-сервер

не расходует память на хранение информации о предыдущих соединениях. Это является одной из причин использования куки. С помощью них сохраняются данные о предыдущих соединениях пользователя, которые могут использоваться последующими HTTP-соединениями.

Например, если вы выполняете банковские операции в системе интернет-банкинга, веб-сервер вашего банка отслеживает ваши действия с помощью куки. Если вы сначала заходите на открытую часть банковского сайта и смотрите информацию о местонахождении офисов, времени их работы, курсах валют, конфиденциальная информация не передается ни в одном из направлений. Как только вы запрашиваете доступ к своему банковскому счету, веб-сервер устанавливает соединение SSL и требует, чтобы вы прошли аутентификацию перед предоставлением вам доступа. Как только вы проходите аутентификацию, сервер создает куки, в который записывает информацию о вашей аутентификации и вашем счете. Сервер отправляет этот куки вашему браузеру, а он сохраняет куки на жестком диске или в памяти вашего компьютера.

ПРИМЕЧАНИЕ. Некоторые куки хранятся на жестком диске в виде текстовых файлов. Эти файлы не должны содержать критичной информации, такой как номера кредитных карт или пароли. Куки, в которых хранится критичная информация, обычно содержатся только в памяти компьютера и не записываются на жесткий диск.

Предположим, что вы проверяете свой текущий счет, выполняете с ним некие операции, а затем запрашиваете информацию о своем депозитном счете. Веб-сервер отправляет запрос вашему браузеру для проверки куки, чтобы убедиться, что вы уже были надлежащим образом аутентифицированы.

Большинство систем интернет-банкинга периодически запрашивает куки у вашего браузера, чтобы убедиться в отсутствии «человека посередине», перехватившего ваш сеанс взаимодействия с банком.

Также важно обеспечить, чтобы безопасные соединения были ограничены по времени. Для этого в куки добавляются штампы времени. Если вы работали на веб-сайте через SSL-соединение, а затем на время отошли от компьютера, не закрывая сайт, программное обеспечение сайта через определенное время разорвет соединение и вам потребуется снова пройти аутентификацию, чтобы продолжить его использование.

Основная часть данных, хранящихся в куки, связана с серверами соответствующих сайтов, но некоторые куки могут содержать имена пользователей и пароли для различных учетных записей пользователя на интернет-сайтах. Куки, содержащие критичную информацию, должны в обязательном порядке быть зашифрованы сервером распространяющего их сайта, но так происходит не всегда, поэтому атакующие могут найти критичную информацию на жестком диске пользователя и попытаться воспользоваться ей в собственных интересах. Некоторые люди настраивают браузер таким образом, чтобы он вообще не принимал куки. Хотя это обеспечивает высокий уровень защиты против различных угроз, связанных с куки, при этом снижается функциональность интернет-сайтов и удобство их использования. Некоторые сайты в обязательном порядке требуют использования куки, они необходимы им для корректной работы в процессе обслуживания пользователей.

ПРИМЕЧАНИЕ. Некоторые программные продукты позволяют ограничить использование куки, разрешая загружать только определенные их типы, скрывать идентификационные данные пользователя в процессе его путешествия по Интернету, скрывать адрес электронной почты пользователя и почтовые сервера, которые он использует, а также все остальное, что позволяет идентифицировать личность пользователя.

SSH

SSH (Secure Shell - Безопасная оболочка) – является разновидностью механизма туннелирования, обеспечивающего терминальный доступ к удаленным компьютерам. SSH – это программа и протокол, которые могут использоваться для входа на другие компьютеры

через сеть. К примеру Поль, работающий на компьютере А, может с помощью SSH получить доступ к файлам на компьютере Б, запускать на компьютере Б приложения, выполнять настройку операционной системы на компьютере Б, работая при этом на компьютере А и физически не касаясь компьютера Б. SSH обеспечивает аутентификацию и безопасную передачу данных через небезопасные каналы связи, такие как Интернет.

ПРИМЕЧАНИЕ. SSH также может использоваться для организации безопасных каналов передачи файлов и перенаправления портов.

SSH может заменить Telnet, FTP, rlogin, rhex и rsh – он может обеспечить ту же функциональность, что и перечисленные средства, но, в отличие от них, SSH делает это значительно более безопасно. SSH – это программа и набор протоколов, которые работают совместно для создания безопасного туннеля между двумя компьютерами. При установлении соединения оба компьютера проходят через процедуру «рукопожатия» и обмениваются (с помощью алгоритма Диффи-Хеллмана) сеансовым ключом, который используется на протяжении всего сеанса взаимодействия этих компьютеров для шифрования и защиты передаваемых данных. Шаги процесса установления соединения SSH показаны на Рисунке 6-26.

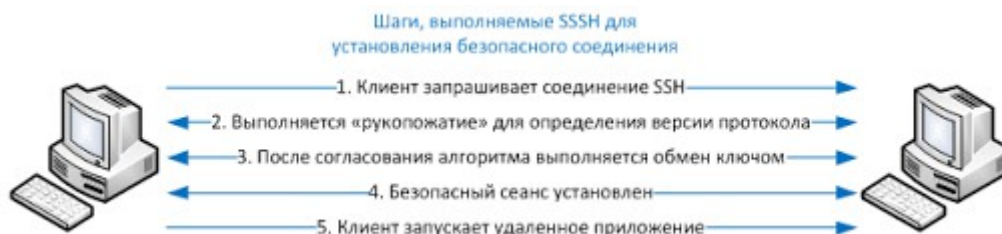


Рисунок 6-26. SSH используется для осуществления терминального доступа

После завершения «рукопожатия» и создания безопасного канала, два компьютера получают возможность для защищенного обмена данными, в процессе которого обеспечивается конфиденциальность и целостность передаваемой информации.

Ссылки по теме:

- SSH (Secure Shell) FAQ, by Thomas Koenig
- The Secure Shell Frequently Asked Questions, by Anne Carasik and Steve Acheson
- Unix System Administration, Chapter 29, “Secure Shell, SSH,” by Frank G. Fiamingo



IPSec

Набор протоколов **IPSec** (Internet Protocol Security) предоставляет способ создания защищенного канала для безопасного обмена данными между двумя устройствами. Такими устройствами, работающими через защищенный канал, могут быть два сервера, два маршрутизатора, рабочая станция и сервер, два шлюза между двумя различными сетями. IPSec – это общепринятый стандарт, обеспечивающий защиту на сетевом уровне. Он может быть более гибок и менее дорог, по сравнению с методами сквозного и канального шифрования.

В IPSec применяются стойкие методы шифрования и аутентификации. Обычно он применяется для создания VPN-туннелей между сетями через Интернет, хотя может

использоваться и для создания коммуникационных туннелей между отдельными компьютерами.

IPSec – это не жесткий протокол, диктующий тип алгоритма, ключей и используемых методов аутентификации. IPSec – это открытая модульная платформа, обеспечивающая большую гибкость для компаний, выбравших эту технологию. В состав IPSec входят два основных протокола безопасности: **AH** (Authentication Header - Аутентификация заголовка) и **ESP** (Encapsulating Security Payload - Безопасная инкапсуляция содержимого). AH – это протокол аутентификации, а ESP – протокол аутентификации и шифрования, используемый криптографическими механизмами для выполнения аутентификации источника, а также обеспечения конфиденциальности и целостности сообщений.

IPSec может работать в одном из двух режимов: **транспортный режим**, в котором защищено содержимое сообщений, и **туннельный режим**, в котором защищено не только содержимое сообщений, но и заголовки пакетов, и информация маршрутизации. При работе ESP в транспортном режиме, он выполняет шифрование только содержимого сообщений, что в случае их перехвата не позволит неуполномоченным лицам прочесть информацию. Туннельный режим обеспечивает более высокий уровень защиты, дополнительно шифруя заголовки и окончания пакетов данных, в которых атакующий может найти полезные для него сведения. Рисунок 6-27 в общих чертах показывает процесс установления соединения IPSec.

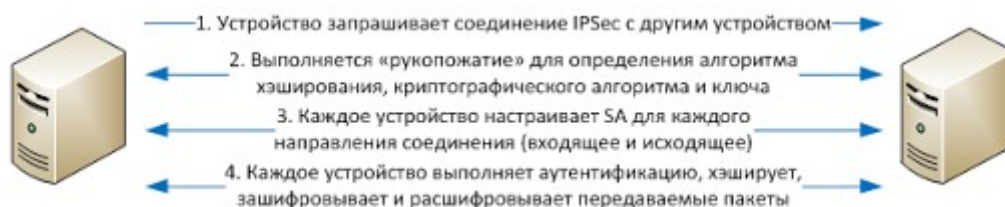


Рисунок 6-27. Шаги, выполняемые двумя устройствами для использования IPSec

Каждое устройство должно иметь одну или более **ассоциаций безопасности** (SA – security association) на каждое используемое им безопасное соединение. SA – это конфигурационная запись в настройках устройства, необходимая для работы соединения IPSec. SA играет одну из важнейших ролей в архитектуре IPSec. Когда два устройства завершили процесс «рукопожатия», в рамках которого они достигли соглашения по большому количеству различных коммуникационных параметров, эти параметры должны быть где-то сохранены. Именно для этого и нужна SA. SA может содержать ключи аутентификации и шифрования, информацию о согласованных алгоритмах, срок жизни ключа, IP-адрес отправителя. Когда устройство получает пакет посредством протокола IPSec, именно SA говорит устройству, что нужно делать с этим пакетом. Так, если устройство Б получает пакет от устройства В посредством IPSec, устройство Б должно посмотреть в соответствующую SA, чтобы узнать, как расшифровать пакет, как правильно аутентифицировать отправителя пакета, какой ключ использовать, как отвечать на сообщение при необходимости.

SA учитывает только одно направление передачи данных, поэтому устройство должно иметь для каждого отдельного коммуникационного канала одну SA для исходящего трафика и одну – для входящего. Таким образом, если устройство подключено к трем другим устройствам, оно должно иметь не менее шести SA – по две (одна для исходящего трафика, другая – для входящего) на каждое удаленное устройство. Каким образом устройства хранят SA и обеспечивают применение нужных SA для соответствующих соединений? Это осуществляется с помощью **индексов параметров безопасности** (SPI – security parameter index). На каждом устройстве есть SPI, который отслеживает различные SA и сообщает устройству, какой ему нужен SA для обработки того или иного пакета. Значение SPI указывается в заголовке пакета IPSec, устройство считывает это значение, чтобы найти нужный SA. Это изображено на Рисунке 6-28.



Рисунок 6-28. SPI и SA помогают системе обрабатывать пакеты IPsec

IPsec может аутентифицировать устройства, отправляющие пакеты, с помощью MAC (коды MAC были рассмотрены ранее в разделе «Односторонние хэши»). Протокол ESP может обеспечить аутентификацию, целостность и конфиденциальность (если устройстве включена и настроена эта функциональность). Таким образом, если компании достаточно просто обеспечить уверенность в источнике пакетов и в целостности пакетов, ей следует выбрать АН. Если компании помимо этих функций необходимо обеспечить конфиденциальность, ей следует использовать протокол ESP, т.к. он предоставляет функции шифрования. В большинстве случаев применение ESP обуславливается наличием у компании потребности в создании безопасных VPN-соединений.

Может показаться, что шифрование является дополнением в протоколе ESP, а в остальном функциональность АН и ESP пересекается. АН обеспечивает аутентификацию и целостность, а ESP может обеспечивать помимо этих функций еще и конфиденциальность. Зачем в таком случае нужен АН? В большинстве случаев это вызвано использованием NAT (трансляция сетевых адресов). IPsec генерирует контрольное значение целостности (ICV – integrity check value), которое в действительности является тем же значением MAC, вычисленным для части пакета. Помните, что отправитель и получатель генерируют свои собственные значения для проверки целостности? В IPsec эти значения называются ICV. Получатель сравнивает самостоятельно рассчитанное значение ICV с аналогичным значением, полученным от отправителя. Если значения совпадают, получатель может быть уверен, что пакет не был изменен в процессе передачи. Если значения отличаются, пакет был изменен и получатель отбрасывает его.

Протокол АН рассчитывает значение ICV как над самими данными, так и над транспортным и сетевым заголовками. Если затем пакет проходит через устройство NAT, это устройство изменяет IP-адрес отправителя пакета. Это его работа. Однако при этом часть содержимого пакета (заголовок сетевого уровня), включенная ранее в расчет значения ICV, изменяется и, если получатель сгенерирует свое значение ICV для полученного пакета, оно будет отличаться от значения ICV, указанного в пакете, и пакет будет автоматически уничтожен.

Протокол ESP выполняет аналогичные шаги, за исключением того, что он не включает заголовок сетевого уровня в расчет значения ICV. Когда устройство NAT изменяет IP-адрес отправителя, это не влияет на значение ICV, т.к. заголовок сетевого уровня в расчете ICV не участвует. Эти отличия показаны на Рисунке 6-29.

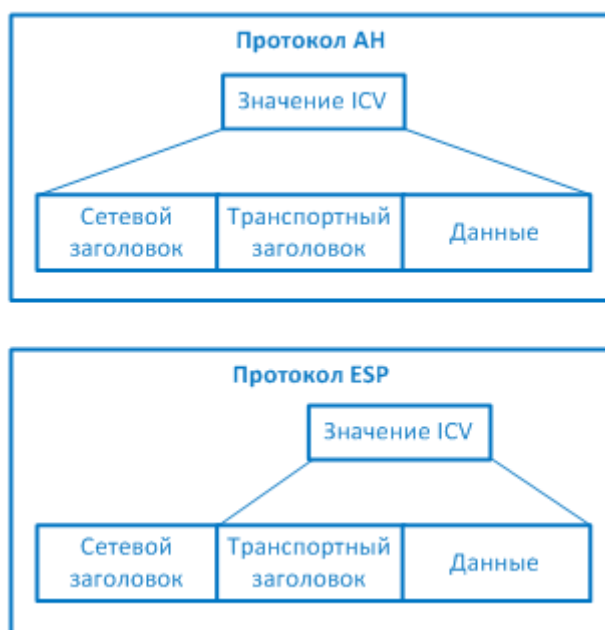


Рисунок 6-29. AH и ESP используют различные части пакета для расчета ICV

Поскольку IPSec является платформой, он не диктует конкретные алгоритмы хэширования и шифрования или процедуры обмена ключами. Управление ключами может выполняться вручную или автоматически с использованием протокола управления ключами. Стандартом «де-факто» для IPSec является использование IKE (Internet Key Exchange - Обмен ключами в Интернет), который является комбинацией протоколов ISAKMP и OAKLEY. **ISAKMP** (Internet Security Association and Key Management Protocol - Протокол управления ключами и ассоциациями безопасности в Интернете) – это архитектура обмена ключами, независимая от типа используемых механизмов, связанных с ключами. ISAKMP предоставляет платформу, соглашение о деталях использования которой достигается в процессе создания соединения IPSec (алгоритмы, протоколы, режимы, ключи). Протокол OAKLEY – это протокол, который как раз и реализует достижение этого соглашения. Представьте, что ISAKMP – это игровое поле (инфраструктура), а OAKLEY – это парень, бегающий по полю туда и обратно (выполнение шагов заключения соглашения).

ПРИМЕЧАНИЕ. SKIP (Simple Key Management Protocol for IP - Простой протокол обмена ключами для IP) – это другой протокол обмена ключами, обеспечивающий в основном аналогичную IKE функциональность. Важно знать, что все эти протоколы работают на сетевом уровне.

IPSec со всеми этими компонентами и различными возможными конфигурациями очень многообразен. Это многообразие обеспечивает высокую степень гибкости, поскольку компания имеет большой выбор конфигураций для достижения необходимого ей уровня защиты.

Ссылки по теме:

- “A Cryptographic Evaluation of IPSec,” by N. Ferguson and Bruce Schneier
- “An Introduction to IP Security (IPSec) Encryption,” Cisco Systems, Inc.

13. Атаки

Перехват и прослушивание передаваемых по сети данных называется **пассивной атакой**, т.к. атакующий не воздействует на протокол, алгоритм, ключ, само сообщение, какие-либо части системы шифрования. Пассивную атаку очень сложно обнаружить, в большинстве случаев проще попытаться предотвратить ее, чем выявить и остановить.

Активными атаками является изменение сообщений, изменение системных файлов,

попытки выдать себя за другого человека. При выполнении активных атак атакующий что-то реально делает, а не просто собирает данные. Пассивные атаки обычно используются для сбора информации перед проведением активной атаки. В следующих разделах рассмотрены некоторые активные атаки, имеющие отношение к криптографии.

13.1. Атака "Только шифротекст"

При выполнении атаки этого типа, атакующий имеет шифротекст нескольких сообщений. Каждое из сообщений зашифровано одним и тем же алгоритмом. Целью атакующего является вскрытие ключа, использованного в процессе шифрования. Если атакующий сможет вскрыть ключ, он сможет расшифровать все остальные сообщения, зашифрованные на том же ключе.

Атака «только шифротекст» (cipher-only attack) – это самый распространенный тип активных атак, поскольку получить шифротекст достаточно просто, например, прослушивая чей-то сетевой трафик. Однако это сложнейшая атака, в которой крайне сложно добиться успеха, поскольку атакующий имеет слишком мало информации о процессе шифрования.

13.2. Атака "Известный открытый текст"

При выполнении атаки типа «известный открытый текст» (known-plaintext attack), у атакующего есть открытый текст и соответствующий ему шифротекст одного или нескольких сообщений. Целью также является вскрытие ключа, использованного при шифровании этих сообщений, чтобы расшифровать и прочесть другие сообщения.

Обычно сообщения начинаются и заканчиваются одним и тем же текстом. Например, атакующий может узнать, что большинство сообщений сотрудников компании начинается с определенного приветствия и заканчивается подписью, в которую входит имя сотрудника, должность и контактная информация. Таким образом, атакующий имеет некоторый объем открытого текста (одинаковые данные в каждом сообщении) и может перехватить зашифрованное сообщение и извлечь из него шифротекст. Это позволит вскрыть несколько частей этой головоломки, а для завершения атаки нужно будет провести обратный инжиниринг, частотный анализ или брутфорс-атаку. Атаки типа «известный открытый текст» использовались США против Германии и Японии во Второй Мировой войне.

13.3. Атака "Выбранный открытый текст"

При выполнении атаки типа «выбранный открытый текст» (chosen-plaintext attack), у атакующего также есть открытый текст и соответствующий ему шифротекст, но он имеет возможность самостоятельно выбирать открытый текст и получать его в зашифрованном виде. Это дает атакующему дополнительные возможности для более глубокого изучения механизмов работы процесса шифрования, а также для сбора большего объема информации об используемом ключе. Если ему удастся вскрыть ключ, он сможет расшифровать другие сообщения, зашифрованные на этом ключе.

Как это делается? Например, атакующий может подготовить специальное сообщение, которое заставит получателя переслать его кому-то еще. Атакующий отправляет это сообщение пользователю, тот пересылает его своему коллеге, а почтовая программа на его компьютере автоматически зашифровывает сообщение перед отправкой. После этого атакующий перехватывает трафик пользователя и получает копию шифротекста к написанному им самим открытому тексту.

13.4. Атака "Выбранный шифротекст"

При выполнении атаки типа «выбранный шифротекст» (chosen-ciphertext attack), атакующий может выбирать шифротекст для расшифрования и имеет доступ к получаемому в результате открытому тексту. Целью опять же является вскрытие ключа. Это более сложная атака по сравнению с предыдущей. Для ее реализации атакующему может потребоваться контроль

над системой, содержащей криптосистему.

ПРИМЕЧАНИЕ. Для всех перечисленных выше типов атак, существуют аналогичные производные типы атак, в начале названия которых добавляется слова «адаптированная»: например, «адаптированная атака с выбранным открытым текстом» (adaptive chosen-plaintext attack) или «адаптированная атака с выбранным шифротекстом» (adaptive chosen-ciphertext attack). Слово «адаптированная» здесь означает, что атакующий выполняет одну из атак обычного типа, а затем, в зависимости от полученной в результате информации, изменяет свою следующую атаку. Так реализуются атаки с помощью обратного инжиниринга или криптоанализа – полученные знания используются для повышения эффективности последующей атаки.

Открытые и секретные алгоритмы. В настоящее время в мире в основном используются хорошо известные и понятные криптографические алгоритмы, а не секретные. Криптографы знают, насколько стойким и хорошо спроектированным должен быть алгоритм, представляемый на суд общественности. Тысячи умов лучше, чем пять, и часто это помогает найти в алгоритме проблемы, которые не заметили разработчики. Именно поэтому различные производители и компании устраивают соревнования по взлому их кодов и процессов шифрования. Если кому-то удастся их взломать, разработчики возвращаются к чертежной доске и усиливают ту или иную часть алгоритма.

Однако не все алгоритмы сделаны общедоступными, например, некоторые алгоритмы, разработанные Агентством национальной безопасности США, являются секретными. Поскольку уровень критичности данных, с которыми работают шифры АНБ, настолько велик, они хотят максимально сохранить процесс в секрете. АНБ не проводит публичных тестов и исследований своих алгоритмов, однако это не говорит о слабости алгоритмов АНБ. Эти алгоритмы разрабатываются, исследуются и тестируются лучшими криптографами, имеющими очень высокую квалификацию.

13.5. Дифференциальный криптоанализ

Целью атаки этого типа также является вскрытие ключа, использованного при шифровании. Эта атака анализирует пары шифротекста, созданного при зашифровании пар открытого текста с определенными различиями, и анализирует их воздействие и результат, получаемый в результате этих различий. Первая такая атака была проведена в 1990 году против алгоритма DES. В дальнейшем она эффективно и успешно применялась для взлома DES и других блочных алгоритмов.

Атакующий берет два сообщения в виде открытого текста и следит за изменениями, которые происходят с блоками при их прохождении через различные S-боксы. (Каждое сообщение зашифровывается на одном и том же ключе). Выявленные расхождения в значениях получаемого в результате шифротекста используются в качестве карты вероятных значений для различных возможных значений ключа. Атакующий выполняет этот процесс для максимально возможного набора различных сообщений и определяет вероятные значения ключа. Постепенно ключ проявляется, и это с высокой вероятностью будет именно тот ключ, который использовался в процессе шифрования. Поскольку атакующий для атаки выбирает сообщения в виде открытого текста, это является атакой с «выбранным открытым текстом».

13.6. Линейный криптоанализ

Линейный криптоанализ является другим вариантом атаки, направленной на выявление наиболее вероятного значения ключа, использованного в процессе шифрования блочным алгоритмом. Атакующий выполняет атаку «известный открытый текст» на несколько различных сообщений, зашифрованных на одном и том же ключе. Чем больше сообщений потенциально может использовать атакующий для этой атаки, тем вероятность нахождения правильного ключа.

Атакующий анализирует входящие и исходящие значения для каждого S-блока. Он анализирует вероятность того, что определенные входящие значения дают в результате определенную комбинацию. Выявление таких результирующих комбинаций позволяет ему оценивать вероятность для различных значений ключа, пока он не найдет повторяющийся шаблон, имеющий высокую вероятность.

13.7. Атаки с использованием побочных каналов

Все рассмотренные нами ранее атаки, основаны в первую очередь на математических аспектах криптографии. Использование открытого текста и шифротекста, а также применение мощных математических инструментов, направлено на вскрытие ключа, использованного в процессе шифрования.

Но существуют и другие методы. Предположим, мы видим какое-то животное, похожее на утку. Оно ходит как утка, издает звуки как утка, плавает в воде, ест жуков и маленьких рыб. Мы можем с уверенностью сделать вывод о том, что это утка. Так же и в криптографии, мы можем увидеть некоторые факты и сделать вывод о значении ключа. Например, мы можем измерить, сколько расходуется электроэнергии при зашифровании и расшифровании (по колебаниям электрического напряжения). Мы можем также перехватить создаваемые при этом излучения и затем рассчитать, сколько времени выполняются процессы. Анализ происходящего вне криптосистемы, измерение различных свойств и характеристик отличается от анализа того, что происходит в самой криптосистеме, но также дает данные, используя которые можно попытаться провести математические расчеты.

Если мне нужно узнать ваши привычки и предпочтения, но я не хочу, чтобы вы знали об этом, я не буду спрашивать вас напрямую. Вместо этого я прослежу, когда вы приходите на работу, когда уходите домой, какую одежду вы носите, что вы делаете, о чем говорите... Это тоже примеры получения информации с помощью побочных каналов (side-channel). Таким образом, в криптографии сбор «внешней» информации с целью вскрытия ключа шифрования – это просто еще один способ атаки на криптосистемы.

Атакующий измеряет потребляемую энергию, излучения, время обработки определенных данных. Полученную информацию он использует при проведении обратного инжиниринга процесса для вскрытия ключа шифрования или получения критичных данных. Например, атака, при которой анализируется потребляемая энергия, анализирует производимое тепло. Эта атака успешно применяется для взлома смарт-карт и получения с них конфиденциальной информации. В 1995 году закрытый ключ алгоритма RSA был взломан с помощью измерения относительного времени, которое расходуется на различные криптографические операции.

Идея этого заключается в том, что вместо атаки устройство, мы просто смотрим, как оно работает. В биологии, ученые часто проводят «бесконтактные» эксперименты, в процессе которых они смотрят, как организм питается, спит, спаривается и т.д. Ученые пытаются изучить организм путем анализа его поведения, вместо того, чтобы убить его и заглянуть внутрь.

13.8. Атаки повтора

Большое проблемой в распределенной среде являются **атаки повтора** (replay attack), при выполнении которых атакующий перехватывает определенные данные, а затем отправляет их снова, надеясь, что получившее их устройство примет их за легитимную информацию. Чаще всего атакующий пытается перехватить и повторно использовать аутентификационные данные, чтобы пройти аутентификацию в системе от имени легитимного пользователя и получить таким образом несанкционированный доступ к ней.

Контрмерами против атаки повтора является использование штампов времени и номеров последовательности. Пакеты могут содержать номера последовательности и каждое

получившее их устройство будет проверять эти номера. Если пакет имеет уже использовавшийся ранее номер, это указывает на атаку повтора. Также на пакеты может ставиться штамп времени. При этом на каждом компьютере настраивается пороговое значение, определяющее временной интервал, в рамках которого указанное в пакете время будет считаться корректным. Если в пакете указано время, выходящее за пределы этого интервала, это также может говорить об атаке повтора.

13.9. Алгебраические атаки

При выполнении алгебраической атаки (algebraic attack), атакующий анализирует уязвимости в математических частях алгоритма и использует его внутренние алгебраические структуры. Для примера, атака на версию «текстовой книги» криптосистемы RSA использует такие свойства алгоритма, как факт, что при шифровании 0 получается 0.

13.10. Аналитические атаки

Аналитические атаки (analytics attack) выявляют структурные слабости и недостатки алгоритма, вместо выполнения брутфорс-атаки, при которой просто перебираются все возможные значения, без учета специфических свойств алгоритма. Примерами являются Атака на Double DES и Атака разложения на множители в RSA.

13.11. Статистические атаки

Статистические атаки (statistical attack) выявляют статистические слабости в структуре алгоритма – например, если удастся выявить статистический шаблон, например, сравнивая количество значений «0» с количеством значений «1». Это может быть вызвано, например, использованием некачественного генератора случайных чисел. Если ключи берутся напрямую из выдачи RNG, распределение ключей может быть предсказуемым. Знания о статистической предсказуемости могут использоваться для снижения времени на поиск ключей.

Ссылки по теме:

- Wikipedia entry for ciphertext-only attack
- Frequently Asked Questions about Today's Cryptography, Version 4.1, Section 2.4.2, "What Are Some of the Basic Types of Cryptanalytic Attacks?" by RSA Laboratories
- "Linear Cryptanalysis: A Literature Survey," by Terry Ritter
- "Linear Cryptanalysis of Block Ciphers," by Edward Schaefer
- "Introduction to Side Channel Attacks," by Hagai Bar-El, Discretix Technologies Ltd.

14. Резюме

Криптография применяется в том или ином виде на протяжении 4000 лет, а атаки на нее выполняются примерно 3999 лет и 364 дня. Одна группа людей работает над поиском новых способов скрытия и передачи секретов, тогда как другая – занимается поиском дыр в новых идеях и продуктах. Хотя это выглядит нехорошим и деструктивным поведением, в компьютерном мире это способствует созданию все лучших и все более защищенных продуктов и сред.

Криптографические алгоритмы предоставляют низкоуровневые инструменты для большинства протоколов безопасности, используемых в современных инфраструктурах. Алгоритмы используют математические функции и предоставляют различные функции и уровни безопасности. Большой скачок произошел при переходе шифрования от чисто симметричных ключей к использованию криптографии с открытыми ключами. Это дало пользователям гораздо больше свободы и гибкости при взаимодействии с другими пользователями по всему миру.

Шифрование может выполняться на различных уровнях модели OSI широким спектром приложений, протоколов и механизмов. Сегодня уже не так просто забыть про криптографию или выполнении функций шифрования, поскольку многие операционные системы, приложения и протоколы заботятся об этом самостоятельно в фоновом режиме. Однако администраторам, занимающимся поддержкой таких сред, а также профессионалам по безопасности, которые выбирают и внедряют решения по безопасности, знание криптографии необходимо.

Тест

Вопросы экзамена CISSP являются концептуальными, поэтому они сформулированы соответствующим образом. Задачей кандидата является выбор наилучшего из всех представленных вариантов ответа. Среди вариантов ответа может не быть идеального ответа на поставленный вопрос - кандидат должен выбрать лучший ответ из имеющихся вариантов.

1. Что является целью криптоанализа?

- ☐ A. Определение стойкости алгоритма
- ☐ B. Увеличение количества функций замещения в криптографическом алгоритме
- ☐ C. Уменьшение количества функций подстановок в криптографическом алгоритме
- ☐ D. Определение использованных перестановок

2. Частота применения брутфорс-атак возросла, поскольку:

- ☐ A. Возросло используемое в алгоритмах количество перестановок и земещений
- ☐ B. Алгоритмы по мере повышения стойкости становились менее сложными и более подверженными атакам
- ☐ C. Мощность и скорость работы процессоров возросла
- ☐ D. Длина ключа со временем уменьшилась

3. Что из перечисленного ниже не является свойством или характеристикой односторонней функции хэширования?

- ☐ A. Она преобразует сообщение произвольной длины в значение фиксированной длины
- ☐ B. Имея значение дайджеста сообщения, невозможно получить само сообщение
- ☐ C. Получение одинакового дайджеста из двух различных сообщений невозможно, либо случается крайне редко
- ☐ D. Она преобразует сообщение фиксированной длины в значение переменной длины

4. Что может указывать на изменение сообщения?

- ☐ A. Изменился открытый ключ
- ☐ B. Изменился закрытый ключ
- ☐ C. Изменился дайджест сообщения
- ☐ D. Сообщение было правильно зашифровано

5. Какой из перечисленных ниже алгоритмов является алгоритмом американского правительства, предназначенным для создания безопасных дайджестов сообщений?

- ☐ A. Data Encryption Algorithm
- ☐ B. Digital Signature Standard
- ☐ C. Secure Hash Algorithm
- ☐ D. Data Signature Algorithm

6. Что из перечисленного ниже лучше всего описывает отличия между HMAC и CBC-MAC?

- ☐ A. HMAC создает дайджест сообщения и применяется для контроля целостности; CBC-MAC используется для шифрования блоков данных с целью обеспечения конфиденциальности
- ☐ B. HMAC использует симметричный ключ и алгоритм хэширования; CBC-MAC использует первый блок в качестве контрольной суммы
- ☐ C. HMAC обеспечивает контроль целостности и аутентификацию источника данных; CBC-MAC использует блочный шифр в процессе создания MAC
- ☐ D. HMAC зашифровывает сообщение на симметричном ключе, а затем передает результат в алгоритм хэширования; CBC-MAC зашифровывает все сообщение целиком

7. В чем преимущество RSA над DSA?

- ☐ A. Он может обеспечить функциональность цифровой подписи и шифрования
- ☐ B. Он использует меньше ресурсов и выполняет шифрование быстрее, поскольку использует симметричные ключи
- ☐ C. Это блочный шифр и он лучше поточного
- ☐ D. Он использует одноразовые шифровальные блокноты

8. Многие страны ограничивают использование и экспорт криптографических систем. Зачем они это делают?

- ☐ A. Без ограничений может возникнуть большое число проблем совместимости при попытке использовать различные алгоритмы в различных программах
- ☐ B. Эти системы могут использоваться некоторыми странами против их местного населения
- ☐ C. Криминальные элементы могут использовать шифрование, чтобы избежать обнаружения и преследования
- ☐ D. Законодательство сильно отстает, а создание новых типов шифрования еще больше усиливает эту проблему

9. Что используется для создания цифровой подписи?

- ☐ A. Закрытый ключ получателя
- ☐ B. Открытый ключ отправителя
- ☐ C. Закрытый ключ отправителя
- ☐ D. Открытый ключ получателя

10. Что из перечисленного ниже лучше всего описывает цифровую подпись?

- ☐ A. Это метод переноса собственноручной подписи на электронный документ
- ☐ B. Это метод шифрования конфиденциальной информации
- ☐ C. Это метод, обеспечивающий электронную подпись и шифрование
- ☐ D. Это метод, позволяющий получателю сообщения проверить его источник и убедиться в целостности сообщения

11. Какова эффективная длина ключа в DES?

- ☐ A. 56
- ☐ B. 64
- ☐ C. 32
- ☐ D. 16

12. По какой причине удостоверяющий центр отзывает сертификат?

- ☐ A. Если открытый ключ пользователя скомпрометирован
- ☐ B. Если пользователь переходит на использование модели РЕМ, которая использует сеть доверия
- ☐ C. Если закрытый ключ пользователя скомпрометирован
- ☐ D. Если пользователь переходит работать в другой офис

13. Что из перечисленного ниже лучше всего описывает удостоверяющий центр?

- ☐ A. Организация, которая выпускает закрытые ключи и соответствующие алгоритмы
- ☐ B. Организация, которая проверяет процессы шифрования
- ☐ C. Организация, которая проверяет ключи шифрования
- ☐ D. Организация, которая выпускает сертификаты

14. Как расшифровывается аббревиатура DEA?

- ☐ A. Data Encoding Algorithm
- ☐ B. Data Encoding Application
- ☐ C. Data Encryption Algorithm
- ☐ D. Digital Encryption Algorithm

15. Кто участвовал в разработке первого алгоритма с открытыми ключами?

- ☐ A. Ади Шамир
- ☐ B. Росс Андерсон
- ☐ C. Брюс Шнайер
- ☐ D. Мартин Хеллман

16. Какой процесс обычно выполняется после создания сеансового ключа DES?

- ☐ A. Подписание ключа
- ☐ B. Передача ключа на хранение третьей стороне (key escrow)
- ☐ C. Кластеризация ключа
- ☐ D. Обмен ключом

17. Сколько циклов перестановки и замещения выполняет DES?

- ☐ A. 16
- ☐ B. 32
- ☐ C. 64
- ☐ D. 56

18. Что из перечисленного ниже является правильным утверждением в отношении шифрования данных, выполняемого с целью их защиты?

- ☐ A. Оно обеспечивает проверку целостности и правильности данных
- ☐ B. Оно требует внимательного отношения к процессу управления ключами
- ☐ C. Оно не требует большого количества системных ресурсов
- ☐ D. Оно требует передачи ключа на хранение третьей стороне (escrowed)

19. Как называется ситуация, в которой при использовании различных ключей для шифрования одного и того же сообщения в результате получается один и тот же шифротекст?

- ☐ A. Коллизия
- ☐ B. Хэширование
- ☐ C. MAC
- ☐ D. Кластеризация ключей

20. Что из перечисленного ниже является определением фактора трудозатрат для алгоритма?

- ☐ A. Время зашифрования и расшифрования открытого текста
- ☐ B. Время, которое займет взлом шифрования
- ☐ C. Время, которое занимает выполнение 16 циклов преобразований
- ☐ D. Время, которое занимает выполнение функций подстановки

21. Что является основной целью использования одностороннего хэширования пароля пользователя?

- ☐ A. Это снижает требуемый объем дискового пространства для хранения пароля пользователя
- ☐ B. Это предотвращает ознакомление кого-либо с открытым текстом пароля
- ☐ C. Это позволяет избежать избыточной обработки, требуемой асимметричным алгоритмом
- ☐ D. Это предотвращает атаки повтора (replay attack)

22. Какой из перечисленных ниже алгоритмов основан на сложности разложения больших чисел на два исходных простых сомножителя?

- ☐ A. ECC
- ☐ B. RSA
- ☐ C. DES
- ☐ D. Диффи-Хеллман

23. Что из перечисленного ниже описывает разницу между алгоритмами DES и RSA?

- ☐ A. DES – это симметричный алгоритм, а RSA – асимметричный
- ☐ B. DES – это асимметричный алгоритм, а RSA – симметричный
- ☐ C. Они оба являются алгоритмами хэширования, но RSA генерирует 160-битные значения хэша
- ☐ D. DES генерирует открытый и закрытый ключи, а RSA выполняет шифрование сообщений

24. Какой из перечисленных ниже алгоритмов использует симметричный ключ и алгоритм хэширования?

- ☐ A. HMAC
- ☐ B. 3DES
- ☐ C. ISAKMP-OAKLEY
- ☐ D. RSA

25. Генерация ключей, для которой используются случайные значения, называется Функцией генерации ключей (KDF). Какие значения обычно не используются при этом в процессе генерации ключей?

- ☐ A. Хэши
- ☐ B. Асимметричные значения
- ☐ C. Соль
- ☐ D. Пароли

Домен 07. Непрерывность бизнеса и восстановление после аварий.

Мы не можем подготовиться к каждой потенциальной возможности, что доказали недавние события. В 2005 году ураган Катрина причинил огромный ущерб, он не просто затронул бизнес – были уничтожены многие здания, погибло много людей. Катастрофическое цунами в Индийском океане, произошедшее в декабре 2004 года, было полной неожиданностью. Падение башен Всемирного торгового центра после атаки террористов нарушило планы многих компаний, жителей США, правительства и даже всего мира – для большинства это было просто невообразимо. Каждый год тысячи компаний страдают от наводнений, пожаров, торнадо, террористических атак, вандализма. Все компании без исключений могут столкнуться с такими событиями, но лишь немногие пытаются подготовиться к ним, оценивают потенциальные последствия, внедряют необходимые защитные меры. Большинство компаний, в которых происходят подобные события, прекращают свое существование, поэтому всем компаниям следует подготовиться и иметь необходимые средства и процедуры специально для таких случаев.

Прибыль любой компании и ее существование на рынке зависят от ресурсов, персонала и непрерывного выполнения ежедневных задач. У большинства компаний есть материальные ресурсы, интеллектуальная собственность, компьютеры, коммуникационные каналы, здания. Если хотя бы что-то одно из этого перечня повреждено или недоступно по той или иной причине, компании может быть нанесен ущерб. Если повреждено или недоступно более одного пункта из этого списка, в компании может возникнуть чрезвычайная ситуация. Если чрезвычайная ситуация продолжается длительное время, это может стать катастрофой для компании. Многие компании уже никогда не восстанавливаются после катастроф. Однако компании, которые надлежащим образом подготовились к ним, и «не держат все яйца в одной корзине», имеют гораздо больше шансов продолжить свой бизнес и остаться на рынке после чрезвычайной ситуации или катастрофы.

1. Непрерывность бизнеса и восстановление после аварий

Целью *восстановления после аварий* (disaster recovery) является снижение влияния аварии и выполнение шагов, необходимых для максимально быстрого восстановления деятельности компании, доступности необходимых для этого ресурсов, бизнес-процессов и персонала. Это отличается от планирования непрерывности (continuity planning), которое относится к методам и процедурам, связанным с длительными простоями и авариями. Целью плана DRP (disaster recovery plan – план восстановления после аварий) является выполнение правильных действий сразу после возникновения аварии; обычно план DRP связан в первую очередь с информационными технологиями.

Работа по плану DRP выполняется, пока что-то находится в аварийном состоянии, и кто-то работает над тем, чтобы вернуть все критичные системы в нормальную работу. BCP (business continuity plan – план обеспечения непрерывности бизнеса) использует более широкий подход к этой проблеме. Работа по плану BCP включает перевод критичных систем в другую среду на время восстановления основной среды, обеспечение наличия нужных людей на нужных местах, выполнение основной деятельности компании в аварийном режиме, пока не будут восстановлены обычные условия. BCP также может включать в себя взаимодействие с клиентами, партнерами и акционерами с помощью различных каналов, пока все не вернется в нормальное состояние. DRP отвечает на вопрос: *«О, ужас! Небеса упали на землю! Как мы будем ставить их на место?»*, а BCP отвечает на другой вопрос: *«Итак, небеса упали на землю. Как наша компания будет продолжать свою деятельность, чтобы остаться на рынке, пока кто-то будет ставить небеса на место?»*.

Тема доступности, целостности и конфиденциальности проходит различные Домены этой книги. Каждый Домен по-своему рассматривает эти компоненты безопасности. Например, в

Домене 02, когда мы обсуждали управление доступом, понятие доступности означало, что ресурс должен быть доступен для использования пользователями и субъектами контролируемым и безопасным образом. Средства управления доступом должны защищать целостность и/или конфиденциальность ресурса. Фактически, средство управления доступом должно выполнять множество шагов для обеспечения конфиденциальности ресурса и отсутствия возможности несанкционированного изменения его содержимого в процессе использования. В этом Домене мы будем говорить о том, что целостность и конфиденциальность должны обеспечиваться не только при выполнении повседневных процедур, но и при выполнении процедур в случае возникновения аварии. Например, не следует оставлять без контроля сервер, на котором хранится конфиденциальная информация, когда все покинули здание, в котором он находится, и перешли в другое здание.

Также, важно отметить, что компания при возникновении аварии или чрезвычайной ситуации может стать значительно более уязвимой, поскольку используемые для ее защиты сервисы безопасности могут оказаться недоступны или работать с ограниченными возможностями. Поэтому для компании, которая имеет множество различных секретов, очень важно обеспечить конфиденциальность и целостность данных и систем даже в случае, если люди и сама компания находятся в сложном положении. Доступность – это один из основных аспектов планирования непрерывности бизнеса, требуется обеспечить гарантии постоянной доступности критичных ресурсов людям и системам, которым они необходимы. Для этого может потребоваться надлежащее выполнение резервного копирования, обеспечение избыточности в архитектуре систем, сетей и выполняемой деятельности. На случай недоступности коммуникационных каналов в течение длительного периода времени, должен существовать быстрый и проверенный способ создания альтернативных коммуникаций и связанных с ними сервисов.

При планировании непрерывности бизнеса, некоторые компании уделяют основное внимание вопросам резервного копирования и обеспечении избыточности используемого оборудования. Хотя эти вещи безусловно очень важны, они являются лишь маленькой частью огромной картины функционирования компании. Компьютеры и оборудование бесполезны без людей, которые настроят их и которые будут работать на них, а данные обычно бесполезны, если они недоступны другим системам и, возможно, внешним субъектам. Поэтому, картина совместной работы всех процессов в рамках всей компании должна быть изучена и понятна перед началом планирования непрерывности бизнеса. Планирование должно включать в себя, в частности, обеспечение наличия нужных людей на нужных местах, документирование необходимых конфигураций, создание альтернативных коммуникационных каналов (для голоса и данных), обеспечение резервного электроснабжения. Для этого необходима уверенность в том, что все зависимости и взаимосвязи, включая процессы и приложения, полностью понятны и учтены. Например, может оказаться невозможным восстановить работу критичного сервера приложения, если в сети недоступен контроллер домена или DNS-сервер.

Также важно понять, каким образом автоматизированные задачи при необходимости можно выполнять вручную, какие изменения можно безопасно произвести в организации бизнес-процессов для продолжения выполнения критичных для компании операций в чрезвычайной ситуации. Это может быть крайне важным для обеспечения уверенности, что компания сможет выжить при воздействии неблагоприятных обстоятельств, а их влияние на бизнес компании будет минимальным. Без понимания этих вопросов и надлежащего планирования, может оказаться, что в чрезвычайной ситуации у компании есть и резервные копии данных, и физически доступные отказоустойчивые сервера на альтернативной площадке, но люди, ответственные за введение их в работу, могут находиться в шоковом состоянии, и не знать, с чего начать и как выполнять обычную работу в альтернативной среде.

Планирование непрерывности бизнеса. Заранее запланированные процедуры позволяют компании:

- Обеспечить немедленную и правильную реакцию в чрезвычайной ситуации
- Защитить жизни людей и гарантировать их безопасность
- Минимизировать негативное влияние на бизнес компании
- Возобновить выполнение критичных для бизнеса функций
- Воспользоваться услугами внешних поставщиков на период восстановления
- Снизить беспорядок и неразбериху во время кризиса
- Обеспечить выживание бизнеса компании
- Обеспечить максимально быстрое восстановление функционирования после аварии

Должны быть приняты, в частности, следующие решения:

- Позволить бизнес-партнерам знать о готовности вашей компании к чрезвычайным ситуациям
- Повысить доверие со стороны акционеров и правления информированием их о готовности к чрезвычайным ситуациям
- Выполнить требования отраслевых регуляторов в области ВСР (при наличии таких требований)

1.1. Шаги планирования непрерывности бизнеса

Хотя нет четкого маршрута, по которому нужно пройти для создания плана ВСР, время от времени появляются различные лучшие практики на этот счет. В частности, NIST (National Institute of Standards and Technology - Национальный институт стандартов и технологий США) отвечает за разработку лучших практик и обеспечение общего доступа к ним. NIST предусмотрел следующие шаги в документе SP 800-34 «Руководство по планированию непрерывности для ИТ-систем» (<http://csrc.nist.gov/publications/nistpubs/800-34-rev1/sp800-34-rev1.pdf>):

1. *Разработать политику планирования непрерывности бизнеса* (continuity planning policy statement). Написать политику, которая будет содержать необходимые руководящие принципы для разработки плана ВСР и определит необходимые ролям полномочия для выполнения возложенных на них задач.
2. *Провести Анализ воздействия на бизнес* (BIA – business impact analysis). Идентифицировать критичные функции и системы, категорировать (приоритезировать) их на основе степени их критичности. Выявить уязвимости и угрозы, рассчитать риски.
3. *Определить превентивные защитные меры*. После выявления угроз, выбрать и внедрить защитные меры и контроли для снижения уровня рисков компании экономически целесообразным способом.
4. *Разработать стратегии восстановления* (recovery strategy). Описать методы, обеспечивающие оперативное восстановление работоспособности критичных систем и функций.
5. *Разработать план действий на случай чрезвычайных ситуаций* (contingency plan). Описать процедуры, разработать руководства, которые обеспечат продолжение функционирования компании в аварийном состоянии.
6. *Протестировать план, провести тренинги и учения*. Проверить план для выявления недостатков в нем, провести тренинги и учения для надлежащей подготовки людей к выполнению задач на случай чрезвычайной ситуации.
7. *Поддерживать актуальность плана*. Предпринять шаги для своевременной актуализации плана.

В различных руководствах и планах компаний предусматриваются шаги, аналогичные указанным в SP800-34, но они иногда используют несколько иную терминологию. Например, (ISC)2 использует следующие шаги:

1. Инициирование проекта
2. BIA
3. Стратегия восстановления
4. Проектирование и разработка плана
5. Внедрение
6. Тестирование
7. Постоянная поддержка

Сначала нужно понять, как работает компания. Компания не может рассчитывать на восстановление процессов после аварии, если у нее нет хорошего понимания того, как она (компания) работает. Это может показаться абсурдным на первый взгляд. Вы можете подумать: «Ну, конечно же, компания прекрасно знает как она работает!». Но вы будете удивлены тем, как на самом деле сложно полностью понимать на детальном (низком) уровне работу компании, детальные требования к воссозданию низкоуровневых процессов при необходимости. Каждый сотрудник компании знает и понимает свою маленькую часть общей работы компании, но изучить и полностью понять работу всех и каждого из бизнес-процессов компании – крайне сложная задача для любой компании. В задачи этой книги не входит детальное рассмотрение бизнес-процессов и корпоративной архитектуры, вы можете посмотреть хорошую модель здесь: www.intervista-institute.com/resources/zachman-poster.html. Это один из самых всеобъемлющих подходов к пониманию архитектуры компании и всех частей, из которых она состоит. Эта модель разбивает корпоративную архитектуру компании на ключевые компоненты, чтобы показать всевозможные требования к каждому бизнес-процессу. Она рассматривает такие компоненты инфраструктуры компании, как данные, функции, сети, персонал, время, мотивация, а также их взаимосвязь с ролями в компании. Преимущество этой модели состоит в том, что она разбивает бизнес-процессы до уровня атомов и показывает существующие взаимозависимости, каждая из которых должна работать надлежащим образом для эффективной и результативной работы процессов.

Модель на приведенной выше ссылке может помочь компании классифицировать различные корпоративные компоненты. На том же сайте содержатся и другие ресурсы, относящиеся к этой модели. Эта модель может оказаться очень полезной для команды ВСП, она может помочь понять ключевые компоненты компании, обеспечить уверенность что работа компании может быть воссоздана при необходимости.

Необходимые для реализации процесса планирования непрерывности бизнеса шаги показаны на Рисунке 7-1.

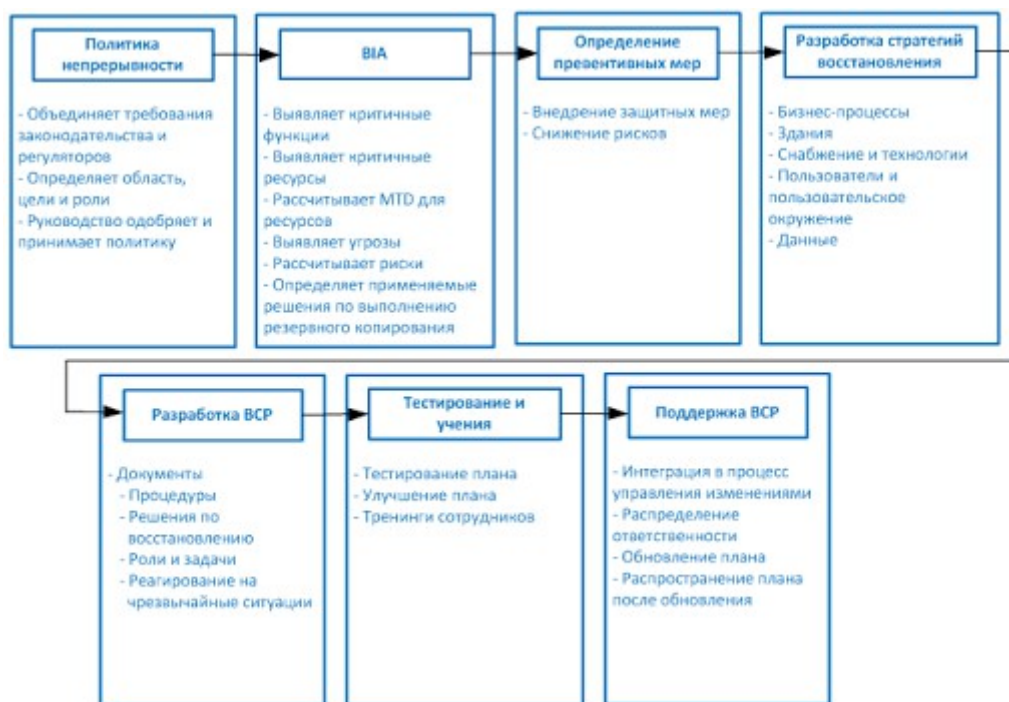


Рисунок 7-1. Составляющие процесса разработки плана непрерывности бизнеса

Хотя документ NIST SP800-34 относится к планам действий на случай чрезвычайных ситуаций в ИТ, те же шаги применимы для планирования непрерывности бизнеса в масштабах всей компании. В этом Домене рассмотрены эти шаги, позволяющие создать эффективный и полезный план BCP.

Ссылки по теме:

- Business Continuity Planning Model, Disaster Recovery Journal
- iFOSYSSEC Business Continuity and Disaster Recovery Planning resources page

1.2. BCP как часть Политики и Программы безопасности

Как было указано в Домене 01, каждая компания должна иметь политики безопасности, процедуры, стандарты и руководства. Наличие всех этих документов является частью хорошо управляемой среды, и дает преимущества с точки зрения функционирования и экономии средств. Все вместе они обеспечивают основу для программы безопасности компании. Эта программа также необходима для «выживания» компании. Поскольку в компании время от времени происходят изменения, следует обеспечить поддержание актуальности, выполнимости и эффективности программы безопасности.

Непрерывность бизнеса должна быть частью программы безопасности, должна учитываться при принятии бизнес-решений, а не просто упоминаться где-то в стороне. При ее надлежащей интеграции с процессами управления изменениями, обеспечивается возможность ее постоянной актуализации и совершенствования. Непрерывность бизнеса является фундаментальной частью эффективной программы безопасности, поэтому очень важно убедиться в их соответствии друг другу.

Очень важным вопросом при первичной разработке плана BCP является вопрос – зачем он разрабатывается. Может показаться, что ответ на этот вопрос очевиден, но это не всегда так. Можно предположить, что причиной является потребность в плане на случай непредвиденной ситуации, позволяющем быстро и безопасно обеспечить возможность продолжения персоналом выполнения своих задач, но истинные причины часто являются несколько иными. Зачем работает большинство компаний? Чтобы зарабатывать деньги и получать прибыль. А раз это обычно является основной целью бизнеса, значит BCP должна

разрабатываться для помощи компании в получении прибыли и поддержании необходимых для этого процессов. Основной причиной разработки плана ВСП является снижение рисков финансовых потерь посредством реализации способности компании быстро восстановиться и продолжить работу. Целью является минимизация влияния аварий и чрезвычайных ситуаций.

Не все компании работают для получения прибыли. Например, правительственные и военные службы, некоммерческие организации и т.п., существуют для обеспечения защиты или предоставления услуг государству или обществу. Таким образом, одни компании должны создавать свои планы ВСП для обеспечения непрерывного получения доходов, чтобы остаться на рынке, другие компании должны создавать ВСП для обеспечения выполнения возложенных на них важных задач. Хотя сферы внимания и бизнес-драйверы таких компаний могут отличаться, их планы ВСП часто имеют похожую структуру, обеспечивающую возможность восстановления работы их критичных процессов.

Обеспечить защиту самого важного для компании гораздо сложнее, если это «самое важное» не было предварительно идентифицировано. К этой задаче обычно привлекается высшее руководство компании, поскольку оно видит далеко вперед, дальше любого функционального руководителя, сосредоточенного на той области, за которую он отвечает. Бизнес-план компании обычно определяет миссию компании и критичные бизнес-функции. Для этих функций должны быть установлены приоритеты, указывающие, какие из них более критичны для выживания компании.

Для многих компаний наиболее критичными являются финансовые операции. Например, на автомобильную компанию значительно больше влияния окажет остановка на день работы ее финансовых служб, чем остановка на день ее сборочной линии. Для других компаний наиболее критичной областью может быть, например, клиентские службы.

При планировании действий в чрезвычайных ситуациях, учитываются предполагаемые и прогнозируемые проблемы. Однако может возникнуть и множество других проблем, которые не были учтены в этом плане, поэтому одним из наиболее важных свойств плана является его гибкость. План реализует систематический подход, предоставляя список действий, которые должны быть выполнены в случае чрезвычайной ситуации. План направлен на повышение эффективности и результативности выполнения необходимых работ в опасной ситуации.

Самым критичным аспектом при создании и поддержке актуальности плана непрерывности, является поддержка руководства. Руководство должно быть уверено в необходимости наличия этого плана. Должна существовать потребность бизнеса в получении и поддержке этого плана. Потребность бизнеса может быть вызвана наличием уязвимостей, требованиями регуляторов и законодательства, текущим состоянием планов восстановления и рекомендациями. Руководство сосредоточено главным образом на вопросах доходов и расходов, поэтому должно быть сделаны предварительные оценки и рассчитаны потенциальные потери. Решение о том, как компании следует восстанавливаться – это полностью бизнес-решение, и это всегда должно быть только так.

1.3. Инициирование проекта

Сначала давайте разберемся с этапом инициирования проекта. На этом этапе компании нужно понять, что она делает и зачем.

После получения надлежащей поддержки руководства, должен быть определен **координатор по непрерывности бизнеса** (business continuity coordinator). Это руководитель команды ВСП, он будет руководить разработкой, тестированием и внедрением планов непрерывности и восстановления после аварий. Желательно, чтобы этот человек обладал хорошими коммуникативными и «политическими» навыками, а также имел достаточно времени для выполнения своих обязанностей в рамках этой роли, поскольку он должен будет

координировать деятельность множества различных подразделений. Этот человек должен иметь возможность напрямую обращаться к руководству, уметь убеждать и иметь полномочия для выполнения своих руководящих задач.

Руководителю нужна команда, для этого следует организовать Комитет по ВСР. Руководство и координатор должны совместно отобрать квалифицированных людей для работы в этом Комитете. Команда должна быть составлена из людей, находящихся в хороших отношениях с различными подразделениями компании, поскольку каждое подразделение имеет уникальные функции, а также свои особые риски и угрозы. Желательно все проблемы и угрозы проработать, обсудить с заинтересованными лицами и свести в одну таблицу. Представители от каждого подразделения должны быть привлечены к этой работе не только на этапе планирования, но и на этапах тестирования и реализации. Также следует обеспечить, чтобы в разработке плана ВСР участвовали люди, которым предстоит выполнять его. Если необходимо, чтобы в критической ситуации люди выполнили определенные задачи, следует особо обратить на это их внимание на этапах планирования и тестирования.

Как минимум, в Комитет должны войти представители следующих подразделений:

- Бизнес-подразделения
- Высшее руководство
- Департамент ИТ
- Департамент безопасности
- Департамент по связям с общественностью
- Юридический департамент

После этого команда совместно с руководством должна проработать конечные цели плана, определить критичные для бизнеса аспекты, которые должны быть учтены в первую очередь при возникновении чрезвычайной ситуации, а также определить приоритеты подразделений и задач. Руководство должно напрямую оказывать помощь команде при определении границ проекта и конкретизации его целей. Казалось бы, что границы и цели проекта очевидны – нужно защитить компанию. Но на самом деле все не так просто. Предполагается, что команда должна разработать план ВСР для одного офиса компании или для всех? Должны ли быть учтены широкомасштабные угрозы, такие, как ураганы, торнадо, наводнения, террористические акты, либо достаточно будет учесть только локальные проблемы, например, неработоспособность коммуникационного канала, нарушение электроснабжения, отсутствие интернет-соединения? Каков профиль угроз для компании? Ведь если границы проекта не определены надлежащим образом, как вы узнаете, что проект завершен?

ПРИМЕЧАНИЕ. Большинство компаний включает в область проекта по разработке плана ВСР только широкомасштабные угрозы. Небольшие локальные угрозы покрываются независимыми планами действий на случай аварий и чрезвычайных ситуаций на уровне отдельных подразделений.

На этом этапе, команда совместно с руководством должна разработать *политику планирования непрерывности бизнеса* (continuity planning policy statement). Эта политика должна определить границы проекта ВСР, роли членов команды и цели проекта. Обычно этот документ описывает, что должно быть сделано после переговоров команды с руководством и достижения соглашения по срокам проекта. Этот документ должен быть одобрен руководством, чтобы гарантировать отсутствие предположений и недомолвок, принятие достигнутых соглашений всеми участниками.

Затем координатору проекта ВСР нужно воспользоваться несколькими хорошими, проверенными навыками управления проектами (см. Таблицу 7-1). При разработке плана проекта, должны быть учтены следующие вопросы:

- Связь целей с задачами
- Связь ресурсов с задачами
- Промежуточные итоги
- Сметы затрат
- Факторы успеха
- Сроки

Задача	Дата начала	Дата завершения (требуемая)	Выполнено (Ф.И.О., дата)	Принято (Ф.И.О., дата)
Инициирование проекта				
Политика непрерывности				
Анализ воздействия на бизнес				
Определение превентивных защитных мер				
Стратегии восстановления				
Разработка документов по BCP и DRP				
Тестирование планов				
Утверждение и внедрение планов				

Таблица 7-1. Шаги, которые должны быть задокументированы и приняты

После разработки плана проекта, его следует направить руководству для формального утверждения. Работа по плану должна начаться только после его утверждения. Очень важно, чтобы в этом плане не было предположений, и чтобы координатору было предоставлено официальное разрешение на использование необходимых для проекта ресурсов. Только после этого можно двигаться дальше.

2. Требования к планированию непрерывности бизнеса

Поддержка руководства является основным требованием для всего, что имеет столь далеко идущие последствия, как планирование непрерывности бизнеса. Крайне важно, чтобы руководство понимало, что реально угрожает компании, каковы последствия реализации этих угроз, каковы масштабы потенциального ущерба от реализации каждой угрозы. Без поддержки руководства, достаточных ресурсов, бюджета и времени, нельзя рассчитывать на хороший результат, а плохой план ВСР только создаст ложное чувство безопасности, что может быть еще хуже, чем полное отсутствие плана. Проблемы в плане ВСР обычно напрямую связаны с проблемами понимания важности этой работы руководством, неправильным видением руководством целей плана.

Руководители несут ответственность в соответствии с различными законами и постановлениями. Они могут быть привлечены к суду акционерами и клиентами, если не обеспечивают должную заботу (due care) и осмотрительность (due diligence), и не выполняют свои обязанности в отношении восстановления после аварий и обеспечения непрерывности бизнеса. В некоторых отраслях к компаниям предъявляются жесткие требования регуляторов и законодательства, которым они обязаны следовать. Эти требования должны быть проанализированы и с самого начала предусмотрены в плане. Например, банки и инвестиционные компании должны гарантировать, что даже при возникновении чрезвычайной ситуации к конфиденциальной информации их клиентов не получают доступ неуполномоченные лица, она не будет искажена и т.п. Работы по планированию восстановления после аварий и обеспечению непрерывности бизнеса выполняются лучше всего при использовании подхода «сверху вниз», а не наоборот. Именно руководство должно управлять этим проектом, а не рядовой персонал компании.

Многие компании стремятся к максимально быстрому развитию и продвижению, и

отказываются от траты времени и ресурсов на вопросы непрерывности и восстановления, не видя в этом возможностей для мгновенного получения выгоды или повышения рыночной доли. Специалистам, работающим в таких компаниях и понимающим ценность результатов этой работы, предстоит пройти через трудный путь убеждения топ-менеджмента компании в целесообразности выполнения этих работ. Но когда произойдет авария, они не пожалеют о потраченных на подготовку усилиях, поскольку результат будет просто неоценим. Сегодняшнему деловому миру требуется два важных аспекта: побуждение для выпуска на рынок прекрасной продукции или услуг, а также мудрость, чтобы предусмотреть возможность возникновения непредвиденных проблем.

Важно, чтобы руководство установило набор высокоуровневых целей для планирования непрерывности и помогло расставить приоритеты, указав, что должно быть сделано в первую очередь. После того, как руководство установит цели, политики и расставит приоритеты, члены команды проекта ВСР могут приступать к своей работе. Однако нужно понимать, что потребность в поддержке руководства не заканчивается на этом. Руководство должно убедиться, что планы и процедуры реально разработаны и внедрены. Руководство должно быть уверено, что планы поддерживаются в актуальном состоянии и отражают реальные приоритеты компании, несмотря на происходящие в ней изменения.

2.1. Анализ воздействия на бизнес

Планирование непрерывности бизнеса связано с неизвестностью и вероятностью. Конечно, вы не можете предсказать, где и когда случится чрезвычайная ситуация, но это не значит, что вам не нужно готовить план на этот случай. То, что мы не знаем, что завтра в 10 утра произойдет землетрясение, не значит, что мы не должны заранее запланировать действия, которые необходимо предпринять, чтобы выжить при землетрясении (или другой аварии, чрезвычайной ситуации). Для создания плана непрерывности бизнеса нужно попытаться продумать все возможные аварии и чрезвычайные ситуации, которые могут произойти, оценить их потенциальные последствия и ущерб от них, категоризировать и приоритезировать их, разработать реальные альтернативы на случай, если такая авария или чрезвычайная ситуация действительно произойдет.

BIA (business impact analysis – анализ влияния на бизнес) – это функциональный анализ, в процессе которого команда собирает данные, проводя интервью и анализируя документальные источники; документирует бизнес-функции, действия и транзакции; разрабатывает иерархию бизнес-функций; и, наконец, определяет уровень критичности каждой отдельной функции с помощью порядка классификации. Но как определить этот порядок классификации, основанный на уровнях критичности? Комитет ВСР должен идентифицировать угрозы, перед лицом которых стоит компания, и определить для них следующие характеристики:

- Максимально допустимое время простоя
- Нарушение работы и снижение производительности
- Возмещение финансовых убытков (financial consideration)
- Ответственность, установленная законодательством и регуляторами
- Репутация

Сам Комитет, конечно же, не может полностью понимать все бизнес-процессы компании, какие действия и ресурсы нужны для работы этих процессов. Поэтому Комитет должен собрать эту информацию от людей, которые ей обладают – руководителей подразделений компании и отдельных сотрудников. Сначала Комитет должен определить, кто их сотрудников будет участвовать в процессе сбора данных для BIA. Комитет должен решить, каким образом он будет собирать данные у выбранных сотрудников – будет ли это анкетирование, интервью или рабочие совещания. Затем команда должна собрать

необходимую для проведения ВИА информацию, проведя анкетирование, интервью и совещания с wybranymi сотрудниками. Полученные при этом данные будут использоваться для последующего анализа. Очень важно, чтобы члены команды спросили, как выполняются различные функции (процессы, транзакции, сервисы) в компании. Должны быть оформлены схемы процессов, которые будут использоваться на этапе проведения ВИА и на других этапах разработки плана.

После завершения этапа сбора данных, Комитет ВСР должен провести их анализ для определения, какие процессы, устройства и операционные функции являются критичными для компании. Например, если система является автономной, не влияет на другие системы и имеет невысокую критичность, она может быть классифицирована, как восстанавливаемая на второй или третьей фазе процесса восстановления. Соответственно, эта система не будет восстанавливаться до тех пор, пока не завершится восстановление и запуск в работу более критичных систем и ресурсов. Этот анализ может быть проведен с использованием стандартной методологии оценки и анализа рисков (процесс анализа рисков мы обсуждали в Домене 01).

Угрозы могут быть рукотворными, природными или техническими. Рукотворными угрозами могут быть поджоги, террористические акты или просто ошибки, приводящие к серьезным последствиям. Природными угрозами могут быть торнадо, наводнения, ураганы или землетрясения. Техническими угрозами могут быть повреждения данных, перебои электроснабжения, неисправности устройств, нарушения работы коммуникационных каналов. Важно выявить все возможные угрозы и рассчитать вероятность их реализации. Некоторые угрозы могут показаться неразумными и надуманными, например, злоумышленные действия сотрудников, вандализм, протесты сотрудников, атаки хакеров и т.п., но эти угрозы также должны быть идентифицированы. Для анализа угроз лучше использовать подход, основанный на сценариях их реализации. Это позволит наиболее полно учесть при подготовке плана влияние этих угроз на задачи бизнеса, подразделения и критичные операции компании. Чем больше угроз учтет компания при планировании, тем лучше она будет готова к наступлению чрезвычайных ситуаций.

Шаги ВИА. Отдельные шаги ВИА перечислены ниже:

1. Подбор людей для опроса в процессе сбора данных.
 2. Выбор метода сбора данных (анкетирование, исследование, количественный и качественный подходы).
 3. Идентификация критичных для компании бизнес-функций.
 4. Идентификация ресурсов, которые необходимы для выполнения этих функций.
 5. Расчет времени, в течение которого эти бизнес-функции могут «прожить» без этих ресурсов.
 6. Выявление уязвимостей и угроз для этих бизнес-функций.
 7. Расчет рисков для каждой отдельной бизнес-функции.
 8. Подготовка документа по результатам ВИА и предоставление его руководству.
- В этом Домене мы рассмотрим каждый из этих шагов.

Комитет должен проанализировать сценарии, которые приведут к следующим последствиям:

- Неисправность или недоступность оборудования
- Недоступность водоснабжения, коммунальных услуг, отопления, вентиляции, электроэнергии (utilities)
- Недоступность (невозможность использования) здания
- Недоступность (неработоспособность) критичного персонала
- Недоступность (неработоспособность) производителей или поставщиков услуг
- Повреждение программного обеспечения и/или данных

Следующим шагом анализа рисков является определение ценности активов, на которые может оказать влияние каждая из угроз. Это поможет сделать план экономически

целесообразным. Как уже говорилось в Домене 01, определение ценности активов не так однозначно, как может показаться. Ценность актива – это не просто сумма денег, которую за него заплатили. Должна быть учтена роль (значение) этого актива для компании, а также трудозатраты на его воссоздание (например, если это часть программного обеспечения). Кроме того, при определении ценности актива должны учитываться вопросы ответственности компании в случае повреждения актива или нарушения его безопасности. (Более подробно вопросы определения ценности активов рассмотрены в Домене 01).

Должна быть собрана количественная и качественная информация в отношении воздействия на бизнес фактов реализации различных угроз, собранная информация должна быть надлежащим образом проанализирована и интерпретирована с целью максимально точного понимания последствий воздействия этих угроз. При этом воздействие может быть экономическим, функциональным (операционным) или одновременно и тем, и другим. Результаты анализа следует передать наиболее квалифицированным людям в компании для рассмотрения, чтобы они подтвердили, что получены адекватные результаты, описывающие реальные риски, перед лицом которых стоит компания, и их воздействия на бизнес. Это поможет сразу исключить все несущественные данные и обеспечить максимально полное понимание всех возможных воздействий на бизнес.

К выявленным угрозам по отдельности должны быть применены критерии потерь (loss criteria). Среди этих критериев могут быть следующие:

- Ущерб репутации и общественному доверию
- Потеря конкурентных преимуществ
- Повышение операционных расходов
- Нарушение контрактных обязательств
- Нарушение законодательства и требований регуляторов
- Отложенные расходы
- Потеря доходов
- Снижение производительности

Эти прямые и косвенные потери должны быть надлежащим образом учтены.

Например, если команда ВСР рассматривает угрозу террористического акта, важно определить, на какие бизнес-функции это скорее всего будет направлено, на какие бизнес-функции будет оказано воздействие, насколько применим для этого случая (прямо или косвенно) каждый из пунктов перечня критериев потерь. Для отдельных бизнес-процессов и компании в целом может быть критично время восстановления. Например, это может быть применимо к деятельности по поддержке клиентов, которая может быть остановлена не более, чем на два дня. Если поддержка клиентов будет остановлена на пять дней, компания может прекратить свое существование.

После идентификации критичных функций, необходимо точно определить, что требуется для работы отдельным бизнес-процессам, реализующим эти критичные функции. Среди необходимых им ресурсов не обязательно будут только компьютерные системы, они могут требовать наличия определенного персонала, выполнения определенных процедур и задач, наличия запасов сырья, поддержки производителей. Команда должна понять, какое влияние на эти процессы окажет недоступность одного или более из необходимых им ресурсов, отсутствие каких ресурсов полностью остановит выполнение соответствующей критичной функции.

В процессе ВИА, команда идентифицирует критичные для компании системы, необходимые для ее функционирования, и рассчитывает допустимое для компании время их простоя,

вызванного различными неблагоприятными событиями. Это время называется **максимально допустимым временем простоя** (MTD – maximum tolerable downtime).

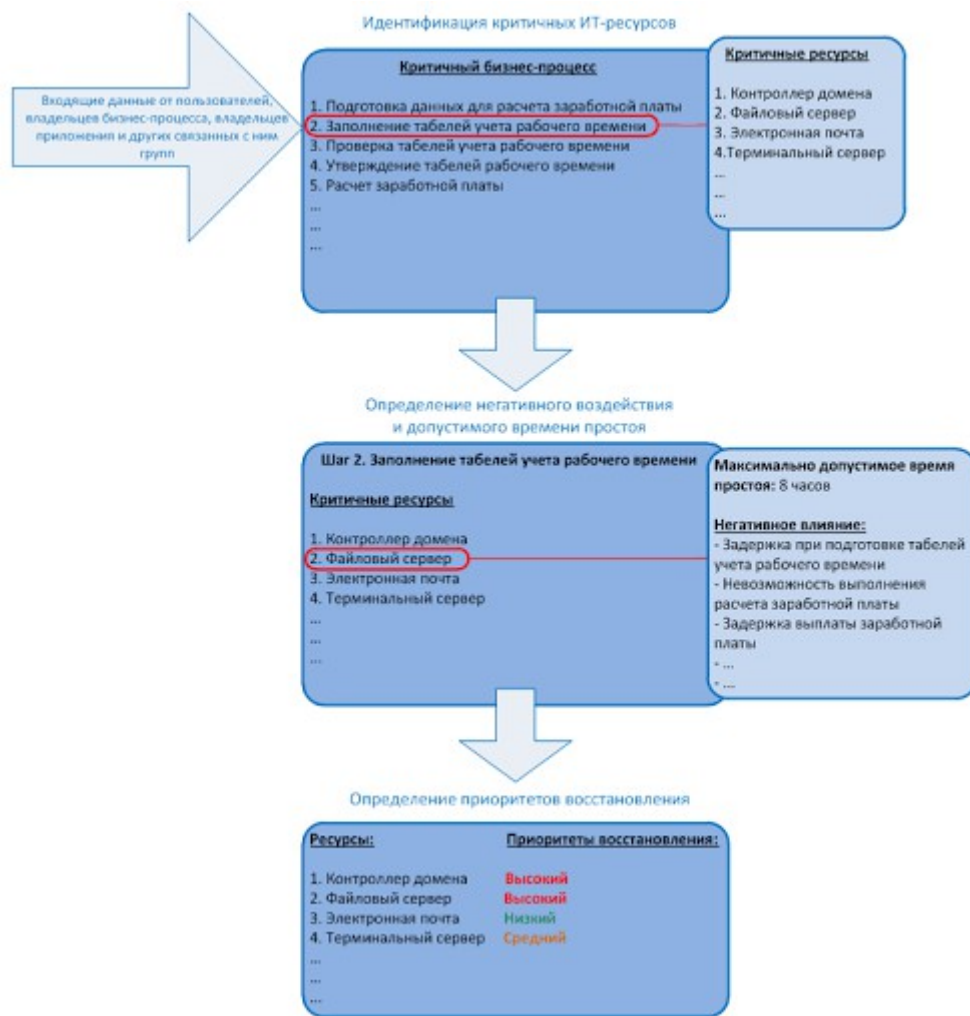
Ниже представлены некоторые значения MTD, которые могут быть использованы компанией:

- **Второстепенная система.** MTD = 30 дней
- **Обычная система.** MTD = 7 дней
- **Важная система.** MTD = 72 часа
- **Необходимая система.** MTD = 24 часа
- **Критичная система.** MTD = менее 1 часа

Каждая бизнес-функция и каждый актив должны быть помещены в одну из этих категорий, в зависимости от времени, в течение которого компания может жить без этой функции или актива. Это поможет компании обоснованно выбрать необходимое ей решение для организации резервного копирования, обеспечивающее гарантии доступности этих ресурсов. Например, если неработоспособность выделенной линии в течение 3 часов обойдется компании в \$130 000, эту выделенную линию следует считать критичной, и компании следует организовать резервный канал связи от другого провайдера. Если недоступность сервера в течение 10 дней нанесет компании ущерб всего на \$250, его следует считать второстепенным, и компании не нужен полностью готовый к работе резервный сервер, который можно будет сразу же включить в работу при неисправности основного сервера. Вместо этого, компания может решить заключить соглашение со специализированной компанией (аутсорсинг), который, в соответствии с согласованным SLA (service level agreement - соглашение об уровне услуг), восстановит работу этого сервера за 8 дней.

Команда ВСР должна попытаться продумать максимально возможное количество событий, которые могут произойти и нанести ущерб компании. При этом команда ВСР должна понимать, что обеспечить защиту от всех возможных событий и сценариев – нереально. Хорошая подготовка к наводнению, землетрясению, террористической атаке или удару молнии не так важна, как хорошая подготовка к нарушению или полной остановке критичных бизнес-функции. Компания должна быть готова к повреждению любого (или всех) из своих бизнес-ресурсов, а не сосредотачиваться на отдельных событиях, которые могут нанести ей ущерб.

ПРИМЕЧАНИЕ. BIA выполняется в самом начале процесса планирования непрерывности бизнеса. Проведение BIA необходимо для выявления областей, которые будут подвержены наибольшему ущербу в случае аварии или разрушения (disruption). В процессе BIA выявляются критичные для компании системы, необходимые для ее существования, проводится оценка максимально допустимого для компании времени простоя, вызванного аварией или разрушением.



Взаимозависимости

Важно рассматривать компанию в виде сложного организма. В компании используется множество различных типов оборудования, работают люди, выполняются различные задачи, созданы различные подразделения, организованы коммуникационные каналы и интерфейсы связи с внешним миром. Самой большой сложностью при правильном планировании непрерывности является понимание всех этих компонентов и их взаимоотношений. Команда может разработать планы резервного копирования и восстановления данных, установить резервное оборудование, обучить сотрудников вручную выполнять автоматизированные задачи и предусмотреть дополнительные источники питания. Но если нет четкого понимания, как все эти компоненты будут совместно работать в другой среде, позволяя получить на выходе тот же продукт, это может оказаться пустой тратой времени и денег.

Следующие взаимосвязи и взаимозависимости задач должны быть выявлены командой ВСР и учтены в разрабатываемом плане непрерывности:

- Определение важнейших бизнес-функций и поддерживающих их подразделений.
- Определение взаимозависимостей между этими функциями и подразделениями.
- Выявление всех возможных нарушений (аварий, разрушений) которые могут оказать воздействие на механизмы, необходимые этим подразделениям для функционирования.
- Выявление и документирование потенциальных угроз, которые могут нарушить взаимодействие подразделений между собой.
- Сбор количественной и качественной информации, относящейся к этим угрозам.

- Обеспечение альтернативных методов восстановления функциональности и коммуникаций.
- Подготовка краткого и понятного описания для каждой угрозы и соответствующей информации.

Основной целью планирования непрерывности бизнеса является возможность восстановить работу компании настолько быстро, насколько это возможно, потратив при этом минимум денег. Полный план возобновления работы компании должен охватывать все организационные единицы компании, определять критичные сервисы и функции, обеспечивать альтернативный режим работы в аварийном режиме и включать в себя планы каждого подразделения. Эта работа может быть выполнена собственным персоналом компании или внешними консультантами, либо с участием и тех, и других. Последний вариант может иметь ряд преимуществ для компании, поскольку консультанты являются экспертами в этой области, хорошо знают, какие шаги нужно выполнить и в какой последовательности, знают, какие вопросы нужно задать, какие могут возникнуть проблемы, на какие из них следует обратить повышенное внимание. Также, внешние консультанты могут дать разумные советы. При этом внутренние сотрудники компании гораздо лучше знают свою компанию и лучше понимают, как те или иные угрозы могут воздействовать на работу отдельных бизнес-функций компании и работу компании в целом. Чаще всего одновременное участие в подготовке плана ВСП собственного персонала компании и внешних консультантов является наиболее правильным выбором и позволяет эффективно решить все необходимые вопросы.

В масштабах предприятия. Определение и согласование границ проекта ВСП даст ответ на вопрос, нужно ли включать в план только один офис или несколько (все) офисов компании. Большинство планов ВСП разрабатывается в масштабах компании в целом, а не для отдельных ее частей. В большой компании может иметь смысл разработать отдельные планы действий на случай чрезвычайных ситуаций для каждого подразделения. План подразделения будет учитывать их специфические потребности в процессе восстановления. Эти планы отдельных подразделений не должны противоречить общему плану ВСП в масштабах всей компании.

Итак, к настоящему моменту мы определили следующие обязанности руководства:

- Принятие решения о разработке ВСП и обеспечение поддержки
- Установка политики и целей
- Предоставление необходимого бюджета и ресурсов
- Принятие на себя ответственности за результаты разработки ВСП
- Создание команды для разработки ВСП

Обязанности команды ВСП заключаются в следующем:

- Определение требований законодательства и регуляторов, которые должны быть учтены
- Выявление всех возможных уязвимостей и угроз
- Оценка вероятности реализации этих угроз и потенциального ущерба от их реализации
- Выполнение BIA
- Определение, работа каких подразделений, систем и процессов должна быть восстановлена раньше других
- Разработка процедур и шагов для возобновления работы бизнес-процессов компании после аварии

Существует несколько программных продуктов, которые могут помочь и упростить процесс разработки плана ВСР. Автоматизация отдельных процедур может уменьшить сроки реализации этого проекта, упростить сбор больших объемов информации. Многие необходимые элементы уже реализованы в этих программных продуктах и предоставляются в виде шаблонов.

Полученная в результате ВИА информация вместе с другими данными, рассмотренными в предыдущих разделах, должна быть предоставлена высшему руководству. Руководство обычно требует предоставлять информацию, выраженную в деньгах или иных количественных терминах, а не в виде субъективных качественных понятий. Конечно, неплохо знать, что в случае торнадо все будет *"совсем плохо"*, но гораздо лучше знать, что в случае торнадо пострадает 65% помещений компании, возникнет риск остановки всех компьютерных систем на срок до 72 часов, прекращения электроснабжения на срок до 24 часов, а работа компании полностью остановится на 76 часов, при этом ежедневно компания будет нести потери, эквивалентные \$125 000. Работать с количественной информацией значительно проще и эффективнее.

Важно, чтобы все это было сделано до того, как команда ВСР уже разработает какой-либо из своих планов. Сначала нужно собрать данные, провести их анализ, а результаты представить руководству. Руководство должно проанализировать эти результаты и одобрить их, после чего команда должна приступить к разработке плана.

Будем считать, что руководство нас горячо поддержало и теперь мы имеем возможность перейти к следующим этапам.

Ссылки по теме:

- Business Continuity Planning & Disaster Recovery Planning Directory, "Business Impact Analysis," Disaster Recovery World
- Business Continuity Institute (BCI)
- DRI International (DRII)

2.2. Превентивные меры

В процессе выполнения ВИА команда ВСР должна определить максимально допустимое время простоя (MTD - Maximum tolerable downtime) для критичных ресурсов. Это необходимо для понимания негативного воздействия на бизнес, вызванного недоступностью активов. Также это создает чувство, что команда могла бы попытаться снизить это воздействие и, тем самым, снизить соответствующий риск, внедрив превентивные меры. Не внедрить после этого превентивные меры - это аналогично походу к доктору и последующему игнорированию его рекомендаций. Зачем тогда вообще нужно было ходить к доктору? Тоже самое справедливо и для компаний. Если команда выявила риск и имеет решение по его минимизации, но компания не внедряет это решение, зачем тогда была организована эта команда?

Таким образом, вместо того, чтобы просто ждать очередную аварию или чрезвычайную ситуацию, чтобы увидеть, как компания справится с ней, следует внедрить контрмеры, обеспечивающие лучшую защиту компании от выявленных вероятных негативных воздействий. Реализация приемлемых и экономически эффективных превентивных мер называется проактивным подходом, он гораздо предпочтительнее реактивного подхода. Выбор превентивных механизмов, которые следует внедрить, зависит от результатов ВИА, но среди них могут быть некоторые из следующих компонентов:

- Укрепление здания, его конструкционных материалов
- Использование резервных серверов и коммуникационных каналов

- Ввод электропитания от разных трансформаторов
- Поддержка дополнительными (избыточными) поставщиками
- Страхование
- Установка источников бесперебойного питания и электрогенераторов
- Внедрение технологий резервного копирования данных
- Обеспечение защиты носителей информации
- Увеличение количества запасных частей для критичного оборудования
- Внедрение систем обнаружения и тушения пожара

ПРИМЕЧАНИЕ. Многие из защитных мер, обсуждаемых в этом Домене, более подробно рассмотрены в Доменах 04 и 10.

2.3. Стратегии восстановления

Итак, команда ВСР выполнила этап инициирования проекта, получила поддержку руководства, необходимые ресурсы, определила границы проекта, назначила членов команды ВСР. Также завершен этап ВИА, т.е. Комитет провел оценку и анализ рисков, сформировал на основе полученных результатов отчет о реальном уровне риска, перед лицом которого стоит компания.

У Комитета ВСР уже есть схема, отображающая работу компании в целом, которая была разработана на этапе ВИА. Работа компании глубоко проанализирована, определены критичные функции, которые обязательно должны выполняться постоянно, чтобы компания могла продолжать работу. Идентифицированы ресурсы, которые требуются этим функциям, рассчитаны значения MTD для отдельных ресурсов и самих функций. Если все это действительно сделано, можно считать этап ВИА выполненным. Все работы, которые были сделаны до этого момента, относятся к фазе "оценки рисков" процесса разработки плана ВСР. Они были направлены на то, чтобы определить, насколько плохо будет компании в различных аварийных сценариях.

На этапе разработки стратегий восстановления, команда анализирует полученную ранее информацию с различных точек зрения. Теперь нужно определить, что необходимо компании, чтобы реально восстановить все важные для нее компоненты. ВИА предоставляет исходную информацию для разработки стратегий восстановления каждого из компонентов. Бизнес-процессы компании всецело зависят от правильной реализации этих стратегий восстановления.

На данный момент, результаты ВИА предоставлены руководству, руководство распределило ресурсы, необходимые для перехода к следующим этапам. Теперь Комитет ВСР должен разработать самые эффективные (в том числе с экономической точки зрения) механизмы восстановления, которые будут являться ответом на угрозы, выявленные на этапе ВИА. Как вы помните, на этапе ВИА команда рассчитывает потенциальный ущерб от каждой выявленной угрозы. (Например, если недоступен офис компании, ее потери составляют \$200 000 в день, если не работает соединение с сетью Интернет, компания несет убытки в размере \$12 000 в час и т.д.). При выборе возможных решений по восстановлению, команда должна учитывать эти значения для анализа затрат и получаемых выгод в отношении каждого возможного решения, направленного на снижение уровня рисков компании.

На текущем этапе команда должна разработать стратегии восстановления, являющиеся набором заранее спланированных действий, которые необходимо будет выполнить в случае аварии или чрезвычайной ситуации. Звучит достаточно просто, но в действительности на этом этапе предстоит выполнить большой объем работы, сравнимый с проведением ВИА.

В процессе ВИА команда рассчитывает необходимое *время восстановления* (recovery times),

которое должно соблюдаться при восстановлении различных критичных бизнес-функций и ресурсов, от которых зависят эти функции. Предположим, команда рассчитала, что компания будет терять \$200 000 упущенной выгоды в день, если здание компании будет уничтожено или непригодно для работы. Теперь команда знает, что у компании есть около 5-6 часов на решение этой проблемы, иначе ей будет нанесен огромный финансовый ущерб. Это может означать, что компании нужна готовая к работе "горячая" площадка (hot site) или дополнительное здание (redundant facility). Это позволит компании восстановить работу за указанное время.

В чем разница между превентивными мерами и стратегиями восстановления? Превентивные механизмы внедряются для того, чтобы попытаться снизить вероятность аварии, а если авария все же произойдет, минимизировать потери от нее. Конечно, компания не может остановить торнадо, но она может перенести свой офис в другое здание, не находящееся наallee торнадо в Канзасе. Компания не может остановить машину, которая в следующую секунду врежется в трансформаторную подстанцию, но она может заранее организовать свое электроснабжение с нескольких независимых подстанций.

Стратегии восстановления – это процессы, направленные на спасение компании после аварии. В состав этих процессов входят такие механизмы, как создание альтернативной площадки, внедрение процедур реагирования на чрезвычайные ситуации, активацию ранее внедренных превентивных механизмов.

После того, как команда определила сроки восстановления для отдельных бизнес-функций, операций и ресурсов, она должна определить механизмы и стратегии восстановления, необходимые для гарантированного восстановления и возобновления их работы в рамках рассчитанных сроков. Команде следует разбить эти стратегии восстановления на следующие разделы:

- Восстановление бизнес-процессов
- Восстановление здания
- Восстановление технической среды
- Восстановление пользовательской среды
- Восстановление данных

2.4. Восстановление бизнес-процессов

Бизнес-процесс – это набор взаимосвязанных шагов, направленных на выполнение определенной задачи. У бизнес-процесса есть точка начала и точка окончания, бизнес-процесс является повторяемым. Эти процессы содержат сведения о сервисах, ресурсах и операциях, предоставляемых компанией. Например, если клиент заказывает автомобиль через интернет-сайт автомобильной компании, компания должна выполнить следующий набор шагов:

1. Проверить наличие автомобиля.
2. Проверить местонахождения автомобиля, определить сроки его доставки.
3. Сообщить клиенту стоимость и дату доставки.
4. Принять информацию банковской карты клиента.
5. Проверить и обработать платеж по банковской карте.
6. Отправить клиенту чек и номер заказа для контроля его состояния.
7. Отправить запрос на доставку по месту нахождения автомобиля.
8. Получить автомобиль.
9. Отправить счет в бухгалтерию.

Команда ВСР должна хорошо понимать эти шаги, выполняемые в рамках критичных для компании процессов. Эти сведения обычно оформляются в виде карты рабочего процесса (workflow document), в котором указаны роли и ресурсы, необходимые каждому процессу. Команда ВСР должна обладать следующей информацией о критичных бизнес-процессах:

- Необходимые роли
- Необходимые ресурсы
- Механизмы на входе и выходе
- Шаги рабочего процесса (workflow steps)
- Требуемое время на выполнение
- Взаимодействие с другими процессами

Эта информация позволит команде выявить угрозы и выбрать соответствующие защитные меры, необходимые для минимизации негативного воздействия в случае прерывания процесса.

2.5. Восстановление здания

Разрушения (disruption) могут быть трех основных типов: некритичные, критичные и катастрофические. *Некритичные* (nondisasters) – это нарушения работы отдельных сервисов, например, неисправности или сбои в работе оборудования. Решение по восстановлению может заключаться в восстановлении аппаратного и программного обеспечения или файлов данных. *Критичные* (disasters) – это события, приводящие к невозможности использования здания на день или более. Обычно это требует использования альтернативного здания для работы, восстановления программного обеспечения и данных с резервных копий, хранящихся вне основного здания (offsite copies). Альтернативное здание должно быть доступно компании все время, пока основное здание не будет отремонтировано и готово к использованию для работы компании. *Катастрофические* (catastrophe) – это основательные разрушения, не позволяющие рассчитывать на восстановление здания. Это требует наличия как быстрого временного решения, которым может быть альтернативная внешняя площадка, так и постоянного, но длительного решения, которым может являться постройка нового основного здания.

Аварии и катастрофы редко можно сравнить с некритичными ситуациями. Некритичные ситуации обычно могут быть решены простой заменой устройства или восстановлением файлов с резервной копии, хранящейся в том же здании (onsite backup). Команда ВСР должна продумать требования к внутреннему хранению резервных копий и принять осознанное решение. Команда должна определить, какое оборудование является критичным, рассчитать, на основе имеющейся статистики, среднее время между сбоями (MTBF – mean time between failures, наработка на отказ) и среднее время его ремонта (MTTR – mean time to repair), чтобы понять, когда потребуется ремонт устройства и когда оно полностью выйдет из строя, и потребуется его замена.

ПРИМЕЧАНИЕ. MTBF – это оценочное время жизни устройства, рассчитанное его производителем или третьей стороной. Значение MTBF нужно, чтобы знать примерные сроки, когда потребуется замена этого устройства. MTTR – это оценочное время ремонта устройства и его возвращения в работу. Эти концепции более подробно будут рассмотрены позднее в Домене 10.

Для более широкомасштабных аварий, воздействующих на основное здание, должно быть предусмотрено и доступно альтернативное здание. Обычно компании заключают договор с внешним поставщиком на получение такой услуги (альтернативного здания). Клиент платит ежемесячную абонентскую плату за возможность воспользоваться альтернативным зданием при необходимости, а когда такая необходимость возникает, оплачивает его использование и в кратчайшие сроки поставщик предоставляет его в распоряжение компании. Оплата за

использование альтернативного здания может быть посуточной, либо почасовой. Использовать альтернативное здание достаточно дорого, поэтому такую услугу следует рассматривать как кратковременное решение.

Важно отметить, что большинство договоров на предоставление альтернативного здания не обещают предоставить компании какое-то конкретное здание или здание в конкретном месте, обычно они обещают предоставить компании здание в определенном районе, где расположено основное здание компании (или в другом районе, указанном в договоре). После катастрофы 11 сентября 2001 года, многие компании, имевшие офисы в Манхеттене, были очень удивлены, когда их поставщики предоставили им альтернативные офисы не в Нью-Джерси (где все было уже занято), а в Бостоне, Чикаго или Атланте. Это приводит к дополнительным сложностям для процесса восстановления, так как возникают вопросы логистики, перевозки людей и оборудования в незапланированное изначально место.

Существует три основных типа альтернативных зданий (офисов), которые компания может арендовать:

- **«Горячая» площадка (Hot site).** Арендованное здание (офис), в котором все предварительно установлено, настроено и готово к работе. Чтобы начать работать в этом здании (офисе), компании нужно всего несколько часов. Там нет только некоторых ресурсов, актуальных данных, которые потребуются восстановить с резервных копий, и людей, которые будут обрабатывать эти данные. Установленное на этой площадке оборудование и системное программное обеспечение должно быть полностью совместимо с данными, которые будут восстановлены с резервных копий, должны быть исключены проблемы взаимодействия систем. Такие площадки являются хорошим выбором для компаний, которым нужны гарантии доступности площадки в любой момент, когда она потребуется, и гарантии восстановления своей работы в кратчайшие сроки. В большинстве случаев, для поддержки «горячих» площадок компанией выполняется ежегодное тестирование, в рамках которого проверяется, что площадка на должном уровне готова к работе. Это самый дорогой из всех трех вариантов альтернативных зданий. При выборе этого варианта могут возникнуть сложности в случае, если компании для работы требуется редко используемое или дорогостоящее программное обеспечение или оборудование.

ПРИМЕЧАНИЕ. Поставщик «горячей» площадки обычно предоставляет стандартное, наиболее часто используемое аппаратное и программное обеспечение, которое удовлетворит основную массу клиентов. Однако в этот список скорее всего не будет входить специализированное и нестандартное программное обеспечение, которое может использоваться у клиента.

- **«Теплая» площадка (Warm site).** Арендованное здание (офис), в котором предварительно установлено и настроено только некоторое оборудование. Можно сказать, что «теплая» площадка – это обычная «горячая» площадка, на которой нет дорогого оборудования. Чрезвычайно дорого иметь в резерве постоянно готовое к использованию здание, в котором установлено и настроено все необходимое оборудование и компьютеры. Более дешевым вариантом является «теплая» площадка, являющаяся просто альтернативным зданием, в котором предварительно установлено некоторое оборудование. Это наиболее широко используемый вариант, поскольку он значительно дешевле «горячей» площадки и при этом обеспечивает возможность возобновления работы компании за не очень большое время. Также это может быть лучшим выбором для компаний, работа которых зависит от дорогостоящего или экзотического программного обеспечения или оборудования, поскольку после аварии они перевезут на эту площадку оборудование с основной площадки и установят на нем свое программное обеспечение. Вероятность нахождения поставщика, имеющего возможность предоставить постоянно готовую удаленную площадку, на которой установлен и готов к работе суперкомпьютер Cray, крайне низка. Однако существенным недостатком при выборе «теплой» площадки может быть отсутствие в

договоре с поставщиком возможности проведения ее ежегодного тестирования, при этом у компании не будет уверенности в возможности возврата к работе за определенное, приемлемое для нее, время.

- **«Холодная» площадка (Cold site).** Арендованное здание (офис), в котором есть только базовые компоненты, такие как электрическая проводка, кондиционирование воздуха, водопровод и т.п., но в нем нет никакого оборудования и дополнительных сервисов. Чтобы подготовить такую площадку к работе, могут потребоваться недели. На «холодных» площадках могут быть смонтированы стойки для оборудования, установлены столы, проложено «темное» оптоволокно (т.е. только кабель, без обеспечивающей передачу сигналов оптики и электроники), но клиенту потребуется привезти на эту площадку свое оборудование, настроить его и запустить в работу. Использование «холодной» площадки является самым дешевым вариантом, но для ее запуска в работу после аварии потребуется гораздо больше времени, чем в первых двух вариантах. «Холодные» площадки часто используются в качестве альтернативных зданий (офисов) для колл-центров и других служб, которым почти ничего перевозить не нужно, которые не требуют дорогого переоборудования и строительства.

ПРИМЕЧАНИЕ. Важно понимать, что перечисленные выше варианты площадок, предоставляются в виде услуг специализированными организациями (поставщиками). Для получения этой услуги, компания платит такой организации ежемесячную абонентскую плату. «Горячая» площадка (hot site) – это сервис по подписке. *Резервная площадка (redundant site)* – это принадлежащее самой компании здание (офис), которое компания поддерживает самостоятельно и никому не платит за него. Хотя резервная площадка также может быть «горячей», т.е. постоянно готовой к работе, нужно понимать разницу между этими понятиями: «горячая» площадка – сервис по подписке, резервная площадка – собственность компании.

Большинство компаний используют *«теплые»* площадки, на которых установлено лишь некоторое оборудование, например, дисковые и ленточные накопители, контроллеры и т.п. Не многие компании могут позволить себе «горячую» площадку, но при этом они не могут себе позволить и слишком длительные простои, поскольку это приведет к значительному ущербу. «Теплая» площадка может являться более долгосрочным решением по сравнению с «горячей» площадкой. Если компания все же решила использовать «холодную» площадку, она должна быть готова приостановить свою деятельность на 1-2 недели.

Ниже указаны ключевые различия между основными типами альтернативных площадок:

Преимущества «горячей» площадки

- Готовность к работе уже через час
- Высокая доступность
- Хотя обычно используется в качестве кратковременного решения, может быть доступна и для более длительного использования
- Существует возможность проведения ежегодного тестирования

Недостатки «горячей» площадки

- Очень дорого
- Возникают ограничения по выбору аппаратного и программного обеспечения

Преимущества «теплой» и «холодной» площадок

- Менее дорогие
- Доступны на более длительное время за меньшие деньги
- Более удобно в случае использования специализированного или дорогостоящего

программного обеспечения и оборудования

Недостатки «теплой» и «холодной» площадок

- Доступны только через некоторое время
- Нет возможности быстро начать работу на них
- Обычно недоступна возможность периодического тестирования

Третичные площадки. На этапе проведения BIA, команда может выявить существование опасности того, что альтернативная (вторичная) площадка окажется недоступной, когда в ней возникнет необходимость. Это может потребовать создания третичной (второй альтернативной) площадки, которая будет использоваться в случае недоступности основной альтернативной (вторичной) площадки. Это своего рода «резервирование резервирования». Обычно это «план Б», на случай, если «план А» не сработает.

При использовании «горячей» площадки, ленты с резервными копиями и другие резервные носители должны периодически проверяться на оборудовании этой площадки, чтобы убедиться, что оборудование «горячей» площадки может читать данные с используемых носителей. Если используется «теплая» площадка, ленты и другие носители с резервными копиями следует приносить тестировать на основной площадке. Это различие вызвано тем, что при использовании компанией «горячей» площадки, она зависит от размещенного на ней оборудования и ей необходимо убедиться, что оно может работать с используемыми резервными носителями. При использовании «теплой» площадки, компания, вероятно, перенесет на нее оборудование со своей основной площадки, поэтому и тестировать резервные носители следует именно на нем.

Размещение альтернативной площадки. Альтернативное здание следует выбирать на существенном отдалении от основного здания, чтобы одна катастрофа не затронула сразу оба здания. Другими словами, нелогично создавать альтернативную площадку в нескольких километрах от основной, поскольку при реализации таких угроз, как торнадо, наводнение и т.п., альтернативная площадка может оказаться также подвержена воздействию той же угрозы и быть разрушена. При низкой и средней критичности рекомендуется расстояние между основным и альтернативным зданием не менее 25 километров; в случае высокой критичности рекомендуется расстояние 80-320 километров, что обеспечит достаточную защиту от региональных катастроф.

Соглашение о взаимной помощи

Другим подходом к организации резервного здания (офиса) является заключение *соглашения о взаимной помощи* (reciprocal agreement) с другой компанией. В рамках такого соглашения, компания А позволяет компании Б использовать свое здание, если компания Б пострадает от катастрофы, и наоборот. Это более дешевый вариант, но он не всегда лучший. Большинство компаний максимально использует пространство своего здания, а также свои ресурсы и вычислительные мощности. Позволить другой компании прийти и работать в том же здании, может оказаться губительным для обеих компаний. Организация полноценной работы двух компаний в одной сети и с одним оборудованием может оказаться крайне сложной задачей и привести к проблемам безопасности.

Вы можете позволить другой компании переехать в ваше здание и работать в нем, например, если генеральный директор этой компании – ваш друг, но как быть с остальными сотрудниками, которых вы не знаете? При этом у вас появится новая группа людей, которым, возможно, потребуется привилегированный или прямой доступ к вашим ресурсам в общей среде. Это другая компания может быть вашим конкурентом на рынке, поэтому многие сотрудники компании, которую вы приютили, могут относиться к вашей компании как к угрозе, а не как к спасителю, протявшему руку помощи попавшим в беду. При такой совместной работе, пристальное внимание следует уделять вопросам предоставления прав доступа и разрешений сотрудникам другой компании к критичным для вашей компании активам и ресурсам.

Соглашения о взаимной помощи хорошо работают только в некоторых областях

деятельности, например, в области печати газет. Компаниям, работающим в этой области, требуются весьма специфичные технологии и оборудование, которые не предоставляются «по подписке». Руководители таких компаний следуют принципу «ты поможешь мне, я помогу тебе». Для компаний, работающих в большинстве других областей, такие соглашения, как правило, являются не более чем вторичным вариантом, «планом Б». Тем не менее, многие компании выбрали именно такое решение, что связано с его дешивизной, либо отсутствием иных вариантов.

Если компания решает принять участие в подобном двустороннем соглашении, ей нужно заранее решить ряд важных вопросов:

- Как долго здание будет доступно компании, в случае необходимости?
- Что потребуется для интеграции двух сред и их последующей поддержки?
- Через какое время компания при необходимости сможет переместиться в это здание?
- Какие могут возникнуть проблемы взаимодействия?
- Какой объем ресурсов будет доступен компании при необходимости?
- Как будут решаться разногласия и конфликты?
- Как будет выполняться управление изменениями и конфигурациями?
- Насколько часто можно проводить учения и тестирования?
- Каким образом можно надежно защитить критичные активы обеих компаний?

Резервные площадки

Некоторые компании принимают решение о создании **собственной резервной площадки** (redundant site), которую они оборудуют и настраивают точно так же, как и свою основную площадку. Такая площадка принадлежит компании, и является полным зеркальным отражением основной среды. Это один из самых дорогих вариантов резервирования здания, поскольку на резервной площадке должна постоянно и в полном объеме поддерживаться готовая к работе среда, которая в обычное время не используется в работе компании, она используется только в аварийных ситуациях, когда на нее перемещается работа компании. Но ее высокая стоимость является относительной. Если прерывание работы компании всего на несколько часов может привести к многомиллионным потерям, высокая стоимость резервной площадки будет полностью оправдана. Кроме того, для ряда компаний наличие резервной площадки является обязательным требованием, в таком случае дороговизна этого решения не принимается во внимание.

Другим типом резервной площадки является **мобильная «горячая» площадка** (rolling hot site). Она может быть реализована в задней части большого грузового автомобиля или в виде прицепа, который легко превращается в небольшое серверное помещение или рабочую область, в которой может быть организовано несколько рабочих мест. Такой грузовой автомобиль и прицеп заранее оснащены всем необходимым: электроэнергией, телекоммуникациями и системами, обеспечивающими обработку данных. Он может стоять на стоянке компании или в каком-либо другом месте. Похожим решением является небольшой сборный дом, который можно легко и быстро собрать. У многих военных организаций и крупных страховых компаний есть мобильные «горячие» площадки, на которых заранее установлено все необходимое оборудование, поскольку им часто требуется гибкость, позволяющая быстро перенести некоторые или все свои вычислительные мощности в другие места по всему миру в зависимости от того, где в этом возникает необходимость.

Другим вариантом для компаний является одновременное использование **нескольких центров обработки данных** (multiple processing centers). Компания может обладать

десятком различных зданий, расположенных по всему миру, в которых установлено все необходимое оборудование и программное обеспечение, достаточное для переноса функций обработки данных из одного здания в другое всего за несколько секунд при возникновении необходимости. Такая технология может быть реализована как между зданиями одной компании, так и между зданием компании и зданием третьей стороны. Некоторые поставщики предлагают своим клиентам услуги такого рода. В этом случае, при возникновении прерывания процесса обработки данных в компании, все или некоторые из компонентов обработки могут быть быстро перемещены на серверы поставщика.

Компания должна понимать все возможные варианты организации альтернативных площадок, чтобы выбрать действительно наилучший для себя вариант, учитывающий реальные потребности бизнеса компании.

2.6. Восстановление технической среды

На данный момент у команды ВСР есть схемы критичных бизнес-функций, которые обязательно должны выполняться, определен наилучший для компании вариант альтернативной площадки. Теперь команда должна углубиться в детали, такие как обеспечение резервирования для следующих элементов:

- Сетевое и компьютерное оборудование
- Коммуникационное оборудование для передачи голоса и данных
- Люди
- Средства для транспортировки оборудования и персонала
- Системы вентиляции, отопления и кондиционирования
- Системы обеспечения безопасности персонала и данных
- Различные расходные материалы (бумага, бланки, кабели и т.п.)
- Документация

Команде должна быть хорошо понятна организация работы имеющейся у компании технической среды. Члены команды, выполняющие планирование, должны детально знать все подробности о работе сети, коммуникаций, компьютеров, сетевого оборудования, программного обеспечения – всего, что необходимо для работы жизненно важных функций компании. Некоторым это покажется поразительным, но в действительности многие компании не имеют *полного* понимания, как настроена и работает их сеть, потому что она, скорее всего, была создана пять-десять лет назад и в течение этого времени постоянно росла и изменялась. В нее добавлялись новые устройства, новые компьютеры, устанавливалось новое программное обеспечение, в нее была интегрирована система IP-телефонии, первоначальная DMZ была разделена на три отдельных DMZ, организованы экстрасети с партнерами компании... Возможно, компания приобрела другую компанию и объединила ее сеть со своей. За десять лет обновились многие технологии, сменился персонал, обеспечивающий поддержку сети. В подразделениях ИТ многих компаний наблюдается текучесть кадров, каждые пять лет персонал полностью обновляется. К тому же во многих компаниях сильно устарели сами подходы и принципы организации сети, поскольку все очень заняты своими текущими задачами и ни у кого нет времени на перестройку сети.

Поэтому команда ВСР должна убедиться, что люди, которые будут восстанавливать сеть в случае ее частичного или полного разрушения, имеют достаточно сведений и знаний, и смогут надлежащим образом восстановить ее.

ПРИМЕЧАНИЕ. Многие компании используют технологии VoIP, а это значит, что при нарушении работы сети, голосовые функции будут также недоступны. Команде следует учесть возможную потребность компании в наличии дополнительной (резервной) голосовой системы.

Команде ВСР необходимо принять во внимание многие вещи, в том числе те, о которых часто забывают: замену оборудования, обновление программных продуктов, документацию, замену персонала.

Резервирование оборудования

Команда должна определить, какое оборудование необходимо для обеспечения работоспособности наиболее важных для компании функций. В состав такого оборудования могут входить серверы, пользовательские рабочие станции, маршрутизаторы, коммутаторы, ленточные накопители для резервного копирования, концентраторы и многое другое. Перечень необходимого оборудования может показаться достаточно простым, пока команда не углубится в детали. Если команда планирует, что при восстановлении серверов и рабочих станций будут использоваться созданные заранее образы их жестких дисков, ей нужно проанализировать вопрос – будут ли работать эти образы на новом оборудовании, приобретенном взамен уничтоженного или вышедшего из строя. Восстановление системы с образа вместо повторной установки и настройки программного обеспечения, может сэкономить массу времени, однако на новом оборудовании эти образы, скорее всего, работать не будут. Поэтому команда ВСР должна специально для такого случая разработать запасной план, предусматривающий восстановление каждой из критических систем «с нуля», без использования образов.

Также, команда ВСР должна выяснить, сколько времени займет доставка нового оборудования. Если команда определила, что для восстановления работоспособности компании на удаленной площадке ей понадобится 20 серверов и 50 рабочих станций, она должна узнать у поставщика, с которым работает компания, сколько времени займет у него поставка этого оборудования на удаленную площадку с момента подачи компанией соответствующей заявки. Может оказаться, что поставщик компании сможет выполнить такой заказ только в течение трех недель. Если компания узнает об этом, когда авария уже произойдет, это может стать для нее катастрофой. Чтобы избежать этого, команда должна проанализировать текущие соглашения об уровне услуг (SLA - Service Level Agreement) поставщиков компании и, при необходимости, организовать заключение новых соглашений, чтобы избежать дополнительного ущерба, вызванного задержками поставки оборудования. После того, как параметры SLA стали известны, команда должна сделать выбор между заблаговременным приобретением избыточных резервных систем и зависимостью от поставщика (сроков поставки). Как уже говорилось ранее, в случае выявления потенциальных рисков компании, следует принять превентивные меры для уменьшения потенциального ущерба. После расчета значений MTD, команда будет знать, как долго компания сможет прожить без того или иного устройства. Эти значения должны быть учтены в процессе принятия решения относительно покупки резервных систем для быстрой замены, либо зависимости от SLA с поставщиком. Если ущерб компании от недоступности даже одного сервера составляет \$50 000 в час, команде следует принять решение в пользу закупки избыточных резервных систем и оборудования.

В случае если компания использует какое-либо устаревшее оборудование или компьютеры, сможет ли она найти ему замену в случае необходимости? Команда должна выявить устаревшие устройства и определить, какие риски несет компания в случае невозможности их замены. Часто эти риски оказываются стимулом для перехода компании от устаревших систем к более современным, чтобы обеспечить возможность замены.

ПРИМЕЧАНИЕ. На рынке существуют ленточные устройства резервного копирования, в которых используются технологии различных типов (Digital Linear Tape, Digital Audio Tape, Advanced Intelligent Tape). Команда должна уточнить тип технологии, используемой в компании, чтобы определить, у какого поставщика можно будет приобрести ленточный накопитель, в случае, если его потребуется заменить.

Резервирование программного обеспечения

В подразделениях ИТ большинства компаний дистрибутивные комплекты программного обеспечения и лицензионные ключи от них хранятся в множестве различных мест, а не централизованно. Как сотрудники ИТ получают доступ к этим дистрибутивным комплектам и лицензионным ключам в случае разрушения здания? Команда ВСР должна обязательно оформить перечень программного обеспечения, необходимого для функционирования критически важных функций, организовать хранение копий дистрибутивов и лицензионных ключей на удаленной площадке. Ведь при отсутствии программного обеспечения, оборудование, как правило, окажется бесполезным. Должны быть созданы резервные копии такого программного обеспечения, как, например, бизнес-приложения, утилиты, системы управления базами данных, операционные системы. План непрерывности должен содержать требования по резервному копированию и защите этого программного обеспечения, а не только оборудования и данных.

Команда ВСР должна убедиться, что существует, по меньшей мере, две копии операционных систем и критически важных приложений, используемых в компании. Одна копия должна храниться на основной площадке, а другая – в безопасном месте на удаленной площадке. Эти копии следует периодически проверять и создавать новые копии, когда в компании внедряются новые версии этого программного обеспечения.

Нередко компании заказывают у разработчиков индивидуальное специализированное программное обеспечение. Например, крупному банку может потребоваться автоматизированная банковская система, которая позволит его служащим работать со счетами, хранить информацию клиентов в базах данных, предоставлять функции дистанционного банковского обслуживания, выполнять репликацию данных, а также обеспечит выполнение сотен других видов банковских операций. Это сложное специализированное программное обеспечение, разработкой которого занимаются всего несколько производителей, специализирующихся на банковской отрасли. После покупки такого программного обеспечения банком, оно должно быть доработано и настроено, чтобы учесть его индивидуальные особенности и потребности. Когда это программное обеспечение внедрено, работа банка всецело зависит от его непрерывной работы.

При этом производители такого программного обеспечения в большинстве случаев не предоставляют своим заказчикам исходных кодов, заказчик получает только скомпилированные версии. Но что будет, если производитель прекратит свое существование по той или иной причине? Заказчику, внедрившему это программное обеспечение, потребуются новый производитель, который будет поддерживать и дорабатывать это программное обеспечение, но для выполнения этой работы новому производителю будут необходимы исходные коды.

Механизмом защиты в данном случае является *передача исходного кода программного обеспечения на хранение независимой третьей стороне* (Software Escrow). Эта третья сторона хранит у себя исходный код, резервную копию скомпилированного программного обеспечения (дистрибутива), комплект инструкций и другие вспомогательные материалы. При этом заключается трехсторонний договор (между производителем программного обеспечения, заказчиком и третьей стороной), который описывает, кто, что и при каких обстоятельствах может делать с исходным кодом. Обычно в этом договоре указывается, что заказчик может получить доступ к исходному коду, только в случае, если производитель прекращает свое существование или по иным причинам не может выполнять свои обязательства перед заказчиком, либо если производитель нарушает условия первоначального договора с заказчиком. Если происходит любое событие из этого перечня, клиент может получить доступ к исходным кодам, обратившись к третьей стороне, а затем заключить договор с новым производителем.

Работа многих компаний была парализована из-за того, что они не использовали Software Escrow. Эти компании платили разработчику за создание специализированного

программного обеспечения, а когда фирма разработчика разорвалась, они теряли возможность выполнения своих важнейших функций.

Комитет ВСР должен выявить эту проблему (уязвимость) в процессе проведения анализа и реализовать превентивные контрмеры: Software Escrow.

Документация

Документирование многим людям кажется просто ужасной задачей, ведь у них много других важных дел, помимо документирования процессов и процедур. Без надлежащей документации, компания не сможет восстановить работу после чрезвычайной ситуации, несмотря на добросовестно выполненную работу по резервированию оборудования и программных средств на альтернативной площадке, качественное сопровождение этого процесса, выполнение ежедневного резервного копирования информации. Просто никто не будет знать, как воспользоваться всем этим, чтобы восстановить работу всех важных процессов и процедур.

Даже восстановление отдельных файлов может вызвать трудности, а восстановление всей инфраструктуры и сети компании, пострадавшей в результате наводнения, может оказаться крайне сложным, или вообще невозможным. Все важные процедуры должны быть хорошо задокументированы, поскольку процесс восстановления, скорее всего, будет происходить в атмосфере хаоса, полной неразберихи и неопределенности. Документация должна содержать подробную информацию о том, как правильно установить образы дисков, настроить операционные системы и серверы, установить утилиты и прикладное программное обеспечение. Другая часть документации должна описывать дерево вызовов (calling tree), которое указывает, кто с кем должен контактировать и в какой последовательности. В документации должна быть указана контактная информация отдельных поставщиков, местных служб по чрезвычайным ситуациям, площадок в удаленных зданиях и любых других людей и компаний, с которыми может потребоваться контактировать в чрезвычайной ситуации.

Большинство сетей используется непрерывно. Одно программное обеспечение устанавливается вместо другого, конфигурации программного обеспечения периодически подстраиваются для учета изменений среды, устанавливаются пакеты обновлений и патчи, предназначенные для исправления ошибок. Ожидать, что один человек (или группа) сможет в критической ситуации по памяти повторить все эти шаги и вернуть работу всей среды в состояние, в котором она находилась до аварии, это просто фантастика.

Поэтому компании все же необходимо обратить внимание на эту страшную задачу – документирование. Документирование – это неотъемлемая часть планирования восстановления после аварий и обеспечения непрерывности бизнеса.

Следует создать роль (одну или несколько), ответственную за подготовку и поддержание актуальности необходимой документации. Документация должна быть разработана, она должна быть понятна, актуальна и надежно защищена. После выявления командой ВСР необходимости выполнения определенных задач, эти задачи должны быть распределены по ответственным сотрудникам, а эти сотрудники должны своевременно отчитаться об их выполнении. Иначе работа команды ВСР может оказаться пустой тратой времени и ресурсов для компании, а компания по-прежнему останется уязвимой.

ПРИМЕЧАНИЕ. Компании может потребоваться организовать взаимодействие с официальными лицами в правительстве и службах реагирования на чрезвычайные ситуации, чтобы иметь возможность заблаговременно получать информацию о возможных чрезвычайных ситуациях и катастрофах в масштабах города или региона. Также, команде следует контактировать с официальными лицами на этапе проведения BIA, чтобы получить достоверную информацию о рисках, существующих в той местности, в которой расположено здание компании, о способах доступа в зоны чрезвычайных ситуаций и т.п. Кроме того, такие контакты следует предусмотреть и для процесса восстановления.

Планы. Как вы думаете, где и в каком количестве должны храниться разработанные планы обеспечения непрерывности бизнеса и восстановления после аварий? Достаточно ли компании хранить только один экземпляр этих планов в сейте руководителя подразделения ИТ? Нет! Должно быть, как минимум, две или три копии этих планов. Одна копия должна находиться на основной площадке, а остальные – на альтернативных площадках на случай, если основное здание будет уничтожено или недоступно. Иногда еще одна копия хранится у ВСР-координатора дома. Хранение копий планов в нескольких удаленных друг от друга местах существенно снижает риск того, что план окажется недоступен в случае необходимости. Кроме того, эти планы не следует хранить в обычном шкафу, лучше хранить их в огнеупорном сейфе. Если планы хранятся вне зданий, принадлежащих компании, должны быть предприняты меры, обеспечивающие уровень их защиты, аналогичный уровню защиты при хранении в здании компании.

Люди

Одним из важнейших ресурсов, о котором часто забывают при планировании непрерывности – это люди. Компания может восстановить функционирование своей сети, установить и запустить критичные приложения, восстановить данные... и только потом понять, что ответ на вопрос «кто будет пользоваться всем этим?» - неизвестен. Люди являются критическим компонентом для любых процессов восстановления и обеспечения непрерывности, поэтому необходимо, чтобы этот компонент был полностью интегрирован в соответствующие планы.

Что случится, если компания переедет в другое здание, расположенное на расстоянии в 200 километров от основного здания? Нельзя ожидать при этом, что персонал просто так согласится ездить туда-сюда из дома на работу. Может быть следует оплатить необходимым сотрудникам временное проживание в новом месте? Или лучше оплатить их расходы на дорогу? А может компании нанять новых сотрудников в том месте, где находится ее альтернативное здание? Какие навыки от них требуется? Команда ВСР должна получить четкие ответы на весь этот длинный список вопросов.

В случае возникновения широкомасштабной катастрофы, которая окажет воздействие не только на здание компании, но также и на окружающую местность, включая дома, как вы думаете, о чем сотрудники будут больше волноваться – о компании или о своей семье? Некоторые компании считают само собой разумеющимся, что сотрудники будут доступны и готовы помочь восстановить работоспособность компании, а в действительности им нужно будет находиться дома, для выполнения своих обязанностей перед своей семьей.

Как это ни печально, но некоторые сотрудники могут погибнуть в случае катастрофы и команда должна учесть, каким образом их можно будет быстро заменить. Конечно, это не очень приятно, но такова реальность. Команда должна выявить все возможные угрозы, продумать все вопросы и заранее подготовить соответствующие решения.

Компаниям следует **спланировать последовательность замены руководителей** (executive succession planning). В этом плане должно быть предусмотрено, что если кто-то из высшего руководства недоступен, уволен из компании или погиб, компания должна иметь заранее продуманные шаги, обеспечивающие ее защиту. Потеря даже одного топ-менеджера может создать «дыру» в организации работы компании, вакуум в управленческом составе, который должен быть быстро заполнен правильным человеком. Последовательности в этом плане определяют, кто возьмет на себя обязанности этой должности (роли). Многие компании имеют должности заместителей. Например, в компании может быть заместитель директора по ИТ, заместитель финансового директора и заместитель генерального директора, уже готовые выполнять необходимые задачи, если не доступны руководители, которых они замещают.

Во многих крупных компаниях существует политика, которая запрещает подвергать одновременно одному и тому же риску двух и более топ-менеджеров. Например, генеральный директор и президент компании не могут путешествовать в одном самолете. Если этот самолет упадет, оба этих руководителя могут погибнуть и компания может

оказаться в опасности. Именно поэтому вы очень редко можете увидеть Президента США в компании с Вице-президентом.

Ссылки по теме:

- BCP IT Examination Handbook, Federal Financial Institutions Examination Council (March 2003)

2.7. Восстановление пользовательской среды

Функционирование среды, в которой работают конечные пользователи, должно быть восстановлено максимально быстро после катастрофы. Для этого команда ВСР должна понимать текущее операционное и техническое функционирование среды и исследовать ее критичные части, на предмет возможностей по их дублированию.

Первым вопросом в отношении пользователей является вопрос о порядке их уведомления о катастрофе и об их дальнейших действиях. Для решения этого вопроса, на случай катастрофы может быть разработано дерево вызовов для руководителей: человек на вершине дерева звонит двум руководителям, они звонят трем руководителям и т.д. пока все руководители не будут проинформированы. Каждый руководитель отвечает за уведомление подчиненных ему сотрудников. После этого, один или два человека должны заняться координацией вопросов, относящихся к пользователям. Среди таких вопросов может быть их перемещение в новое здание, проверка наличия у них необходимых для выполнения своих задач ресурсов, восстановление данных, обеспечение взаимодействия между различными группами. Люди, выполняющие работу координаторов, должны быть легко узнаваемы, например, они могут носить «аварийную» кепку и майку, они должны находиться в таких местах, где все их видят. Это упростит работу и снизит панику в это трудное и стрессовое время.

В большинстве случаев, после катастрофы только ключевой персонал возвращается на работу. Для определения этого ключевого персонала, Комитет ВСР на этапе анализа идентифицирует наиболее критичные функции, работа которых должна быть восстановлена в первую очередь, а также сотрудников, выполняющих эти функции, которые должны первыми вернуться к работе. Процесс восстановления пользовательской среды выполняется в несколько этапов. На первом этапе обеспечивается возможность для возвращения к работе наиболее критичных подразделений, на втором этапе – вторых по степени критичности и т. д.

Команда ВСР должна определить потребности пользователей, например, могут ли они работать за локальными компьютерами или они обязательно должны быть подключены к сети для выполнения определенных задач. К примеру, в финансовой компании сотрудники могут использовать локальные компьютеры для выполнения отдельных задач, таких как заполнение бланков документов, обработка текстов, выполнение задач по учету, но для выполнения других операций им может в обязательном порядке требоваться подключение к сети – для работы с базами данных, обмена информацией и т. п.

Команда ВСР должна определить, какие из автоматизированных задач могут выполняться вручную, при необходимости. Если сеть отключилась на 12 часов, какие важные задачи могут быть выполнены с помощью ручки и листа бумаги? Если соединение с Интернетом пропало на пять часов, какую важную информацию можно передать с помощью телефонного звонка? Могут ли курьеры передавать информацию в случае неработоспособности внутренней электронной почты? Сегодня мы полностью зависим от технологий, и часто нам кажется, что они всегда будут доступны нам для использования. Но, увы, это не так. Задачей команды ВСР является минимизация времени недоступности информационных технологий и предоставление решения по организации работы на это время.

2.8. Варианты резервного копирования данных

Резервирование необходимо выполнять в отношении оборудования, программного обеспечения, данных, персонала, а также здания (офиса). Команда ВСР должна определить, какие компоненты необходимы для «выживания» компании, и какие для них нужны варианты резервирования.

Для большинства компаний информация стала одним из самых критичных активов. Среди такой информации могут быть финансовые отчеты, проекты новых продуктов, сведения о клиентах, описания продуктов, коммерческая тайна и многое другое. В Домене 01 мы рассматривали процедуры анализа рисков и процессы классификации данных. В обязанности команды ВСР не входит внедрение и поддержание процедур классификации данных компании, но команда должна выявить риск, которому подвержена компания, если у нее не внедрены эти процедуры. Это должно быть представлено как уязвимость, о которой должно быть сообщено руководству. Руководство должно организовать другую группу сотрудников, которые проведут инвентаризацию данных компании, определят критерии потерь, проведут классификацию данных и процессов.

В обязанности команды ВСР входит предоставление решений для защиты этих данных и определения способов их восстановления после аварий. В этом Домене мы рассмотрим различные способы, которыми могут быть защищены и, при необходимости, восстановлены данные.

Обычно данные изменяются гораздо чаще, чем оборудование и программное обеспечение, поэтому процедуры резервного копирования данных должны проводиться на постоянной основе. Процесс резервного копирования данных должен быть понятным, обоснованным и разумным. Если данные изменяются несколько раз в день, процедура резервного копирования должна выполняться не реже одного раза в день (например, каждую ночь), чтобы обеспечить сохранность произведенных изменений. Однако если данные изменяются редко, например, раз в месяц, выполнение ежедневного резервного копирования будет пустой тратой времени и ресурсов. Резервное копирование файлов и произошедших в них изменений обычно более предпочтительно, чем создание множества копий одних и тех же файлов. Системы резервного копирования обычно создают журнал изменений, произошедших с каждым файлом, и работают с этим журналом отдельно от исходных файлов.

Персонал, обеспечивающий сопровождение систем, должен определить, какие данные подлежат резервному копированию и с какой периодичностью. Резервное копирование может быть полным, дифференциальным или инкрементальным; обычно используются некоторые комбинации различных типов резервного копирования. Большинство файлов не меняется ежедневно, поэтому с целью более бережного использования времени и ресурсов, следует разработать план резервного копирования, который не будет предусматривать постоянного резервного копирования данных (файлов), которые не меняются. Но как можно определить, какие данные изменились и нуждаются в резервном копировании, не просматривая при этом дату последнего изменения каждого файла? Это делается с помощью бита архивирования (archive bit). Файловые системы операционных систем отслеживают изменения файлов и устанавливают бит архивирования. Если был создан новый файл или изменен существующий, файловая система устанавливает для этого файла бит архивирования в 1. Программное обеспечение, выполняющее резервное копирование, просматривает состояние этого бита и на основании него принимает решение, нужно ли включать этот файл в резервную копию или нет.

Первым шагом является создание **полной резервной копии** (full backup), которая представляет собой копию всех данных на некотором внешнем устройстве хранения. В процессе создания полной резервной копии бит архивирования очищается (устанавливается в 0). Компания может решить выполнять только полное резервное копирование,

позволяющее восстанавливать данные в одно действие, однако при этом создание резервных копий и восстановление с них может занимать много времени.

Большинство компаний использует комбинацию полного резервного копирования с дифференциальным или инкрементальным резервным копированием. При создании **дифференциальной резервной копии** (differential process) копируются только те файлы, которые были изменены с момента предыдущего полного резервного копирования. При необходимости восстановления данных, сначала восстанавливаются данные с полной резервной копии, а затем поверх них записываются данные из дифференциальной резервной копии. При создании дифференциальной копии значение бита архивирования не изменяется.

При создании **инкрементальной резервной копии** (incremental process) копируются все файлы, которые были изменены с момента последнего создания полной или инкрементальной резервной копии. В процессе создания инкрементальной копии бит архивирования сбрасывается (устанавливается в 0). При необходимости восстановления данных, сначала производится восстановление файлов с полной резервной копии, а затем в правильном порядке поверх них записываются файлы из каждой созданной инкрементальной резервной копии.

Какой вариант резервного копирования является наилучшим? Если компания нужно сделать процесс резервного копирования и восстановления простым и прямолинейным, она может выбрать использование только полного резервного копирования. Однако это потребует большого количества внешних носителей информации (или большого объема дискового пространства) для хранения резервных копий, а процесс создания резервных копий будет занимать достаточно много времени. Использование дифференциального и инкрементального резервного копирования является более сложным процессом, но они требуют значительно меньше ресурсов и времени. Дифференциальное резервное копирование требует больше времени на создание резервных копий, но меньше времени на этапе восстановления данных, который всегда делится только на два этапа (восстановление полной копии и восстановление дифференциальной копии). Инкрементальное резервное копирование, по сравнению с дифференциальным, занимает меньше времени при создании резервных копий (т.к. копируется меньший объем информации), но для восстановления информации требуется больше времени из-за необходимости восстановления информации с нескольких копий.

Выбирая вариант организации резервного копирования, не следует смешивать дифференциальное и инкрементальное резервное копирование. Это может привести к потере данных, поскольку при создании инкрементальной копии бит архивирования изменяется, а при создании дифференциальной копии – нет.

Для критичных данных должно быть организовано резервное копирование, а резервные копии должны храниться как на основной площадке, так и на внешней. Резервные копии на основной площадке должны быть легко доступны в некатастрофичных случаях, позволяя быстро восстановить данные и работу компании. Однако одних только резервных копий на основной площадке не достаточно для обеспечения реальной защиты. Дополнительно резервные копии следует хранить в отдельном здании на случай чрезвычайной ситуации или катастрофы. Необходимо принять решение, на каком удалении от основной площадки должно находиться это альтернативное здание с резервной площадкой. Если резервная площадка, на которой хранятся дополнительные резервные копии, расположена вблизи основной площадки, это упрощает доступ к резервным копиям в случае необходимости, однако это также подвергает резервные копии опасности в случае широкомасштабных катастроф, в результате которых компания может утратить как основную, так и резервную площадку. Разумнее организовать резервную площадку как можно дальше от основной. Конечно, это усложнит доступ к резервным копиям, но снизит общие риски компании. Некоторые компании принимают решение об организации нескольких резервных площадок,

одна из которых находится вблизи от основной, а другая (другие) – на значительном расстоянии.

На основной площадке носители информации с резервными копиями должны храниться в огнеупорных, жаропрочных, водонепроницаемых сейфах. Процедуры резервного копирования и восстановления данных должны быть просты и понятны каждому оператору или администратору, даже если он не достаточно хорошо знаком с соответствующей системой. Иначе может случиться так, что в аварийной ситуации того парня, который всегда выполнял резервное копирование и знает как восстанавливать данные, может не быть поблизости, либо временно компании может потребоваться привлечь внешних консультантов (подрядчиков) для восстановления своей работы.

Стратегия резервного копирования должна предусматривать возможность возникновения проблем на любом шаге процесса. Специально на случай возникновения непредвиденной проблемы, приведшей к повреждению данных в процессе резервного копирования или восстановления, в стратегии должна быть предусмотрена возможность «отката» произведенных изменений, либо реконструкции данных с самого начала.

А действительно ли мы можем восстановить данные? Резервное копирование – прекрасная вещь, но еще лучше убедиться, что существует возможность надлежащего восстановления данных из созданных резервных копий. У многих компаний возникает ложное чувство безопасности, основанное на том, что у них построен хорошо организованный и эффективный процесс резервного копирования критичных данных. Это чувство безопасности может улетучиться в секунды, когда компания после аварии увидит, что процесс восстановления не работает так, как ожидалось. Например, известен случай с одной крупной компанией, у которой была внешняя резервная площадка, курьер компании еженедельно отвозил резервные копии на резервную площадку и убирал их на хранение в сейф. Курьер ездил на метро и часто ставил сумку с кассетами на пол, пока ожидал поезд. Никто не учел, что в метро постоянно присутствуют мощные магнитные поля, действующие как большой магнит и приводящие к уничтожению или повреждению данных на магнитной ленте. Компания не считала нужным тестировать процессы восстановления, но когда у нее произошла авария и потребовалось восстанавливать данные с резервных копий, она была очень сильно удивлена тому, что данные за последние три года оказались повреждены и непригодны для использования.

2.9. Средства автоматизированного резервного копирования

Выполнение вручную операций резервного копирования систем и данных может отнимать много времени, вести к ошибкам, в результате чего такое решение может оказаться слишком дорогостоящим. Другим подходом к этому процессу является выполнение операций резервного копирования с помощью автоматизированных средств. Хотя средства автоматизированного резервного копирования обычно достаточно дороги, они выполняют эту работу гораздо быстрее и точнее, что необходимо для систем, функционирующих в режиме реального времени, данные в которых меняются очень часто.

Среди множества технологий и способов резервного копирования данных можно выделить технологию *теневого копирования дисков* (disk shadowing), которая очень похожа на зеркалирование (disk mirroring) дисков.

ПРИМЕЧАНИЕ. Дуплексирование дисков (disk duplexing) означает, что используется более одного дискового контроллера. При этом если один контроллер выходит из строя, другой остается доступным и сразу готов к работе.

Теневое копирование дисков применяется для обеспечения гарантий доступности данных и создания отказоустойчивого решения путем дублирования на аппаратном уровне и обеспечения наличия двух или более копий информации. Данные автоматически записываются на два или более идентичных диска. При использовании зеркалирования

дисков, для каждого диска требуется дополнительный зеркальный диск, при этом оба диска содержат абсолютно одинаковую информацию, данные записываются на оба диска одновременно в синхронном режиме. При использовании теневого копирования дисков, создание теневой копии происходит в асинхронном режиме, данные хранятся в виде образов на двух или более дисках.

Системы, которым нужно работать с данными, записанными на теневых дисках, подключаются ко всем этим дискам одновременно. Для пользователя эта технология абсолютно прозрачна, для него все эти диски выглядят просто как один единственный диск. Если пользователю нужно открыть какой-либо файл, ему не нужно думать, на каком из дисков он находится. При сохранении файла, данные записываются на все диски, включенные в теневой набор (shadow set).

Теневое копирование дисков обеспечивает резервное хранение информации, копии создаются в режиме реального времени, что может снизить или полностью удовлетворить потребности компании в выполнении периодического ручного резервного копирования. Другим преимуществом такого решения является то, что оно может повысить скорость операций чтения. В связи с дублированием данных на нескольких дисках, теневой набор может одновременно выполнять несколько операций чтения.

Теневое копирование диска часто выглядит дорогим решением, поскольку для хранения одних и тех же данных используются два или более жестких дисков. При этом если данные, которые нужно хранить компании, занимают объем 100 жестких дисков, компании нужно будет купить и поддерживать не менее 200 жестких дисков. Однако, если компании нужна отказоустойчивость, она может выбрать это решение.

Если один дисковый накопитель выходит из строя, остается доступен, как минимум, еще один накопитель в этом теневом наборе. Для замены вышедшего из строя диска, к теневому набору может быть подключен новый диск, и данные на него будут скопированы с оставшихся дисков теневого набора. Такое копирование не всегда может выполняться непосредственно в процессе работы теневого набора, т.е. данные будут недоступны в течение некоторого времени, пока выполняется их копирование на новый диск. Большинство продуктов, реализующих функциональность теневого копирования, позволяют выполнять копирование на новый диск, подключенный к теневому набору, непосредственно в процессе работы (в режиме онлайн), без перерыва в предоставлении доступа к данным.

Другими решениями, о которых следует знать компании, являются электронное хранение и удаленное журналирование. При использовании **электронного хранения** (electronic vaulting) копии файлов создаются по мере их изменения и периодически переносятся на резервную площадку. Перенос копий файлов выполняется в пакетном режиме (batches), а не в режиме реального времени. Компания может самостоятельно настроить интервал переноса измененных файлов на резервную площадку, например, раз в час, день, неделю или месяц. При этом информация может сохраняться на внешней резервной площадке и за короткое время доставляться с нее в случае необходимости.

Такой вариант резервного копирования применяется во многих финансовых компаниях. Например, когда банковский кассир зачисляет деньги на депозит, изменение остатка на клиентском счете производится не только локально в базе данных соответствующего отделения банка, но и на удаленной резервной площадке, посредством чего банк обеспечивает резервное копирование всей информации о клиентах.

Электронное хранение – это способ переноса больших объемов информации на внешнюю площадку с целью резервирования. **Создание удаленных журналов** (remote journaling) является другим способом переноса данных на внешнюю площадку, но в этом случае на внешнюю площадку обычно переносятся журналы или логи транзакций, а не сами файлы. Эти логи содержат только изменения (deltas), произведенные в отдельных файлах. Если

данные будут повреждены и их потребуется восстановить, компания будет использовать эти логи для реконструкции утраченных данных. Журналирование наиболее эффективно для восстановления баз данных, для этого требуется только повторное применение к ней последовательности изменений.

ПРИМЕЧАНИЕ. Создание удаленных журналов выполняется в режиме реального времени, в процессе него на резервную площадку передаются только изменения файлов. Электронное хранение выполняется посредством пакетных заданий, которые переносят на резервную площадку обновленные файлы целиком.

Компании может потребоваться сохранять различные версии программного обеспечения и файлов, в особенности в среде разработки программного обеспечения. При этом исходные коды и скомпилированные файлы программ следует резервировать вместе с библиотеками, патчами и исправлениями. Внешняя площадка должна являться зеркальной копией основной площадки, а не просто хранилищем исходных кодов. На каждой площадке должен храниться полный набор актуальной информации и файлов.

Другой технологией резервного копирования является хранение данных на ленточных носителях. Многие компании записывают резервные копии своих данных на ленту, которую затем курьер или сотрудник перевозит на внешнюю площадку. Автоматизированное **ленточное хранение** (tape vaulting) позволяет передавать данные по последовательному каналу на систему резервного копирования, установленную непосредственно на внешней площадке. Единственное, что нужно при этом делать компании, это поддерживать работу системы резервного копирования на внешней площадке и периодически менять кассеты. Данные достаточно быстро резервируются и не менее быстро могут быть восстановлены при необходимости. Эта технология снижает количество шагов, выполняемых вручную при использовании традиционных процедур резервного копирования на ленту.

Чем больше шагов выполняется вручную, тем выше вероятность ошибок. Использование автоматизированного ленточного хранения повышает скорость восстановления данных, снижает число ошибок, позволяет чаще выполнять резервное копирование.

2.10. Выбор здания для хранения резервной информации

При выборе здания, в котором будут храниться носители информации с резервными копиями, компания должна учесть множество вопросов и дать на них четкие ответы. Ниже приведены некоторые из таких вопросов, которые должна решить компания, прежде чем согласиться на услуги того или иного поставщика:

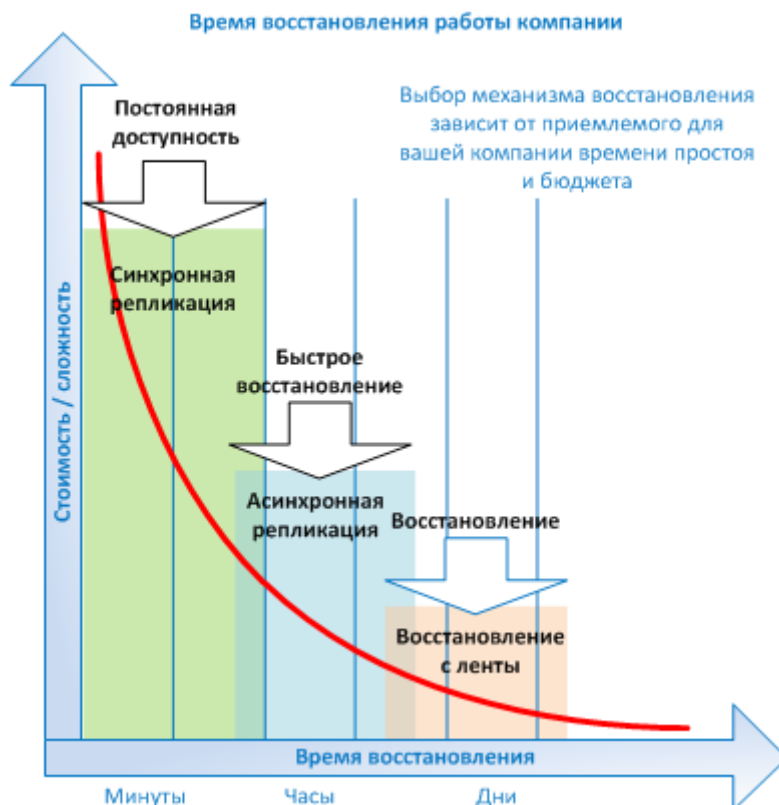
- Может ли компания получить доступ к носителям информации в нужное ей время?
- Закрывается ли здание на выходные и праздники, работает ли оно постоянно или только по определенному графику?
- Механизмы контроля доступа связаны с системой оповещения и/или местным полицейским участком?
- Обеспечивается ли в этом здании защита носителей информации от различных угроз?
- Какие транспортные услуги предоставляются?
- Существуют ли опасности, непосредственно связанные с местностью, в которой расположено здание (наводнения, землетрясения, торнадо и т.п.)?
- Установлена ли системы выявления и тушения пожара?
- Обеспечивается ли в здании управление и мониторинг температуры и влажности?
- Какой используется вид физического, административного и технического управления доступом?

Перечень вопросов, которые должна выяснить компания, будет зависеть от типа компании, ее потребностей и требований к резервной площадке.

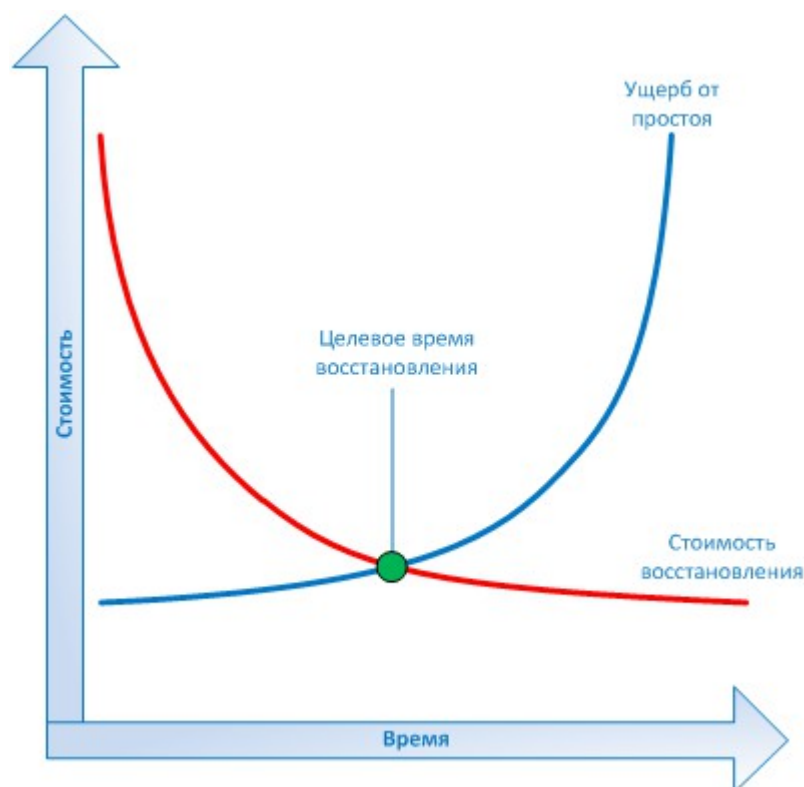
Как выбрать решение для резервного копирования данных?

На данный момент уже должна быть проведена классификация данных, основанная на их критичности для бизнеса.

- Команде ВСР нужно разделить данные по степени критичности времени их восстановления
- Самые критичные данные, которые должны быть доступны постоянно, могут быть восстановлены из зеркальных копий, систем электронного хранения или удаленного журналирования
- Остальные данные могут быть восстановлены с магнитной ленты



При асинхронной репликации первичные и вторичные данные синхронизируются всего за несколько миллисекунд, т.е. репликация осуществляется почти в режиме реального времени. При выполнении синхронной репликации первичные и вторичные копии постоянно синхронизированы, что обеспечивает дублирование данных в режиме реального времени. Команда использовать сбалансированный подход, учитывая стоимость восстановления данных и потери от их утраты. Точка, в которой пересекаются соответствующие кривые, является целевым временем восстановления.



2.11. Страхование

В процессе проведения ВИА команда наверняка выявит ряд угроз, реализацию которых компания не сможет предотвратить. Принятие в полном объеме рисков, вызванных этими угрозами, часто является слишком опасным. Именно для таких случаев и существует страхование (insurance). Решение о страховании определенных угроз и сумме страхового покрытия на случай реализации этих угроз должно быть основано на вероятности реализации угрозы и потенциальных потерях, которые понесет компания в случае их реализации. Эти данные (вероятность и потери) были определены в процессе проведения ВИА. Команде ВСР следует взаимодействовать с руководством компании, чтобы понять текущий уровень страхового покрытия, используемые компанией варианты страховки, лимиты на каждый вариант. Целью этой работы является обеспечение уверенности в том, что страховка заполнит имеющиеся недостатки превентивных контрмер, не позволяющие обеспечить надлежащую защиту от выявленной угрозы. Мы можем есть здоровую пищу, заниматься спортом, есть витамины – но все это не спасет нас от смерти. На этот случай мы страхуем свою жизнь, чтобы после смерти о наших родных позаботилась страховая компания.

Люди платят страховым компаниям различные суммы страховых премий, страхуя свое здоровье и жизнь, в зависимости от типа страховки, которую они покупают. Аналогично, различные типы страховки могут быть приобретены и компаниями, одним из таких типов является **страхование в компьютерной сфере** (cyberinsurance). Страхование в компьютерной сфере – это новый тип страховых продуктов, которые страхуют потери компании, вызванные, например, DoS-атаками, вирусами, атаками хакеров, кражами электронных данных и т.п. Когда человек страхует свою жизнь, его спрашивают, сколько ему лет, каково его здоровье, курит ли он и т.д., на основании его ответов определяется размер страховой премии. В аналогичной ситуации у компании спрашивают о ее программе безопасности, в частности, есть ли у нее система IDS, антивирусное программное обеспечение, межсетевые экраны и другие средства и меры обеспечения безопасности.

Компания также может принять решение о страховании прерывания своего бизнеса. При этом, если компания будет вынуждена прекратить свою работу на некоторое время,

страховая компания возместит ее расходы и упущенную прибыль. Другой возможностью является страхование доступности для компании ее счетов. Если компания по тем или иным причинам не сможет воспользоваться средствами на своих счетах, она получит от страховой компании компенсацию части или всех своих потерь и убытков.

Страховые контракты компании следует пересматривать на ежегодной основе, поскольку может изменяться уровень угроз, компания может решиться выпустить на рынок еще один рискованный продукт – все это должно быть надлежащим образом учтено. Покупка страховки не должна рассматриваться компанией как решение всех своих проблем и создавать ложное чувство безопасности. Сумма страхового покрытия ограничена, а если компания не проявляет должную заботу, страховая компания может получить законные основания, чтобы вообще не платить в случае возникновения аварии. Перед заключением страхового контракта, компания должна внимательно ознакомиться с ним, при этом важно прочитать и правильно понять в том числе и то, что написано мелким шрифтом, чтобы знать, в каких случаях не стоит ждать помощи от страховой компании.

2.12. Восстановление и реконструкция

Координатор ВСП должен организовать несколько различных команд, провести их обучение и тренировки, обеспечить их доступность в случае аварии. Необходимые компании типы таких команд зависят от самой компании. Ниже приведено несколько примеров команд, которые может потребоваться организовать в компании:

- Команда оценки повреждений
- Юридическая команда
- Команда взаимодействия со средствами массовой информации
- Команда восстановления сети
- Команда перемещения оборудования и персонала на новое место
- Команда восстановления
- Команда спасения имущества
- Команда безопасности
- Телекоммуникационная команда

Координатор ВСП должен понимать потребности компании и типы команд, которые должны быть организованы и обучены. Набор сотрудников в эти команды должен основываться на их знаниях и навыках. В каждой команде должен быть назначен руководитель, который будет управлять деятельностью членов команды. Руководители этих команд должны отвечать не только за достижение целей своей команды, но также и за взаимодействие с другими командами, что необходимо для обеспечения эффективной и слаженной работы команд особенно на взаимосвязанных этапах, когда работа одной команды зависит от результатов работы другой, либо когда работы нескольких различных команд должны выполняться параллельно.

Команда восстановления (restoration team) отвечает за приведение в рабочее состояние альтернативной площадки и ее окружения, **команда спасения имущества** (salvage team) отвечает за запуск работ по восстановлению основной площадки. Обе команды должны уметь выполнять множество задач, таких как установка операционных систем, настройка рабочих станций и серверов, прокладка электрической проводки и сетевых кабелей, организация сети, настройка сетевых сервисов, установка оборудования и приложений. Обе команды должны также знать, как безопасно восстанавливать данные из резервных копий, обеспечивая сохранение конфиденциальности, целостности и доступности систем и данных.

План ВСР должен описывать необходимые команды, их обязанности, процедуры информирования. План должен определять способы связи с руководителями команд в рабочее и нерабочее время.

Должна быть создана роль (или команда) для выполнения **оценки повреждений** (damage assessment) в случае аварии. Процедуры оценки должны быть задокументированы и включать следующие шаги:

- Определение причин аварии
- Оценка вероятности того, что произойдут дополнительные повреждения
- Определение бизнес-функций и областей, на которые оказано негативное воздействие
- Определение текущего уровня функционирования критичных ресурсов
- Определение ресурсов, которые должны быть заменены немедленно
- Оценка времени, которое потребуется для восстановления работы критичных функций (если время, которое потребуется для восстановления функционирования, превышает предварительно рассчитанные значения MTD (максимально допустимого времени простоя), должна быть объявлена аварийная (чрезвычайная) ситуация и план ВСР должен быть приведен в действие).

После сбора и анализа этой информации станет понятно, какие команды должны приступить к выполнению возложенных на них обязанностей, и действительно ли требуется приведение в действие плана ВСР. Координатор ВСР и команда ВСР должны разработать критерии активации плана ВСР. Если после оценки повреждений, выполняется один или более критериев, план ВСР приводится в действие и команды приступают к восстановлению.

У каждой компании будут свои критерии, поскольку бизнес-драйверы и критичные функции сильно отличаются в различных компаниях. Такие критерии могут учитывать некоторые (или все) из перечисленных ниже элементов:

- Опасность для человеческой жизни
- Опасность для безопасности города или государства
- Опасность для здания
- Опасность для критичных систем
- Приблизительное ожидаемое время простоя

После завершения оценки повреждений и активации плана, различные команды должны приступить к своей работе, что будет говорить о том, что компания перешла к этапу восстановления. У каждой команды есть собственные цели и задачи. Например, команда восстановления готовит внешнюю площадку (при необходимости), команда восстановления сети организует на ней сеть и устанавливает системы, команда перемещения начинает готовить штат к переезду в новое здание.

Процесс восстановления должен быть максимально организованным, что позволит восстановить работу компании в максимально короткие сроки. Проще сказать это, чем реализовать на практике. Именно поэтому так важно документировать все процедуры. В процессе ВИА выявляются критичные функции и необходимые для их работы ресурсы. Существуют вещи, над которыми команды должны работать совместно, чтобы максимально быстро восстановить их и запустить в работу в первую очередь. При разработке плана следует создать схемы процессов выполнения работ. Эти схемы будут использоваться различными командами для прохождения определенных этапов и документирования результатов. Например, если один из этапов не может быть завершен, пока не закуплена новая система, это должно быть указано в соответствующей схеме. Если этап завершен

только частично, это должно быть зафиксировано, чтобы команда не забыла вернуться к нему и закончить, когда это будет возможно. Эти схемы напоминают командам об их задачах, а также позволяют руководителям команд быстро оценить прогресс, возникшие затруднения и потенциальное время, которое потребуется для восстановления.

ПРИМЕЧАНИЕ. Примеры шаблонов можно найти в документе NIST *Contingency Planning Guide for Information Technology Systems* по адресу <http://csrc.nist.gov/publications/nistpubs/800-34-rev1/sp800-34-rev1.pdf>.

Когда у компании появится возможность вернуться обратно на основную площадку или полностью перейти на новую площадку, настанет время перехода компании на **этап реконструкции** (reconstruction phase). Компания не выйдет из аварийного состояния, пока ее функционирование не будет возвращено на основную площадку (или на новую площадку, построенную взамен основной), поскольку при работе на альтернативной площадке компания остается уязвимой. Должно быть решено множество логистических вопросов, прежде чем компания сможет вернуться с альтернативной площадки на основную. Ниже приведен список некоторых из таких вопросов:

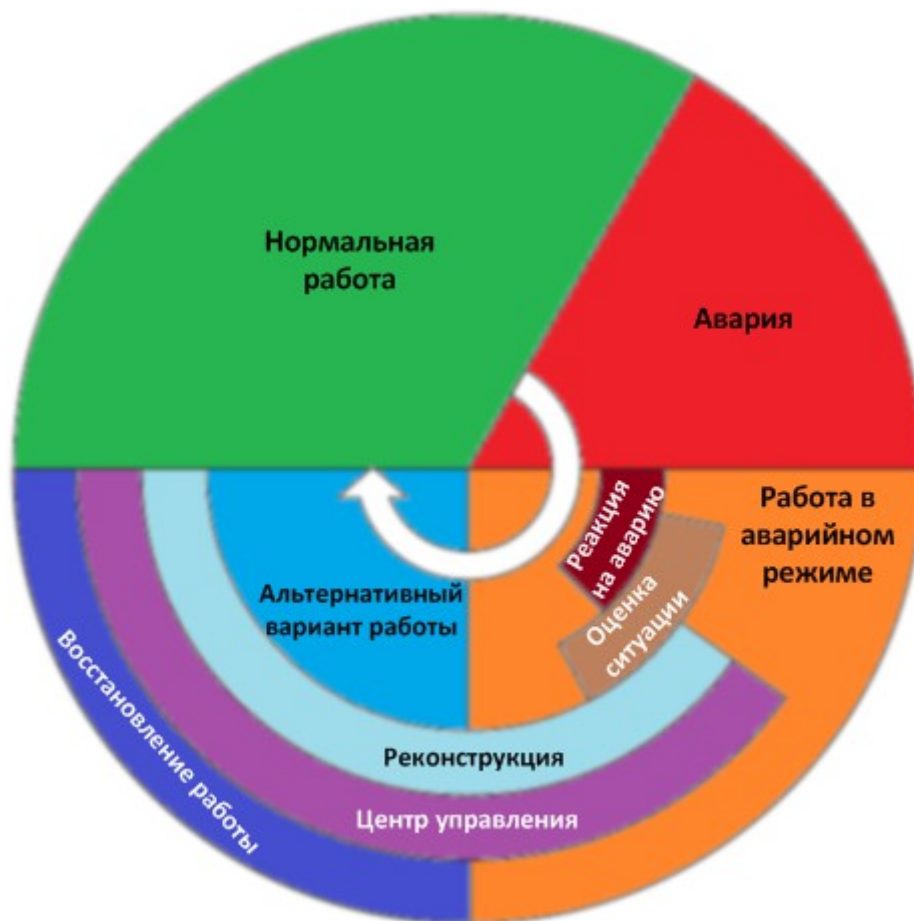
- Обеспечить безопасность персонала
- Обеспечить приемлемую внутреннюю среду (электроснабжение, инфраструктура здания, водоснабжение, отопление, вентиляция, кондиционирование)
- Обеспечить наличие расходных материалов, снабжения, а также наличие оборудования и его работоспособность
- Обеспечить наличие и работоспособность коммуникационного оборудования и связи
- Протестировать надлежащим образом новое окружение

После того, как координатор, руководство и команда спасения имущества подпишутся за готовность основного (нового) здания, команда спасения имущества должна выполнить следующие шаги:

- Сделать резервную копию данных на альтернативной площадке и восстановить их на основной (новой) площадке
- Аккуратно прекратить выполнение работ в аварийном режиме
- Безопасно доставить оборудование и персонал в основное (новое) здание

Выполнение наименее критичных операций следует возратить назад в первую очередь, поскольку в случае возникновения каких-либо проблем, например, с настройками сети или связью, важные для компании операции не будут подвержены их негативному влиянию. Зачем выполнять сложный процесс перемещения важных систем и операций со стабильно работающей площадки на основную площадку, только для того, чтобы увидеть, что она не была протестирована и находится в непригодном для использования состоянии?

Предоставьте это наименее критичным подразделениям. Если они смогут работать на основной площадке, перемещайте на нее все остальные подразделения и системы.



К настоящему моменту команда ВСП выполнила следующие шаги:

1. Разработана политика планирования непрерывности бизнеса
 - Определены границы и цели ВСП, а также роли команды ВСП
2. Выполнен ВИА
 - Идентифицированы критичные бизнес-функции, необходимые им ресурсы и значения MTD
 - Идентифицированы угрозы и рассчитано воздействие от этих угроз
 - Определены решения
 - Результаты представлены руководству
3. Определены и внедрены превентивные защитные меры
 - Внедрены защитные меры для снижения выявленных рисков компании
 - Заключен договор страхования, произведены структурные укрепления здания, внедрены решения по организации резервного копирования, установлены избыточные и отказоустойчивые механизмы и т.д.
4. Разработаны стратегии восстановления
 - Реализованы процессы восстановления работы компании в течение необходимого времени
 - Организованы необходимые команды, определены цели и разработаны процедуры для каждой команды, определены шаги процесса оповещения и критерии приведения в действие плана, выбрано решение по организации резервирования на альтернативной площадке и т.д.

Итак, команда BCP долго и хорошо работала и на данный момент имеет все указанное выше. Теперь нужно отразить все эти решения и шаги в самом плане, протестировать план, провести обучение и тренировку людей, определить порядок хранения и использования плана, разработать стратегию поддержания плана в актуальном состоянии.

Ссылки по теме:

- Business Continuity Planning & Disaster Recovery Planning Directory, Disaster Recovery World
- Business Continuity Directory, Business Continuity Planning Group

Автоматизированные средства разработки плана BCP. Сбор, анализ и поддержка данных DRP и BCP требует большого объема работы, поэтому применение для этой работы средств автоматизации может быть очень полезным. Автоматизация разработки плана может помочь вам создать:

- Настраиваемые опросные листы, основанные на шаблонах экспертных систем
- Расписания выполнения процедур восстановления после аварий
- Смоделировать сценарии «что-если»
- Отчет по анализу финансового и операционного воздействия
- Графическое представление результатов анализа
- Примеры опросных листов, форм и шаблонов
- Поддержка плана, основанная на разрешениях
- Централизованный контроль версий и интеграция
- Соответствие требованиям законодательства и регуляторов

2.13. Разработка целей плана

Если у вас нет определенных целей, как вы узнаете, что ваша работа выполнена и что ваши усилия были успешными? Цели определяются, чтобы все знали, какие требуются конечные результаты. Определение целей важно для любой задачи, но для разработки планов непрерывности бизнеса и восстановления после аварий это особенно важно. Определение целей помогает правильно распределить ресурсы и задачи, разработать необходимые стратегии, помочь с экономическим обоснованием планов и программы в целом. По сути, установленные цели являются руководством по разработке самих планов. Цели устанавливаются для возможности контроля их реализации и получения нужных результатов.

Прекрасно, мы поняли, что цели очень важны. Но цель может быть сформулирована, например, так: «Обеспечить, чтобы компания осталась на рынке, если случится землетрясение». Хорошая цель, но не очень полезная, поскольку в ней очень мало ясности и конкретики. Чтобы была действительно полезной, она должна содержать определенную ключевую информацию, такую как:

- **Обязанности.** Каждый человек, участвующий в восстановлении и обеспечении непрерывности, должен иметь свои обязанности, изложенные в письменном виде для четкого их понимания в критической ситуации и состоянии хаоса. Каждая задача должна быть назначена определенному, обоснованно выбранному, человеку. Эти люди должны знать, чего от них ожидают, зачем нужно проводить учения, отработку, готовить документацию. К примеру, человек должен знать, что он обязан сначала выключить сервер, а уже потом может с криками бежать из здания.
- **Полномочия.** Во время кризиса, очень важно знать, кто является главным. Командная работа имеет ключевое значение в таких ситуациях, а практически любая команда работает гораздо эффективнее, когда у нее есть назначенный руководитель, которому

доверяют члены команды. Такие руководители должны знать, что они обязаны определить истинное положение вещей в момент кризиса, выбрать и сообщить подчиненным сотрудникам направление, в котором они должны работать. Четкое определение полномочий будет способствовать сокращению хаоса и расширению плодотворного сотрудничества.

- **Приоритеты.** Чрезвычайно важно знать, что является действительно важным, а чем просто желательно обладать. Различные подразделения выполняют различные функции в компании. Критичные подразделения должны быть отделены от подразделений, обеспечивающих функциональность, без которой компания может спокойно прожить одну – две недели. Необходимо знать, какое подразделение должно вернуться в работу первым, какое вторым и т.д. Это обеспечит наиболее эффективное, полезное, целенаправленное и последовательное выполнение работ. Наряду с приоритетами подразделений, должны быть установлены приоритеты для систем, информации и программ. Например, может быть необходимо подготовить и запустить серверы баз данных до начала работ по восстановлению файлового сервера. Основные приоритеты должны быть установлены руководством при участии различных подразделений и персонала ИТ.
- **Внедрение и тестирование.** Конечно, прекрасно записать глубокие идеи и разработать планы, но если они не проверены и невыполнимы на практике, они могут не иметь никакой ценности. После разработки плана непрерывности, он должен быть официально введен в действие. Он должен быть задокументирован, его хранение должно быть организовано в местах, легко доступных в кризисной ситуации. Люди, на которых возложены конкретные задачи, должны быть обучены выполнению этих задач, должны быть проведены учения, чтобы люди могли на практике отработать свои задачи в различных ситуациях. Подобные учения должны проводиться не реже одного раза в год, а программа в целом должна постоянно обновляться и совершенствоваться.

Исследования показали, что 65% компаний, которые теряют свои вычислительные возможности более, чем на одну неделю, уже не могут восстановить свою работу и выходят из бизнеса. Если компания не сможет быстро восстановить свою работу, она может потерять свой бизнес и свою репутацию. В мире конкуренции у клиентов есть много вариантов. Если одна компания не может вернуться к работе после аварии или стихийного бедствия, клиенты могут перейти к другой компании и остаться с ней.

2.14. Внедрение стратегий

После определения стратегий, они должны быть задокументированы и внедрены командой ВСП. Это переводит работу с этапа планирования на этап фактической реализации и действий.

Как было сказано ранее, копии плана должны храниться в одном или нескольких местах, отличных от основной площадки, поскольку в случае, если основная площадка будет уничтожена или подвержена негативному воздействию, план все равно должен быть доступен команде. Важно, чтобы команде был доступен план как в электронной, так и в бумажной форме. Помимо планов восстановления аналогичным образом должны храниться документы с информацией о критичных процедурах и дереве вызовов. Контактная информация, которая будет необходима в чрезвычайной ситуации (часть дерева вызовов), может быть напечатана, например, на обратной стороне бейджей руководителей и сотрудников, которые участвуют в процедурах оповещения, или выданы им в виде карточек, которые можно хранить в бумажнике. В критической ситуации ценность каждой минуты очень высока и лучше потратить время на работу в рамках реакции на инцидент, чем на поиск документа или ожидание, пока загрузится ноутбук.

План должен в деталях учитывать все вопросы, которые мы рассмотрели к настоящему времени. Формат реального плана зависит от окружения, целей плана, приоритетов и выявленных угроз. После того, как каждый из этих аспектов проанализирован и документирован, разделы плана могут быть разделены на необходимые категории.

Общепринятая структура плана ВСП показана на Рисунке 7-2. План ВСП каждой компании может выглядеть по-разному, но эти ключевые разделы должны быть рассмотрены в плане любой компании. План предназначен для предварительной подготовки последовательной структуры реализации задач, предусмотренных в каждой категории. План должен обеспечивать определенную гибкость, поскольку никто не знает, какая конкретно авария или чрезвычайная ситуация произойдет и какое она окажет воздействие на компанию.

Процедуры для всех этапов плана должны быть документированы, но при этом должен быть достигнут баланс между детализацией и гибкостью, чтобы не оказалось, что компания готова только к одному единственному типу аварии.

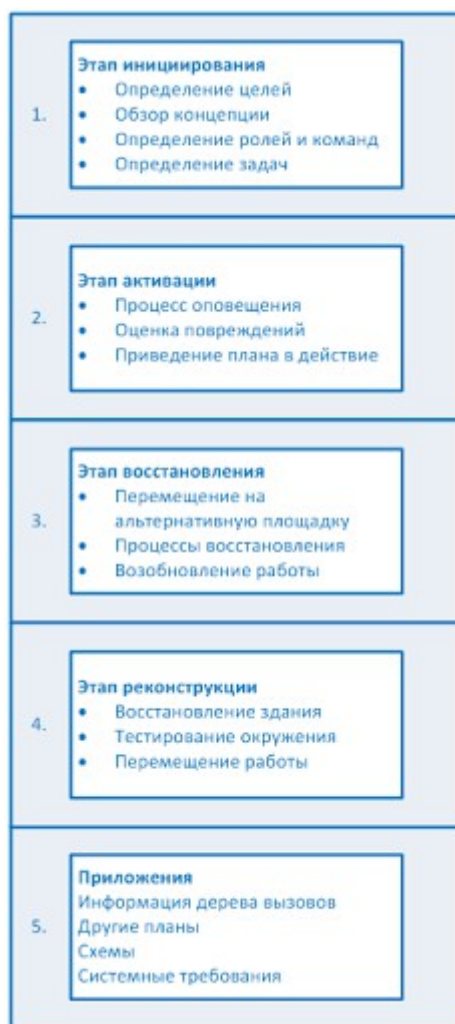


Рисунок 7-2. Основные компоненты структуры плана обеспечения непрерывности бизнеса

Некоторые компании разрабатывают отдельные планы для отдельных целей и задач. Такие планы приведены в Таблице 7-2. Руководство совместно с командой ВСП должны определить количество и типы планов, которые следует разработать и внедрить.

Тип плана	Описание
План возобновления бизнеса (Business resumption plan)	Ориентирован на воссоздание бизнес-процессов, которые должны быть восстановлены, а не на восстановление ИТ-компонентов (т.е. ориентация на процессы, а не на процедуры).
План непрерывности функционирования (Continuity of operations plan (COOP))	Определяет высшее руководство и главный офис компании после аварии. Описывает роли и полномочия, последовательность замещения должностей, а также задачи отдельных ролей.
План на случай непредвиденных ситуаций в ИТ (IT contingency plan)	План восстановления систем, сетей и основных приложений после аварии. План на случай непредвиденных обстоятельств должен быть разработан для каждой важной системы и приложения.
План коммуникаций в кризисной ситуации (Crisis communications plan)	Содержит структуру и роли внутренних и внешних коммуникаций. Определяет конкретных людей, которые должны взаимодействовать с внешними организациями и другими лицами. Содержит предварительно разработанные заявления, которые должны быть сделаны после аварии.
План реакции на компьютерный инцидент (Cyber incident response plan)	Ориентирован на воздействия вредоносного кода, атаки хакеров, вторжения и другие проблемы безопасности. Описывает процедуры реакции на инцидент.
План восстановления в случае аварии (Disaster recovery plan)	Ориентирован на восстановление различных механизмов ИТ после аварии. Тогда как план действий на случай непредвиденных ситуаций обычно направлен не на аварии, план восстановления в случае аварий предназначен именно для аварий (катастроф), которые требуют перемещения ИТ процессов на другую площадку.
План действий персонала в чрезвычайной ситуации (Occupant emergency plan)	Определяет порядок обеспечения безопасности персонала и процедуры эвакуации.

Таблица 7-2. Различные типы планов восстановления

Команда BCP может решить интегрировать многие из этих компонентов в план BCP. Чаще всего, наилучшим вариантом является оформление этих отдельных планов в виде приложений к основному плану, чтобы каждый документ был ясен, краток и действительно полезен.

2.15. Тестирование и пересмотр плана

План BCP следует регулярно тестировать, поскольку окружение постоянно меняется. Интересно, что многие компании сейчас уходят от концепции полного тестирования, поскольку такое тестирование требует значительных ресурсов и оказывается не очень продуктивным. Вместо этого они используют концепцию проведения упражнений по отдельным частям плана, что требует меньше ресурсов, приводит к меньшим стрессам для персонала, лучше сфокусировано и гораздо более продуктивно. После каждой проверки плана, обычно выявляются возможности для его усовершенствования и повышения эффективности, которые учитываются в обновленной версии плана, обеспечивая постоянное улучшение результатов. Ответственность за проведение периодического тестирования или упражнений по плану, а также поддержку плана в актуальном состоянии следует возложить на определенного человека или группу людей, которые являются владельцами процессов, связанных с обеспечением непрерывности бизнеса всей компании.

Как было отмечено ранее, поддержание актуальности плана должно быть внедрено в процедуры управления изменениями, чтобы все изменения в окружении сразу же отражались в плане.

Тестирование плана, учения и упражнения по восстановлению после аварий должны проводиться не реже одного раза в год. Компания не может доверять разработанному плану, пока он не был проверен на практике. Тестирование и учения готовят персонал к тому, с чем они столкнутся в аварийной ситуации, дают контролируемую среду для обучения сотрудников выполнению возложенных на них задач. Также, тестирование и учения позволяют команде и руководству выявить проблемы, которые были недостаточно продуманы или вообще не учтены в процессе планирования. Проводимые в конце упражнения на практике показывают, действительно ли компания сможет восстановиться

после аварии.

Упражнения должны проводиться по заранее подготовленным сценариям, с которыми компания может однажды столкнуться. Будет гораздо лучше, если основные процедуры, предусмотренные в планах, будут отработаны до того, как прозвучит сигнал тревоги. Команда тестирования должна согласовать, что именно будет тестироваться и по каким критериям нужно будет определить, прошло ли тестирование успешно или неудачно. Команда должна согласовать время и продолжительность упражнений, определить тех, кто будет принимать в них участие, кто будет получать задания и какие шаги он должен выполнять. Также, команда должна определить, что будет тестироваться и в какой комбинации: оборудование, программное обеспечение, персонал, процедуры, коммуникационные каналы и т.д. Если, например, в рамках упражнений предполагается перемещение некоторого оборудования на альтернативную площадку, должны быть учтены и проанализированы вопросы транспортировки, установки дополнительного оборудования на основную площадку, подготовки альтернативной площадки.

Большинство компаний не могут допустить, чтобы эти упражнения прерывали работу или снижали производительность компании, это может установить определенные ограничения для упражнений в части их объема или времени поведения, что может потребовать дополнительного логистического планирования. План проведения упражнений следует утвердить официально, в нем должно быть указано, какие слабые места в общем плане восстановления должны быть протестированы. При выполнении упражнений первые несколько раз, не следует привлекать к ним всех сотрудников компании, лучше задействовать в каждом случае различные небольшие группы людей, чтобы они могли изучить свои обязанности. После этого более широкомасштабные учения не окажут негативного воздействия на работу компании. Люди, участвующие в этих учениях, должны быть готовы к тому, что возникнут различные проблемы и ошибки. Собственно из-за этого и проводятся эти учения. Компания сможет узнать, что при выполнении определенных действий или процедур сотрудники часто совершают ошибки, что позволит ей внести необходимые изменения в план, либо дополнительно обучить сотрудников, чтобы они лучше выполняли свои задачи в реальной аварийной ситуации.

ПРИМЕЧАНИЕ. После аварии телефонная связь может оказаться недоступной. Поэтому компании следует предусмотреть альтернативные коммуникационные каналы, например, сотовые телефоны или радиации.

Может применяться несколько различных способов проведения учений, каждый из них имеет свои плюсы и минусы. В следующих разделах рассмотрены различные способы проведения учений.

Тестирование содержания плана

При выполнении тестирования содержания плана, подразделениям компании рассылаются копии плана ВСП для изучения и анализа. Руководитель каждого подразделения при рассмотрении плана может выявить, что что-то было упущено, что некоторые подходы следует изменить, а некоторые действия можно удалить без всяких последствий. Этот способ дает уверенность, что ничего не было забыто. Все полученные от подразделений замечания и предложения команда учитывает и вносит изменения в основной план.

Структурированное сквозное тестирование

При выполнении тестирования этим способом, представители каждого подразделения собираются вместе, чтобы пройти по плану и убедиться в его правильности. Группа проводит анализ целей и задач плана, обсуждает его границы и сделанные в плане допущения, анализирует структуру отчетности, оценивает результаты тестирования, подходы к поддержке плана, описание требований к процессу учений. Это позволяет ответственным лицам в компании убедиться, что восстановление работы компании после аварии будет выполнено эффективно и результативно, а также лучше понять, что в

аварийной ситуации ожидается лично от них.

Группа от начала до конца проходит по различным сценариям выполнения плана, чтобы убедиться, что ничего не упущено. Также, это позволяет повысить осведомленность членов команд по процедурам восстановления.

Тестирование с помощью моделирования

Этот тип тестирования требует значительно больше планирования и людей. При выполнении такого тестирования, все сотрудники, которые принимают участие в выполнении бизнес-процессов компании и функций поддержки, собираются вместе для практической отработки плана восстановления после аварий на основе заранее определенного сценария. Как и в предыдущих случаях, этот тип тестирования выполняется, чтобы убедиться, что ничего не было пропущено, и никакие угрозы не были забыты. Также, это работает как катализатор для повышения осведомленности вовлеченных в этот процесс людей.

В процессе учений используется только то, что будет доступно в случае реальной аварии (соответствующей выбранному сценарию), для придания учениям большей реалистичности. Моделирование продолжается до момента перемещения персонала и оборудования компании на альтернативную площадку и восстановления работы на ней.

Параллельное тестирование

Параллельное тестирование выполняется для того, чтобы убедиться, что определенные системы действительно могут работать на альтернативной площадке. Эти системы перемещаются на альтернативную площадку и включаются в работу. Результаты их работы на альтернативной площадке сравниваются с их обычной работой на основной площадке. Это позволяет увидеть, требуются ли какие-либо улучшения, изменения настроек или другие действия.

Тестирование с полным прерыванием

Этот тип тестирования оказывает самое глубокое воздействие на работу компании. Основная площадка реально отключается и работа продолжает выполняться на альтернативной площадке. Команда восстановления реально выполняет свои задачи по подготовке систем и среды на альтернативной площадке. Вся работа выполняется только на оборудовании альтернативной площадки.

Это полноценные учения, которые предусматривают серьезную работу по планированию и координации. Оно позволяет выявить множество недостатков в плане, которые будет необходимо исправить до того, как произойдет реальная авария. Тестирование с полным прерыванием следует проводить только после того, как все другие виды тестирования были успешно выполнены. Такое тестирование является самым рискованным, оно может оказать очень серьезное и разрушительное воздействие на бизнес, если ситуация выйдет из под контроля, поэтому необходимо получить разрешение высшего руководства на проведение такого тестирования.

Выбор наиболее эффективного способа проведения учений зависит от типа компании и ее целей. Каждая компания может использовать различные подходы и иметь уникальные особенности. Для более качественного планирования может потребоваться проведение специализированного обучения участвующих в нем сотрудников. Специализированное обучение в данном случае будет лучше обзорного курса. К тому же высококачественное обучение повысит заинтересованность сотрудников.

В процессе выполнения любого вида тестирования все существенные шаги и события должны быть задокументированы, оформлены в виде отчета и доведены до сведения руководства компании, чтобы оно имело достоверную информацию о результатах тестирования.

Другие типы учений

Сотрудники должны также пройти обучение по другим вопросам, помимо процедур восстановления в случае аварий, например, по вопросам оказания первой помощи пострадавшим, использования огнетушителя, способам управления толпой, процедурам коммуникации в чрезвычайных ситуациях, изучить маршруты эвакуации из здания и способы правильного выключения оборудования при авариях.

Как можно больше технических специалистов должны знать, как восстановить сетевые ресурсы, как переключить работу на резервные телекоммуникационные каналы, если основные выйдут из строя. Дополнительные источники питания должны быть проверены, а процедуры перевода критичных систем от одного источника питания к другому должны быть понятны техническим специалистам и протестированы ими.

Реагирование на чрезвычайную ситуацию

Часто первоначальная реакция на чрезвычайную ситуацию оказывает решающее влияние на конечный результат. Процедуры реакции на чрезвычайные ситуации – это заранее подготовленные планы действий, предназначенные для того, чтобы помочь людям преодолеть последствия аварии. Эти процедуры являются первой линией защиты в отношении кризисных ситуаций.

Если сотрудники постоянно поддерживают свои навыки по восстановлению после аварий, в кризисной ситуации они будут выполнять свою работу гораздо лучше, поэтому очень важно проводить различные учения и практические упражнения. Чрезвычайные происшествия непредсказуемы, никто не знает, когда они произойдут.

Защита жизни людей является наиболее важной задачей, она должна быть выполнена самой первой, а уже после нее можно думать о сохранении материальных ценностей. В процессе учений и упражнений ответственные за это люди должны понять, как безопасно эвакуировать персонал (см. Таблицу 7-3). Весь персонал должен знать о расположении аварийных выходов и пунктах сбора. Пункты сбора в случае аварий должны учитывать сезонные влияния погоды. В каждой группе следует назначить человека, который должен будет проконтролировать, что все люди, за которых он отвечает, покинули здание и благополучно добрались до пункта сбора. Другого человека следует назначить ответственным за уведомление соответствующих уполномоченных органов и служб: полицейского департамента, службы безопасности, пожарной части, скорой помощи и руководства. Прошедшие надлежащее обучение и практические занятия сотрудники, будут гораздо лучше готовы к действиям в чрезвычайной ситуации, а не просто побегут к выходу.

Если возникшая ситуация не представляет угрозы для жизни людей, следует позаботиться о том, чтобы надлежащим образом отключить системы, закрыть файлы с важными данными, а также вынести из здания ценные вещи сотрудников в процессе эвакуации, такие, как сумки, бумажники, одежду. Чтобы организовать это наиболее эффективно, требуется предварительное планирование и упорядочивание этих действий. Как и в других процессах, здесь существует зависимость последующих действий от сделанных ранее. Поспешное решение о пропуске отдельных шагов в действительности может нанести больше вреда, чем принести пользы.

Если в компании возникнет серьезная авария, вероятно потребуется один или несколько сотрудников для взаимодействия с внешними лицами и организациями, такими как пресса, клиенты, акционеры или представители общественности. Желательно, чтобы эти сотрудники имели заранее подготовленные заявления и ответы, которые в достаточном объеме, разумно и непротиворечиво объясняют сложившуюся ситуацию, предпринимаемые компанией действия, пояснения о том, что могут ожидать от компании ее клиенты, контрагенты и партнеры. Компании следует быстро сообщить эту информацию, чтобы не позволять другим делать собственные умозаключения и порождать фальшивые слухи. Как минимум один

человек должен быть постоянно доступен для прессы, чтобы гарантировать, что правильная информация и ответы на вопросы будут своевременно предоставлены.

Также, заранее следует учесть другие возможные неблагоприятные последствия чрезвычайной ситуации, такие как вероятное мародерство, вандализм, появление удобных возможностей для мошенничества и т.п. После того, как компания столкнется с масштабной аварией или чрезвычайной ситуацией, она становится очень уязвима, и некоторые могут попытаться воспользоваться этим. Поэтому следует продумать это заранее и запланировать соответствующие шаги, обеспечивающие приемлемый уровень защиты компании.

Процедура: Описание процесса эвакуации персонала	Место	Фамилии сотрудников, обученных по выполнению этой процедуры	Дата последних учений
На каждом этаже в здании должны быть назначены два сотрудника, которые обязаны убедиться, что весь персонал эвакуирован из здания после наступления чрезвычайной ситуации. В их обязанности также входит пересчет сотрудников, взаимодействие с Координатором ВСП, принятие решений о действиях в возникшей ситуации сотрудников, за которых они отвечают.	Автостоянка	Иванов И.И. Петров П.П.	05.01.2010
Комментарии: Назначенные сотрудники обязаны поддерживать в актуальном состоянии списки персонала на своем этаже. Они должны быть обеспечены корпоративной рацией, они должны пройти обучение и практические занятия по выполнению своих задач.			

Таблица 7-3. Пример процедуры реакции на чрезвычайную ситуацию

2.16. Поддержка плана

К сожалению, рассмотренные в этом Домене планы могут быстро потерять свою актуальность. Неактуальный план ВСП может дать компании ложное чувство безопасности, и приведет к массе сюрпризов, когда произойдет реальная авария.

Среди причин, по которым планы теряют свою актуальность, можно отметить следующие:

- Процесс обеспечения непрерывности бизнеса не интегрирован в процесс управления изменениями
- Произошли изменения инфраструктуры и окружения
- Реорганизация компании, увольнения, поглощения
- Изменения в аппаратном и программном обеспечении, приложениях
- После создания плана сотрудники решили, что на этом работа по обеспечению непрерывности бизнеса компании закончена
- Текучесть персонала
- Планам требуется большой объем работы для поддержки их актуальности
- Планы не имеют прямой связи с прибылью компании

Компания может сохранить актуальность плана, выполняя следующие действия:

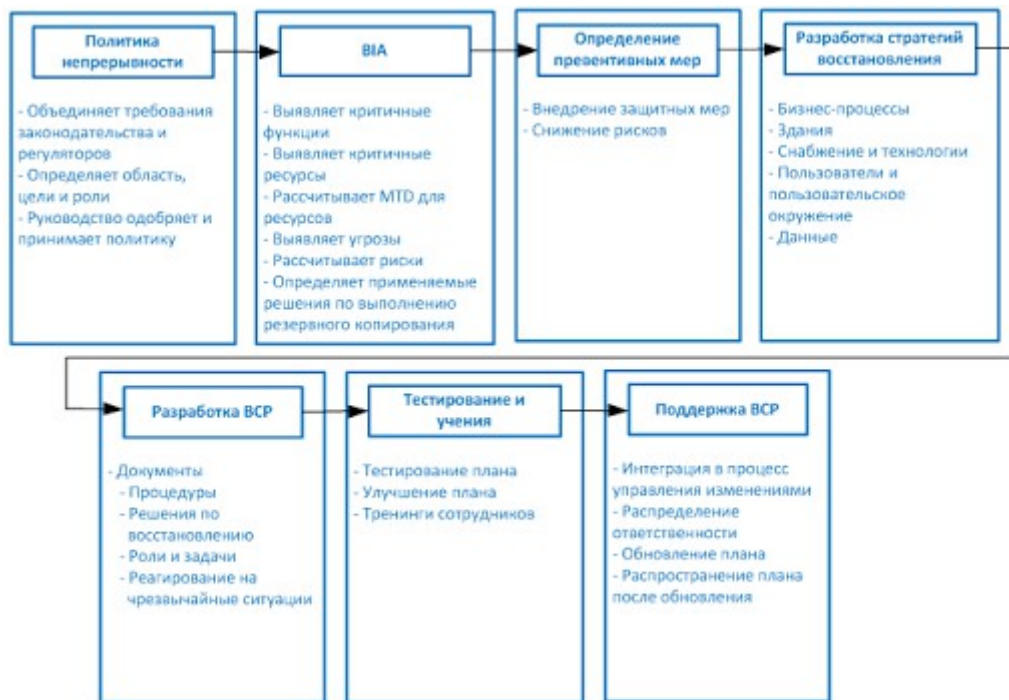
- Сделать обеспечение непрерывности бизнеса частью любого бизнес-решения
- Включить обязанности по поддержке плана в должностные инструкции
- Включить результаты поддержки плана в оценку работы персонала
- Выполнять внутренние аудиты, включающие проверку восстановления после аварий, документации по обеспечению непрерывности и соответствующих процедур

- Выполнять регулярные практические учения по выполнению плана
- Интегрировать вопросы непрерывности бизнеса в процесс управления изменениями компании

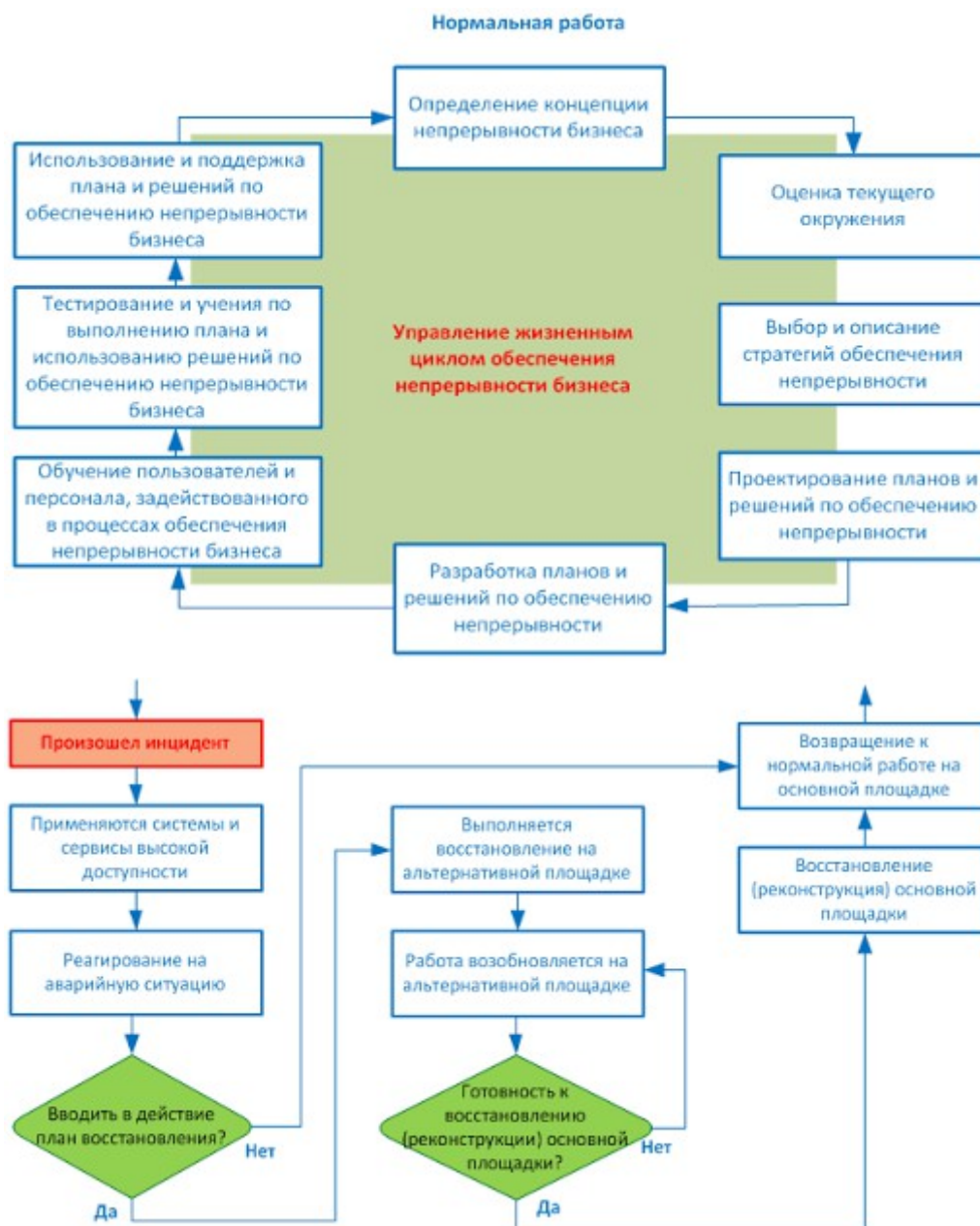
Одним из простейших и наиболее экономически эффективных и результативных способов сохранения актуальности плана, является его внедрение в процесс управления изменениями компании. Вы должны вспоминать о нем при вводе в эксплуатацию новых приложений, оборудования, сервисов, когда вы устанавливаете обновления и в других подобных случаях. Следует обновить процесс управления изменениями компании, чтобы в нем были учтены функции по уведомлению команды ВСП о планируемых изменениях, предусматривалось выделение ресурсов для обновления документации по восстановлению. Какой смысл ежегодно вытирать пыль с плана, если он разработан для систем, которые использовались в компании три года назад?

Ссылки по теме:

- Business Continuity Planning Model, Disaster Recovery Journal
- Disaster Prevention and Recovery Program of the Virginia Community College System



Жизненные циклы. Помните, что у планов DRP и ВСП есть свои жизненные циклы. Если компании нужны действительно работоспособные планы, она должна понимать это и поддерживать планы на каждом этапе их жизненного цикла.



3. Резюме

Хотя в настоящее время большинство компаний присваивает низкий приоритет задачам по планированию непрерывности бизнеса, это не означает, что эти задачи не важны. К сожалению, чтобы понять важность планирования непрерывности, во многих случаях компания должна пройти через серьезную аварию или чрезвычайную ситуацию. Тогда она поймет, насколько важно заранее предпринять шаги, чтобы избежать подобных случаев в будущем или снизить их последствия.

Чтобы разработка планов обеспечения непрерывности бизнеса оказалась успешной, она должна пройти через определенные этапы. Должны быть выявлены и поняты реальные угрозы, перед лицом которых стоит компания, должны быть продуманы и внедрены контрмеры для них, а если неприятность все же произойдет, должны быть разработаны соответствующие планы действий.

Тест

Вопросы экзамена CISSP являются концептуальными, поэтому они сформулированы соответствующим образом. Задачей кандидата является выбор наилучшего из всех представленных вариантов ответа. Среди вариантов ответа может не быть идеального ответа на поставленный вопрос - кандидат должен выбрать лучший ответ из имеющихся вариантов.

1. Какие процедуры должны быть выполнены для восстановления систем и данных после системного сбоя?

- ☐ A. Восстановление с резервных копий
- ☐ B. Проведение параллельного тестирования
- ☐ C. Выполнение процедур восстановления
- ☐ D. Выполнение сквозного тестирования

2. Что является одним из первых шагов при разработке плана обеспечения непрерывности бизнеса?

- ☐ A. Определение имеющихся средств резервного копирования
- ☐ B. Принятие решения, нужно ли компании выполнять свозное, параллельное тестирование или моделирование
- ☐ C. Проведение анализа воздействия на бизнес (BIA)
- ☐ D. Разработка плана возобновления бизнеса

3. Насколько часто следует тестировать план обеспечения непрерывности бизнеса?

- ☐ A. Не реже одного раза в десять лет
- ☐ B. Только после изменений инфраструктуры или окружения
- ☐ C. Не реже одного раза в два года
- ☐ D. Когда в компании происходят существенные изменения

4. Одним из важных шагов в процессе тестирования процедур восстановления является ведение записей обо всех существенных шагах и произошедших событиях. Что из перечисленного ниже является не менее важным?

- ☐ A. Спланировать следующее тестирование, при котором будут учтены возникшие проблемы
- ☐ B. Убедиться, что назначен человек, который готов ответить на вопросы прессы
- ☐ C. Подготовить отчет для руководства об этих шагах и событиях
- ☐ D. Определить наиболее важные бизнес-функции

5. Что из перечисленного ниже является наименее важным при проведении оценки рисков, связанных с потенциальными чрезвычайными ситуациями?

- ☐ A. Сбор информации из отчетов специальных агентств, которые позволяют оценить вероятность природных катаклизмов в определенной местности
- ☐ B. Идентификация ключевых функций компании и требований бизнеса
- ☐ C. Идентификация критичных систем, обеспечивающих работу компании
- ☐ D. Оценка потенциальных потерь и негативного воздействия на компанию в зависимости от продолжительности простоя

6. Действия, выполняемые сразу после возникновения чрезвычайной ситуации, должны быть направлены на предотвращение человеческих жертв и вреда здоровью людей, а также на _____.

- ☐ A. Обеспечение защиты от мошенничества и мародерства
- ☐ B. Минимизацию дальнейших повреждений
- ☐ C. Защиту доказательств и улик
- ☐ D. Оценку масштабов повреждений

7. Что из перечисленного ниже является наилучшим способом обеспечения гарантированной возможности восстановления данных с резервных лент и их использования на «теплой» площадке?

- ☐ A. Взять ленты с внешней площадки и проверить их работу на оборудовании основной площадки
- ☐ B. Попросить поставщика внешней площадки протестировать их и пометить те, которые прочитались
- ☐ C. Протестировать их на системе поставщика, которую не планируется использовать в случае аварии
- ☐ D. Дважды в месяц составлять опись лент, хранящихся на площадке поставщика

8. Что из перечисленного ниже лучше всего описывает отличия «горячей» площадки от «теплой» или «холодной»?

- ☐ A. Это площадка, на которой установлены жесткие диски, контроллеры и ленточные приводы
- ☐ B. Это площадка, на которой установлены все необходимые компьютеры, серверы и телекоммуникационные системы
- ☐ C. Это площадка, на которой проложена электрическая проводка, установлена централизованная система кондиционирования воздуха и фальшполы
- ☐ D. Это мобильная площадка, которая может стоять на парковке возле здания компании

9. Что из перечисленного ниже лучше всего описывает создание удаленных журналов (remote journaling)?

- ☐ А. Резервное копирование больших объемов данных на внешнюю площадку
- ☐ В. Резервное копирование журнала транзакций на удаленную площадку
- ☐ С. Одновременная запись транзакций на два зеркальных сервера, установленных на основной площадке
- ☐ D. Сохранение транзакций на носителе информации другого типа

10. Что из перечисленного ниже требуется для внешней площадки, на которой хранятся носители информации с резервными копиями данных компании?

- ☐ А. Площадка должна находиться в 10-15 минутах езды от основной площадки, чтобы резервные копии были легкодоступны
- ☐ В. На площадке должны быть установлены все необходимые компьютеры и серверы, смонтирован фальшпол
- ☐ С. Площадка должна охраняться вооруженной охраной
- ☐ D. Площадка должна быть защищена от несанкционированного доступа

11. Что из перечисленного ниже не может быть выявлено при проведении анализа воздействия на бизнес (BIA)?

- ☐ А. Подходит ли компании параллельное тестирование или тестирование с полным прерыванием
- ☐ В. Какая область может наиболее пострадать с функциональной и финансовой точки зрения в при аварии или чрезвычайной ситуации
- ☐ С. Какие системы наиболее критичны для компании и должны быть максимально защищены
- ☐ D. Какая продолжительность простоя приемлема для компании и не окажет катастрофического воздействия на ее бизнес

12. Для каких областей компании рекомендуется подготовка плана восстановления?

- ☐ А. Наиболее важных функциональных и финансовых областей
- ☐ В. Областей, в которых находятся критичные системы
- ☐ С. Всех областей
- ☐ D. Областей, без которых компания не сможет «выжить»

13. Кто утверждает план обеспечения непрерывности бизнеса?

- ☐ А. Комитет по планированию
- ☐ В. Руководитель каждого подразделения
- ☐ С. Руководство
- ☐ D. Внешнее лицо

14. Что является самым важным при разработке плана обеспечения непрерывности бизнеса?

- ☐ А. Анализ воздействия на бизнес
- ☐ В. Внедрение, тестирование и дальнейшее следование плану
- ☐ С. Участие всех и каждого подразделений компании
- ☐ D. Поддержка руководства

15. В процессе разработки, тестирования и поддержки плана обеспечения непрерывности бизнеса крайне важно обеспечить надлежащее взаимодействие и коммуникации. Почему?

- ☐ А. Это является одним из требований регуляторов к этому процессу
- ☐ В. Чем больше людей говорят об этом плане и участвуют в его создании, тем выше уровень осведомленности о нем
- ☐ С. Это не важно для разработки плана, тем более что такое взаимодействие будет отрывать людей от важной работы и может нанести ущерб производительности работы компании
- ☐ D. Руководство вероятно поддержит это

16. Что из перечисленного ниже описывает параллельное тестирование?

- ☐ А. Оно проводится, чтобы убедиться, что определенные системы могут работать на альтернативной площадке
- ☐ В. Все подразделения получают копию плана восстановления после аварий и «проходят» по нему
- ☐ С. Представители от каждого подразделения собираются вместе и совместно проводят это тестирование
- ☐ D. Выполнение операций в обычном режиме прекращается

17. Что из перечисленного ниже описывает структурированное сквозное тестирование?

- ☐ А. Оно проводится, чтобы убедиться, что критичные системы могут работать на альтернативной площадке
- ☐ В. Все подразделения получают копию плана восстановления после аварий и «проходят» по нему
- ☐ С. Представители от каждого подразделения собираются вместе и совместно проводят это тестирование
- ☐ D. Выполнение операций в обычном режиме прекращается

18. Когда компания может считать, что чрезвычайная ситуация закончилась?

- ☐ A. Когда люди пересчитаны и находятся в безопасности
- ☐ B. Когда выполнение всех операций и весь персонал возвращены на основную площадку
- ☐ C. Когда выполнение операций перенесено на альтернативную внешнюю площадку
- ☐ D. Когда официальные лица объявили об этом

19. Что из перечисленного ниже не имеет отношения к соглашению о взаимной помощи (reciprocal agreement)?

- ☐ A. Соглашение имеет юридическую силу
- ☐ B. Это дешевое решение
- ☐ C. Оно может быть использовано сразу после аварии
- ☐ D. Может оказаться очень сложным реализовать это на площадке, на которой уже обрабатываются данные другой компании

20. Что из перечисленного ниже является описанием «холодной» площадки?

- ☐ A. Она полностью оборудована и готова к работе компании на ней уже через несколько часов
- ☐ B. Она частично оборудована средствами обработки данных
- ☐ C. Она полностью настроена, но это дорогое решение
- ☐ D. На ней обеспечены только самые необходимые возможности, оборудования на ней нет

21. В каком из приведенных ниже пунктов наиболее полно перечислены компоненты плана восстановления после аварий?

- ☐ A. Оборудование, программное обеспечение, люди, аварийные процедуры, процедуры восстановления
- ☐ B. Люди, оборудование, внешняя альтернативная площадка
- ☐ C. Программное обеспечение, взаимодействие устройств, люди, оборудование, вопросы, связанные с руководством
- ☐ D. Оборудование, аварийные процедуры, программное обеспечение, идентифицированные риски

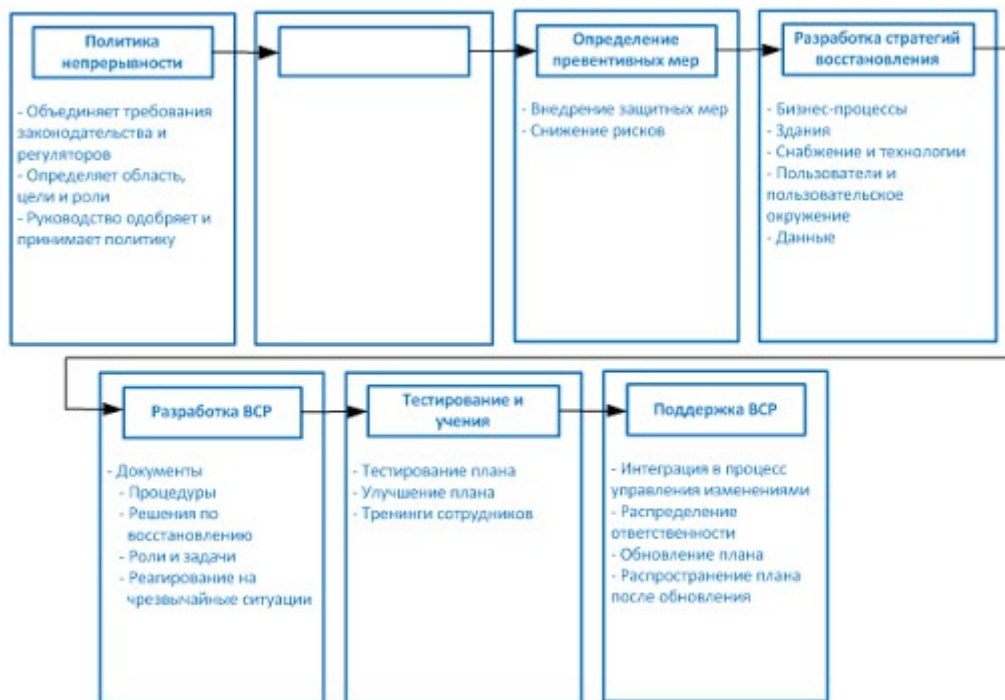
22. Что из перечисленного ниже не является преимуществом «горячей» площадки?

- ☐ A. Предлагает широкий выбор оборудования и программного обеспечения
- ☐ B. Постоянно доступна
- ☐ C. Может быть запущена в работу всего за несколько часов
- ☐ D. Возможно проведение ежегодного тестирования

23. Планы восстановления после аварий могут поддерживаться в актуальном состоянии при выполнении ряда условий. Что из перечисленного ниже не является таким условием?

- ☐ A. Сделать восстановление после аварий частью любого бизнес-решения
- ☐ B. Внести это в должностные инструкции сотрудников
- ☐ C. Регулярно проводить учения по использованию плана
- ☐ D. Сделать копии плана и хранить их на внешней площадке

24. Какой шаг пропущен на приведенной ниже схеме?



- ☐ A. Анализ воздействия на бизнес
☐ B. Стандарт NIST
☐ C. Утверждение руководством и распределение ресурсов
☐ D. Управление изменениями

25. Как называется процесс, элемент которого приведен на изображении ниже?

Из перечисленных ниже пунктов выберите тот, который лучше всего описывает последствия прерывания работы бизнес-процесса на указанное время
<input type="radio"/> Прерывание на 8 часов – последствия катастрофические <input type="radio"/> Прерывание на 24 часа – последствия критические <input type="radio"/> Прерывание на 3 дня – последствия значительные <input type="radio"/> Прерывание на 5 дней – последствия существенные <input type="radio"/> Прерывание на 10 дней – последствия не критичные <input type="radio"/> Прерывание на 30 дней – последствия не существенные
Укажите Целевое время восстановления (RTO) работы бизнес-процесса
Целевое время восстановления _____

- ☐ A. Анализ воздействия на бизнес
☐ B. Определение значений фаз активации
☐ C. Определение максимально допустимого времени простоя
☐ D. Связь времени реконструкции и стоимости

Домен 08. Законодательство, требования, соответствие, расследования.

Компьютерные преступления являются вполне естественным явлением, реакцией преступников на появившиеся новые возможности, а также зависимость современного общества от технологий. Преступность существовала во все времена. Компьютер просто стал еще одним инструментом, который, также как и другие инструменты, может быть использован и для добра, и для зла.

Мошенничество, кражи, хищения всегда были частью жизни общества, но компьютерный век принес новые возможности для воров и мошенников. Преступники начали использовать Интернет для шантажа и финансового мошенничества. Существенно усложнилось ведение учета, отчетности, проведение денежных переводов и т.п., что повлекло за собой появление новых уязвимостей в этих процессах, которыми стали пользоваться преступники.

Киберпреступники шантажируют компании, обнаруживая уязвимости в их сетях. Они занимаются хищением коммерческой тайны компаний через «дыры» в их системах безопасности. Растет мошенничество в системах Интернет-банкинга, проводятся атаки компьютерных сетей розничных магазинов с целью хищения баз данных с информацией банковских карт клиентов. Быстрыми темпами растет направление, связанное с кражей персональных данных.

В современном мире прочно закрепились системы электронной коммерции, интернет-магазины и т.п., что также представляет собой постоянно растущую угрозу. Стремительно растет количество взломов и атак, компании хорошо это знают. А правовые системы и правоохранительные органы пока сильно отстают, не имея возможностей эффективно и оперативно находить компьютерных злоумышленников и успешно привлекать их к ответственности. В мире постоянно разрабатываются новые технологии для борьбы с различными видами атак, но еще большая потребность существует в части разработки актуальных законов, политик и методов, позволяющих реально ловить преступников и заставлять их возмещать ущерб, который они наносят. В этом Домене рассматриваются некоторые из этих вопросов.

1. Многогранное киберправо

Правовые вопросы очень важны для компаний, т.к. нарушение компанией своих обязательств и предъявляемых к ней требований законодательства может нанести значительный ущерб бизнесу компании и ее репутации. У любой компании есть множество этических и правовых обязанностей, за которые она несет ответственность, в том числе и в отношении компьютерного мошенничества. Чем лучше компания понимает свои обязанности, тем проще ей соблюдать их и не выходить за рамки дозволенного.

Законодательство и регуляторы могут устанавливать требования по вопросам управления инцидентами, защиты конфиденциальных данных, компьютерных злоупотреблений, сохранения доказательств, этического поведения, ожидаемого от компании, ее руководства и персонала. Мы живем в очень интересное время для права и технологий. Законодатели, судьи, правоохранительные органы и юристы находятся в крайне затруднительном положении из-за неспособности идти в ногу со стремительными изменениями технологий в компьютерном мире и сложности этих вопросов. Правоохранительные органы должны знать, как им искать злоумышленника, как его ловить, собирать и контролировать доказательства, как использовать доказательства представителями обвинения и защиты. Обе эти стороны должны понять, что действительно произошло в компьютерном преступлении, как оно было осуществлено, какие законы и прецеденты можно использовать, чтобы доказать в суде свою точку зрения. Технологии и связанные с ними термины и понятия часто вызывают сложности у судей и присяжных, а новые законы разрабатываются недостаточно быстро, что не всегда позволяет доказать вину киберпреступников и надлежащим образом наказать их.

Хотя в 21 веке правоохранительные органы, правовые и судебные системы также начали развиваться и приспосабливаться к новым технологиям.

Многие компании имеют офисы в различных городах и странах. Это приводит к еще большим проблемам, когда встает вопрос, каким законам нужно следовать. В разных регионах и городах может действовать местное законодательство, существенно отличающееся от других. Одни и те же действия в разных странах могут оцениваться по-разному, в одной стране некое действие может не считаться преступлением, а в другой – то же самое действие может наказываться пятилетним заключением в тюрьме. Например, если злоумышленник из другой страны украл у американских банков большое количество номеров банковских карт и был пойман, американский суд хотел бы преследовать его по американским законам. Однако на его родине это может вообще не считаться незаконным. Границы стран не ограничивают действия злоумышленников, но они часто ограничивают действие законов.

Несмотря на все эти сложности, у компаний есть определенные обязательства, относящиеся к вопросам компьютерной безопасности и определяющие, в частности, как компания будет предотвращать, обнаруживать и сообщать о преступлениях.

2. Проблемы киберправа

Законы, связанные с компьютерной преступностью (иногда называемые *киберправом* (cyberlaw)), во всем мире имеют дело с рядом ключевых вопросов: несанкционированная модификация или уничтожение информации, разглашение критичной информации, несанкционированный доступ, использование вредоносных программ (malware, malicious software).

Хотя обычно мы думаем только о жертвах и их системах, подвергшихся атаке, были разработаны законы для борьбы с тремя категориями преступлений. **Преступления, совершенные с помощью компьютера** (computer-assisted crime), при совершении которых компьютер используется просто как инструмент, помогающий преступнику осуществить свои замыслы. **Преступлениями, направленными на компьютер** (computer-targeted crime), являются преступления, в которых сам компьютер становится «жертвой» нападения, организованного непосредственно для нанесения ему ущерба (и, соответственно, его владельцам). В преступлениях третьего типа, компьютер не обязательно используется для совершения преступления или является «жертвой», он просто участвует в процессе совершения преступления. Это называется **преступлениями с побочным использованием компьютера** (computer is incidental).

Ниже приведены некоторые примеры преступлений, совершаемых с помощью компьютера:

- Атака на финансовые системы с целью хищения денежных средств и / или конфиденциальной информации
- Получение военных и разведывательных данных с помощью атаки на военные системы
- Проведение промышленного шпионажа с помощью атак на компьютерные системы конкурентов и сбора конфиденциальных данных конкурентов
- Проведение информационных войн, в рамках которых производятся атаки на важнейшие системы государственной инфраструктуры
- Проведение акций протеста против правительства или деятельности компании, с помощью атак на их системы, выведения из строя их веб-сайтов или изменения информации на них

Теперь рассмотрим примеры преступлений, направленных на компьютеры:

- DDoS-атаки
- Перехват паролей и других критичных данных
- Установка вредоносных программ с целью причинить вред
- Установка руткитов и снифферов с целью причинить вред
- Выполнение атаки переполнения буфера с целью получения контроля над системой

Иногда возникают сложности с отнесением преступления к категории «совершенного с помощью компьютера» или «направленного на компьютер», поскольку интуитивно кажется, что любая атака одновременно попадает в обе эти категории - один компьютер используется для проведения атаки, а другой подвергается этой атаке. Отличие между этими категориями заключается в том, что «преступления, совершенные с помощью компьютера», используют компьютер исключительно в качестве инструмента для совершения традиционных видов преступлений. Даже без компьютера, люди по-прежнему могут совершать кражи, что-то ломать, устраивать акции протеста, похищать коммерческую тайну компаний и т.п. При совершении таких преступлений компьютер может использоваться, как один из инструментов злоумышленника. «Преступления, направленные на компьютер» не могут быть совершены без компьютера. Таких преступлений не существовало до того момента, когда компьютеры стали использоваться повсеместно. Иными словами, в старые добрые времена вы не могли выполнить атаку переполнения буфера на вашего соседа, или установить вредоносную программу вашему противнику. «Преступления, направленные на компьютер» требуют участия компьютера.

К категории «преступлений с побочным использованием компьютера» относятся преступления, к совершению которых причастен компьютер, но его участие в них остается незначительным и второстепенным. Например, ваш друг, которого уволили из компании, занимающейся лотереей, сообщил вам три ближайших выигрышных номера, а вы ввели их в свой компьютер, чтобы не забыть. Ваш компьютер является только местом хранения. Вы могли бы записать эти номера и на листе бумаги, без всякого компьютера. Другим примером является детская порнография – преступлением является получение и распространение детской порнографии, графических изображений. Эти изображения могут быть сохранены на компьютере, а могут быть напечатаны на бумаге. В преступлениях этой категории компьютер не используется для атак на другие компьютеры и не подвергается атакам сам, он просто используется некоторым образом.

Вы можете сказать: «Ну и что? Преступления в любом случае остаются преступлениями. Зачем делить их на какие-то категории?». Это нужно для того, чтобы применять к компьютерным преступлениям существующие законы. Скажем, кто-то просто включил ваш компьютер и заглянул в несколько папок с вашими файлами, не нанеся никакого ущерба, но вы не давали ему такого разрешения. Нужно ли разрабатывать новый закон, указывая в нем запрет на использование чужих компьютеров? Или лучше просто воспользоваться существующим законом? Что делать, если хакеры взломали систему управления светофорами в городе и получили возможность включить на всех светофорах одновременно зеленый свет? Нужно ли и для этой цели создавать новые законы или тут тоже можно воспользоваться существующими и понятными законами, по которым наработана обширная практика? Помните, что любое преступление является преступлением, а компьютер является просто новым инструментом.

Использование уже существующих законов облегчает работу судей, которые могут выносить приговоры, основываясь на хорошем знании законов, направленных на соответствующие преступления. Правительство разработало руководящие документы по вынесению приговоров, чтобы стандартизировать наказания для однотипных преступлений во всех судах. Для этого они используют систему баллов. Например, если вы похитили кого-нибудь, вы получаете 10 баллов. Если для этого вы еще и перешли через государственную границу,

вы получаете дополнительные 2 балла. Если вы ударили похищенного человека, вы получаете еще 4 балла. Чем больше баллов, тем более суровым будет наказание.

Аналогично, если вы украли деньги с чужого счета, атаковав сервер банка, вы можете получить 5 баллов. Если вы использовали эти деньги для поддержки террористической группы, вы получаете еще 5 баллов. А за то, что вы не указали эти доходы в своей налоговой декларации, вам дополнительных баллов не начислят.

Конечно, это не означает, что с любым компьютерным преступлением можно бороться с помощью уже существующих законов. Многие страны были вынуждены разрабатывать новые законы, которые непосредственно касаются различных видов компьютерных преступлений. Ниже приведены для примера некоторые из законов США, которые были разработаны или доработаны для учета различных видов компьютерных преступлений:

- Параграф 1029 Титула 18 Свода законов США: Мошенничество, связанное с устройствами доступа (*предусмотрена ответственность за торговлю похищенными или поддельными устройствами доступа, которые могут быть использованы для получения денег, товаров или услуг*)
- Параграф 1030 Титула 18 Свода законов США: Мошенничество, связанное с компьютерами (*предусмотрена ответственность за посяательства на «защищенный компьютер» (принадлежащий правительству или финансовой организации) и находящуюся на нем компьютерную информацию*)
- Параграф 2510 Титула 18 Свода законов США: Перехват проводных и электронных коммуникаций, голосовой информации (*предусмотрена ответственность за несанкционированный доступ к передаваемой информации*)
- Параграф 2701 Титула 18 Свода законов США: Хранение данных проводных и электронных коммуникаций, журналов доступа (*определен порядок хранения и использования журналов доступа и сохраняемых данных*)
- Закон об авторском праве в цифровом тысячелетии (*предусмотрена ответственность за нарушение авторских прав путем копирования, производство и распространение технологий, позволяющих обходить технические средства защиты авторских прав*)
- Акт о дополнительных мерах по компьютерной безопасности (*ужесточает наказания за киберпреступления*)

ПРИМЕЧАНИЕ. Для сдачи экзамена CISSP не нужно знать все эти законы, это просто примеры.

3. Сложности борьбы с киберпреступностью

Раз у нас есть столько законов, относящихся к цифровым преступникам, означает ли это, что мы держим под контролем всю киберпреступность? Увы, количество хакерских атак и взломов постоянно росло на протяжении многих лет и не будет снижаться в ближайшее время. Возникают различные сложности при попытке остановить или хотя бы сдержать злоумышленников. К ним относятся сложности при идентификации и поиске злоумышленников, обеспечение необходимого уровня защиты сети и успешное судебное преследование злоумышленников после их поимки.

Большинство злоумышленников так и не находят, поскольку они скрывают (используют ложные) адреса и идентификаторы, применяют методы скрытия своих действий. Многие атакующие взламывают сети, получают несанкционированный доступ к любым ресурсам в них, а затем очищают лог-файлы, с помощью которых можно было бы отследить их действия. Из-за этого, многие компании даже не знают, что их сети были взломаны. Даже если система выявления вторжений (IDS) смогла выявить действия злоумышленника, это все равно, как правило, не позволяет определить настоящую личность нападающего, хотя это и

позволяет компании узнать о факте атаки, а также об использовании конкретной уязвимости.

Злоумышленники при доступе к системе жертвы обычно используют несколько промежуточных систем между своим компьютером и компьютером жертвы, что существенно осложняет процесс их отслеживания, для чего нужно пройти по всей цепочке промежуточных систем в обратной последовательности, начиная с системы жертвы. Преступники часто используют компьютеры обычных людей для выполнения с них своих преступных действий. Для этого злоумышленник устанавливает вредоносную программу на компьютере такого человека, используя различные методы: отправку вредоносного кода во вложении к сообщению электронной почты, размещение троянской программы на веб-сайте и т.п. В большинстве случаев процесс загрузки, запуска и дальнейшей работы вредоносной программы остается абсолютно незаметен для пользователя, вредоносная программа не производит никаких действий, пока не получит команду от злоумышленника – например, команду на проведение атаки на другую систему. Такие скомпрометированные злоумышленниками системы называются **зомби** (zombie), установленное на них вредоносное программное обеспечение называется **ботом** (bot), а если у злоумышленника есть несколько таких скомпрометированных систем, подключенных к сети Интернет, это называется **бот-сетью** (botnet). Бот-сеть может использоваться для проведения распределенных DoS-атак, рассылки спама – любых команд злоумышленника. Более детально эти вопросы рассмотрены в Домене 09, здесь мы говорим о них только для того, чтобы проиллюстрировать, как нападающие могут легко скрыть свою личность.

Расследованием компьютерных преступлений занимаются правоохранительные органы. Они обучают своих сотрудников для нахождения и задержания компьютерных преступников, но пока правоохранительные органы далеко позади, они не обладают теми навыками и инструментами, которые есть у хакеров. Атакующие используют автоматизированные инструменты, они за короткое время могут провести несколько серьезных атак. Правоохранительные органы, расследуя эти атаки, работают практически «вручную», проверка журналов регистрации событий, опрос людей, исследования носителей информации, сканирование систем на наличие уязвимостей, создание ловушек на случай, если атакующий вернется. В каждом государственном правоохранительном ведомстве есть лишь небольшое число сотрудников, способных заниматься расследованием компьютерных преступлений, но даже их опыт и знания обычно отстают от опыта и знаний хакеров. Все это приводит к тому, что большинство злоумышленников так и не находят, а из тех, кого находят, очень немногих привлекают к реальной ответственности.

Это никоим образом не означает, что все компьютерные преступники уходят от наказания. Правоохранительные органы постоянно совершенствуют свою тактику, обучают своих сотрудников. С информацией о компьютерных преступлениях в США можно ознакомиться на сайте www.cybercrime.gov.

В действительности лишь немногие законы специально направлены на компьютерные преступления, что дополнительно усложняет проведение расследований в отношении пойманных злоумышленников. Многие пострадавшие от хакерских атак компании, хотят просто исправить уязвимость, которой воспользовался злоумышленник, а не тратить время и деньги на его преследование и наказание. Это является одной из главных причин ухода злоумышленников от ответственности. Большинство компаний не сообщают о компьютерных преступлениях (за исключением некоторых, в основном тех, которые обязаны сообщать об этом в соответствии с требованиями законодательства). Ни одна компания не хочет, чтобы о ее проблемах знали все, поскольку это может подорвать доверие клиентов, акционеров и инвесторов. Даже те компании, которые по закону обязаны сообщать о нарушениях безопасности и компьютерных преступлениях, не всегда делают это. Часто они просто исправляют уязвимость и делают вид, что никакого инцидента не было. Все это приводит к тому, что в мире отсутствует даже достоверная статистика компьютерных преступлений.

Законодательство, требования регуляторов, различные стандарты и лучшие практики позволяют руководителям компаний быть осведомленными в вопросах безопасности, однако они не всегда считают эти вопросы действительно важными и не предпринимают достаточных усилий для обеспечения надлежащего уровня безопасности своих компаний. Зачастую, вопросы безопасности становятся очень важны для компании только после того, как ее имя попадает в заголовки газет, описывающих, как у компании украли сто тысяч номеров банковских карт.

3.1. Электронные активы

Другой сложностью, которую принес в общество цифровой мир, является определение того, что должно быть защищено, и необходимого уровня защиты. Изменились сами активы, которые требуется защищать компаниям. Еще пятнадцать лет назад такими активами были материальные вещи (оборудование, сооружения, инструменты и т.п.). Теперь компании должны добавить в этот список данные, причем данные, как правило, находятся на самом верху этого списка: проекты новых продуктов, номера банковских карт, медицинская информация, персональные данные, коммерческая тайна, военные стратегии и т.д. Хотя военным всегда приходилось заботиться о защите своих тайн, они никогда не работали с таким большим количеством «точек входа» в свои секреты, каждую из которых необходимо контролировать. В настоящее время компании испытывают трудности не только при защите своих электронных данных, но и при определении того, что именно является конфиденциальной информацией и где она должна храниться.

ПРИМЕЧАНИЕ. В законодательстве многих странах для более эффективной борьбы с компьютерными преступлениями, определение собственности было расширено и в него были включены данные.

Многие компании с удивлением узнали, что защита нематериальных активов (таких как данные и репутация) является намного более сложной задачей, чем защита материальных активов.

3.2. Эволюция атак

Около десяти лет назад хакерами были люди, которым просто доставляло удовольствие исследовать и взламывать системы. Они смотрели на это, как на сложную игру, они не имели намерений причинить вред. Чтобы попасть в заголовки газет, хакеры взламывали и нарушали работу крупных веб-сайтов (Yahoo, MSN, Excite) и хвастались этим перед своими коллегами. Вирусописатели создавали вирусы, которые просто распространялись и выполняли на зараженных компьютерах действия, которые сложно назвать вредоносными (в основном различные шутки). В настоящее время, к сожалению, все изменилось.

Хотя и сегодня существуют хакеры-любители и люди, которые взламывают системы для забавы, появилась организованная компьютерная преступность, что значительно увеличило суммы ущерба от деятельности хакеров. В прошлом, скрипт-кидди (script kiddy) сканировали тысячи и тысячи систем в поисках определенной уязвимости, которой они могли воспользоваться. Для них не было никакой разницы, находилась ли эта система в корпоративной сети, в сети правительственной организации или являлась домашним компьютером пользователя. Такой хакер хотел воспользоваться уязвимостью просто для развлечения, «поиграть» с системой и сетью, к которой она подключена. В отличие от него, современные хакеры делают это максимально скрытно и не стараются привлекать к себе никакого внимания. Они являются членами организованных преступных групп и выполняют конкретные и целенаправленные задачи, обычно с целью получения прибыли. Например, они могут перехватывать номера банковских карт, персональные данные людей, чтобы затем использовать их для проведения мошеннических операций.

ПРИМЕЧАНИЕ. Скрипт-кидди – это хакеры, которые не имеют навыков для проведения атак без помощи специальных инструментов, существенно упрощающих их задачи. Они не всегда понимают, как в действительности происходит атака, поэтому они часто не догадываются об

уровне ущерба, который они могут причинить своими действиями.

Наиболее часто применяемые схемы интернет-мошенничества

- Мошенничество на аукционах
- Поддельные банковские чеки
- Ликвидация долгов
- Рассылки по электронной почте
- Трудоустройство / деловое сотрудничество
- Мошенничество при выполнении посреднических услуг
- Мошенничество с инвестициями
- Лотереи
- «Нигерийские письма»
- Финансовые пирамиды (схема Понци)
- Перепродажа
- Получение средств третьей стороной

Чтобы подробнее узнать обо всех этих видах компьютерных преступлений, посетите сайт www.ic3.gov/crimeschemes.aspx.

В настоящее время значительно снизилось и продолжает снижаться количество компьютерных вирусов, созданных для развлечения и не наносящих существенного ущерба. Зато постоянно растет количество очень опасных вредоносных программ. Они стали более целенаправлены. Чаще всего они предназначены для установки на компьютеры жертв бэкдоров, ботов и руткитов. Продолжает увеличиваться изощренность атак и растет ущерб от них.

Инсайдеры. Злоумышленнику, как правило, нужен доступ к системам, содержащим ценные ресурсы, а инсайдерам (внутренним злоумышленникам, работающим в компании) значительно проще, чем внешним нарушителям, получить доступ к защищаемым ресурсам компании. У сотрудников компании гораздо больше возможностей для совершения компьютерных преступлений, чем у внешних нарушителей. Многие специалисты действительно отмечают, что инсайдеры виновны в большинстве компьютерных преступлений, но средства массовой информации обычно продвигают только рассказы о внешних хакерах, поскольку они интереснее читателям. Это привело к тому, что угрозы хакерства часто переоценивают, и при этом уделяют недостаточно внимания угрозам, исходящим от собственного персонала, пользующегося своим положением и наличием разрешенного доступа к компьютерным системам компании.

Итак, мы перечислили некоторые сложности, возникающие при борьбе с киберпреступностью: анонимность, которую предоставляет злоумышленнику сеть Интернет; появление организованной компьютерной преступности; повышение сложности и изощренности атак; недостатки и отставание правовой системы; отсутствовавшее до недавнего времени внимание компаний к защите своих информационных активов. Дополнительные сложности возникают при проведении хакерских атак через границы различных стран.

3.3. Трансграничные преступления

Если хакер, находящийся на Украине, провел атаку на банк во Франции, в чьей правовой юрисдикции будет рассматриваться это преступление? Как эти страны будут осуществлять совместную работу для поимки преступника и осуществления правосудия? В какой стране будет происходить суд над этим человеком? На многие из этих вопросов пока нет ответов.

Когда компьютерное преступление пересекает государственные границы, сложность его расследования многократно повышается, а вероятность привлечения злоумышленника к суду – уменьшается. Это может быть связано с отличиями правовых систем разных стран,

отсутствием в некоторых странах законов, касающихся компьютерных преступлений, вопросами юрисдикции, а также нежеланием отдельных стран помогать друг другу. Представьте, что иранский хакер взломал систему в Израиле. Каковы шансы, что иранское правительство поможет израильским правоохранительным органам выследить злоумышленника? Может даже оказаться, что атака вообще была осуществлена по распоряжению правительства.

Были предприняты определенные усилия для унификации подхода различных стран к компьютерным преступлениям, поскольку компьютерные преступления не знают государственных границ. Злоумышленнику из Китая не составляет труда передать через Интернет сетевые пакеты в банк, расположенный в Саудовской Аравии, но крайне сложно (из-за особенностей правовых систем, различий в культуре и политике) побудить правительства этих стран к совместной работе.

Примером попытки создания международного стандарта реагирования на киберпреступления является **Конвенция по киберпреступности Совета Европы**. Фактически, это первое международное соглашение, направленное на гармонизацию национального законодательства различных стран, предоставляющее методики расследования преступлений и совместную работу на международном уровне.

Расположенные в различных странах офисы международных компаний постоянно взаимодействуют между собой с помощью электронной почты, телефонной и спутниковой связи, волоконно-оптических каналов и т.п. Для таких компаний очень важно проанализировать законодательство стран, в которых они работают, связанное с информационными потоками и вопросами конфиденциальности.

При передаче данных через границы различных стран, компании должны учитывать требования **Руководящих принципов Организации экономического сотрудничества и развития (ОЭСР) для трансграничных потоков информации и правила трансграничной передачи данных**, которые были рассмотрены в Домене 01. Каждая страна имеет собственные, отличные от других, законы, устанавливающие различные требования, использующие различные категории информации. Все это еще больше усложняет ведение международного бизнеса. ОЭСР является международной организацией, которая помогает правительствам различных стран собраться вместе и решить экономические, социальные и управленческие проблемы в условиях глобализации экономики. Для этих целей ОЭСР выпустила руководство, которому должны следовать различные страны, чтобы обеспечить надлежащую защиту информации, применяя при этом одни и те же правила.

ОЭСР определила семь ключевых принципов:

- Сбор персональных данных должен быть ограничен, осуществляться законными и честными способами, с согласия субъекта персональных данных.
- При хранении персональных данных должна обеспечиваться полнота и актуальность персональных данных, хранение должно соответствовать заявленным целям.
- Субъекты персональных данных должны быть заранее уведомлены о причинах сбора их персональных данных, компании должны использовать персональные данные только в заявленных целях.
- Раскрытие, предоставление доступа или использование персональных данных в целях, отличных от первоначально заявленных, должно осуществляться только с разрешения субъекта персональных данных, либо на основании требований законодательства.
- Должны быть внедрены разумные защитные меры для защиты персональных данных от таких рисков, как их утрата, несанкционированный доступ к ним, их несанкционированное изменение или раскрытие.

- Порядки использования, политики в отношении персональных данных, должны открыто сообщаться. Любая компания должна предоставлять субъекту персональных данных возможность выяснения существования в этой компании его персональных данных, состава этих данных, названия и местонахождения компаний, в которые были переданы его персональные данные для обработки.
- Субъекты должны иметь возможность определения, какие организации обрабатывают их персональные данные и какие конкретно данные они обрабатывают, чтобы они могли исправлять ошибочные данные, запрещать определенные действия с ними.

Компании должны нести ответственность за реализацию мер, направленных на выполнение указанных выше принципов.

ПРИМЕЧАНИЕ. Информацию о руководящих принципах ОЭСР можно найти на сайте www.oecd.org/document/18/0,2340,en_2649_34255_1815186_1_1_1_1,00.html.

Хотя работа ОЭСР является отличным началом, предстоит еще большой объем работы по стандартизации подходов к вопросам киберпреступности на международном уровне.

Компании, которые не знают или не соблюдают указанные правила и руководящие принципы, могут быть оштрафованы и осуждены, что может разрушить их бизнес. Если ваша компания имеет планы по международной экспансии, было бы неплохо организовать в ней юридический департамент, который будет работать над этими вопросами, чтобы у компании никогда не возникли подобные проблемы.

Если компания обменивается данными с европейскими организациями, ей необходимо соблюдать требования "Безопасной гавани" (Safe Harbor). В Европе вопросы защиты конфиденциальной информации всегда контролировались более жестко, чем в США и других частях мира. Раньше при взаимодействии американских и европейских компаний возникали значительные сложности, обусловленные существенными различиями в законодательстве, что наносило серьезный ущерб бизнесу этих компаний. Чтобы устранить этот беспорядок, были разработаны документы "Безопасной гавани", в которых определены требования по защите данных, передаваемых в / из Европы. Американские компании, которые постоянно взаимодействуют с европейскими компаниями, могут получить сертификат соответствия этим правилам, чтобы быстрее и проще передавать данные.

Европейский союз (ЕС) относится к частной жизни граждан гораздо более серьезно, чем большинство других стран в мире, поэтому в ЕС разработаны строгие законы, относящиеся к персональным данным, основанные на **Принципах конфиденциальности Европейского союза**. Этот набор состоит из шести принципов, определяющих порядок использования и передачи конфиденциальной информации. Все государства в Европе должны соблюдать эти обязательные принципы.

Европейские принципы конфиденциальности:

- Цели сбора данных должны быть указаны до начала их сбора.
- Данные не могут быть использованы для других целей.
- Ненужные данные не должны собираться.
- Данные должны храниться столько времени, сколько необходимо для достижения указанной цели, но не дольше.
- Доступ к данным должен предоставляться только лицам, которым он необходим для выполнения задач по достижению указанной цели.
- Должен быть назначен ответственный за обеспечение безопасного хранения данных, в обязанности которого входит недопущение непреднамеренных утечек этих данных.

Ссылки по теме:

- Stanford Law University
- Cyber Law in Cyberspace
- Organisation for Economic Co-operation and Development
- International Safe Harbor Privacy Principles

3.4. Типы права

Как было отмечено ранее, правовые системы различных стран имеют существенные различия. В этом разделе мы рассмотрим основные компоненты этих систем и их отличия.

- **Гражданское право (Civil (code) Law)**
 - Система законов, используемая в странах континентальной Европы (например, Франции и Испании).
 - Отличается от Общего права (common law), используемого в США и Великобритании.
 - Гражданское право основано на правилах, а не на прецедентах.
 - Система гражданского реализуется с помощью Кодексов (codified law) – «писанных» законов. Не все страны следуют этому подходу.
 - История гражданского права начинается в шестом веке, когда византийский император Юстиниан разработал Кодексы Рима.
 - Гражданскую правовую систему не следует путать с гражданским законодательством США (деликтное право (tort law)).
 - Гражданское право устанавливается различными странами для саморегулирования (существует, например, Французское гражданское право, Немецкое гражданское право и т.п.).
 - Это самая распространенная правовая система в мире.
 - В соответствии с гражданским правом, суды нижнего уровня не обязаны следовать решениям, принятым вышестоящими судами.
- **Общее право (Common Law)**
 - Разработано в Англии.
 - Основано на следующем толковании законов:
 - В прошлом, судьи ходили по всей стране, обеспечивая соблюдение законов и урегулируя споры.
 - У них не было написанного свода законов, они основывали свои законы на обычаях, традициях и прецедентах.
 - В двенадцатом веке Королева Англии создала единую правовую систему для всей страны.
 - Она отражала моральные принципы и ожидания общества.
 - Это положило начало появлению адвокатов (lawyer), которые принимают активное участие в судебных процессах, представляя свои аргументы и доказательства.
 - В наше время в общем праве используются судьи (judge) и коллегии присяжных (jury of peers).

- Обычно система состоит из суда высшей инстанции (higher court), нескольких апелляционных судов (appellate court) среднего уровня и множества местных судов первой инстанции (trial court). Прецеденты в этой системе идут сверху вниз. Традиционно существуют магистратские суды (magistrate's court), которые предназначены для принятия административных решений.
- Общее право делится на:
 - Уголовное (Criminal)
 - Основано на общем праве, статутном (основанном на законодательных актах) праве (statutory law) или их комбинации.
 - Относится к поведению, неприемлемому для общества.
 - Наказанием обычно является лишение свободы или денежный штраф.
 - Гражданское (Civil) / деликтное (tort)
 - Ответвление уголовного права.
 - В соответствии с гражданским правом обвиняемый имеет правовые обязательства по отношению к пострадавшему. Другими словами, обвиняемый обязан придерживаться определенных норм поведения, чтобы избежать заведомого нанесения вреда потерпевшему.
 - Нарушение обвиняемым этих обязанностей ведет к нанесению ущерба потерпевшему (обычно физического или финансового).
 - Категории нарушений гражданского права:
 - **Умышленные (intentional)**. Примером может быть нападение, умышленное причинение страданий, незаконное лишение свободы.
 - **Направленные на чужую собственность (wrongs against Property)**. Примером может быть нарушение прав землевладельца.
 - **Направленные на личность (wrongs against Person)**. Например, автомобильные аварии, нападения собак.
 - **Халатность (negligence)**. Смерть в результате преступного бездействия.
 - **Совершенные чиновниками и другими государственными служащими (dignitary wrongs)**. Например, вмешательство в личную жизнь, нарушение гражданских прав.
 - **Экономические (economic wrongs)**. Примером могут быть нарушения, связанные с патентами, авторскими правами и торговыми марками.
 - **Нарушение гражданских обязательств (strict liability)**. Например, уведомление о рисках, дефектах продукции и т.п.
 - Административное (administrative (regulatory))

- Законодательство и правовые принципы разрабатываются государственными учреждениями для множества различных областей, включая международную торговлю, производство, вопросы иммиграции.
- Обязанности по доказыванию вины лежат на стороне обвинения, доказательства не должны оставлять никаких разумных сомнений (невиновен пока не доказано обратное).
- Используется в Канаде, Великобритании, Австралии, США, Новой Зеландии.
- **Правовой обычай (Customary Law)**
 - В основном относится к поведению людей и моделям поведения.
 - Основано на традициях и обычаях региона.
 - Появилось, когда возникла потребность в сотрудничестве отдельных людей.
 - Немногие страны работают в рамках системы правовых обычаев, чаще используются смешанные системы, в которых правовые обычаи являются одной из составляющих (кодексы гражданского права возникли из правовых обычаев).
 - Применяется в основном в тех регионах мира, где используются смешанные правовые системы (например, Китай, Индия).
 - Наказания обычно выражаются в виде денежного штрафа или общественных работ.
- **Религиозное право (Religious Law Systems)**
 - Основано на религиозных убеждениях региона.
 - В исламских странах право основывается на нормах Корана.
 - Однако в каждой исламской стране закон отличается.
 - Юристы и священники обладают высоким уровнем доверия.
 - Охватывает все аспекты жизни людей.
 - Обычно делится на:
 - Ответственность и обязательства перед другими.
 - Религиозные обязанности.
 - Знания и правила исходят от Бога, который управляет человеческими делами.
 - Религиозные законы включают в себя кодексы этики и морали, соблюдать которые требует Бог. Например, индуское право (Hindu law), шариат (исламское право, Islamic law), Галаха (еврейское право, Jewish law) и т.д.
- **Смешанные правовые системы (Mixed Law Systems)**
 - Две (или более) правовые системы используются совместно, дополняя друг друга.
 - Чаще всего смешанные правовые системы состоят из гражданского права и общего права.
 - Комбинация используемых систем является результатом более или менее четкого определения областей применения.

- В пределах одного региона, гражданское право может применяться к одним видам преступлений, а религиозное право – к другим.
- Примером стран, в которых применяются смешанные правовые системы, являются Канада, Голландия, Южная Африка.

ПРИМЕЧАНИЕ. С распределением различных типов права по странам мира можно ознакомиться на сайте University of Ottawa Faculty of Law.

Гражданское право имеет дело с преступлениями, направленными на отдельных лиц или компании, в результате которых им был нанесен вред или они понесли ущерб. Это называется *деликтным правом* (tort law). Примером может быть злоупотребление, халатность или ответственность за некачественную продукцию и т.п. Результатом гражданского иска может быть денежный штраф или общественные работы, а не лишение свободы. Коллегия присяжных в гражданском суде решает вопрос ответственности, а не виновности или невиновности. Если коллегия решает, что ответчик несет ответственность за содеянное, коллегия определяет размер штрафа для возмещения убытков пострадавшей стороне.

ПРИМЕЧАНИЕ. Гражданское право в основном произошло от общего права (прецедентного права), судебные разбирательства инициируются частными лицами, а в суде определяется, несет ли ответственность обвиняемый за причиненный ущерб. Уголовное право обычно четко прописано в законодательстве, инициатива судебного разбирательства исходит от правительственных прокуроров, а в суде определяется, виновен ли обвиняемый или нет.

Уголовное право применяется, когда действия человека нарушают государственные законы, разработанные для защиты общества. Наказанием по уголовным делам, как правило, является лишение свободы.

Административное право использует нормативные стандарты, которые регулируют деятельность и поведение. Государственные учреждения разрабатывают эти стандарты, которые обычно применяются к компаниям и физическим лицам в рамках соответствующих отраслей. В качестве примера можно привести требования, предъявляемые к любому офисному зданию, в которых указано, что каждое офисное здание должно быть оборудовано системой выявления и тушения пожара, на стены помещений и коридоров должны быть нанесены указатели на аварийные выходы, а двери в случае пожара должны разблокироваться. В качестве другого примера, можно привести требования стандартов, предъявляемых к компаниям, производящим пищевые или фармацевтические продукты. Эти стандарты направлены на защиту общества от некачественной и опасной для здоровья продукции. Если компания нарушает или игнорирует предъявляемые к ее деятельности требования, ее должностные лица могут быть привлечены к ответственности.

Люди, которые хотят успешно бороться с компьютерной преступностью, должны хорошо понимать психологию своего противника, точно так же, как сотрудники полиции, занимающиеся более традиционными преступлениями. В большинстве случаев, в самом начале расследования компьютерного преступления (как и любого другого), необходимо понять, почему и как оно было совершено. Для успешного расследования преступления, нужно знать, как думает преступник, что мотивирует его на преступную деятельность, каковы его цели и опасения, как они отражаются на преступлениях, которые он совершает. Чтобы реально остановить или хотя бы сократить киберпреступность, необходимо хорошо понимать, почему люди совершают такие преступления.

4. Законодательство в области интеллектуальной собственности

Законы, связанные с интеллектуальной собственностью, направлены в первую очередь на предоставление компании возможности защитить то, что по праву принадлежит ей, а не выяснение – кто прав или неправ.

Основным вопросом при нарушении интеллектуальной собственности чаще всего является

вопрос, что компания сделала для защиты ресурсов, которые по ее утверждению, были использованы с нарушением ее прав. Компания должна предпринять множество шагов по защите ресурсов, на которые она претендует, как свою интеллектуальную собственность, компания должна показать, что она проявила должную заботу и затратила определенные усилия для защиты этих ресурсов. Если сотрудник компании отправил другу некий файл, и компания пытается уволить этого сотрудника, заявляя, что он незаконно предоставил постороннему лицу доступ к интеллектуальной собственности компании, компания должна доказать суду и присяжным, во-первых, почему данный файл является столь важным для компании, какой ущерб может быть (или был) причинен в результате того, что с содержимым файла ознакомилось постороннее лицо, а, во-вторых, и это самое главное, какие меры предприняла компания, чтобы защитить этот файл. Если компания не обеспечила защиту этого файла, а просто сказала своим сотрудникам, что им не разрешается копировать и распространять его, компания, скорее всего, проиграет дело. Если же компания действительно применила ряд мер для защиты этого файла, объяснила своим сотрудникам, что нельзя копировать и распространять информацию из этого файла, что наказанием за нарушение этого требования может быть увольнение, в таком случае компании не может быть предъявлено обвинение в незаконном увольнении сотрудника.

В зависимости от типа ресурса, информация которого является интеллектуальной собственностью компании, для его защиты могут применяться различные законы. Интеллектуальная собственность делится на две категории: промышленная собственность (industrial property) – например, изобретения (патенты), промышленные проекты, торговые марки, и авторское право (copyright), которое распространяется на литературу, искусство и т.п. Эти вопросы рассматриваются более детально далее в этом Доме.

4.1. Коммерческая тайна

Законодательство в области коммерческой тайны защищает определенные типы информации или ресурсы от несанкционированного использования или разглашения. Ресурс содержит коммерческую тайну, если он дает компании некоторое конкурентное преимущество, а его создание требует специальных навыков, знаний и / или денежных и трудовых затрат. Это означает, что компания не может назвать своей коммерческой тайной, например, утверждение, что небо – синее.

Коммерческая тайна (trade secret) – это то, что является собственностью компании и имеет важное значение для ее существования на рынке и получения прибыли. Примером коммерческой тайны может быть формула производства безалкогольных напитков, например, кока-колы, исходный код компьютерной программы и т.п. Компания должна принять разумные меры по защите ресурса, который она объявила своей коммерческой тайной.

Многие компании требуют от своих сотрудников подписать соглашение о неразглашении (NDA – nondisclosure agreement), подтверждающее, что сотрудники взяли на себя обязательство не делиться секретами компании с ее конкурентами. При этом компании преследуют сразу две цели: во-первых, это информирование работников о важности сохранения в тайне определенной информации, а, во-вторых, это способ удержать сотрудников от разглашения такой информации. Подписанное сотрудником соглашение о неразглашении дает право компании уволить его или привлечь к ответственности в случае, если сотрудник раскроет коммерческую тайну компании.

4.2. Авторское право

В США **законодательство по защите авторских прав** (copyright law) защищает права автора на контроль распространения, воспроизведения, отображения и изменение его оригинальной работы. Это законодательство охватывает множество различных видов произведений: живопись, графика, музыка, драма, литература, пантомима, кино, скульптура,

звуковые записи и архитектура. Законодательство по защите авторских прав не распространяется на конкретные ресурсы, в отличие от коммерческой тайны. Оно защищает идею создания ресурса, а не сам ресурс. Закон об авторском праве, как правило, используются для защиты работ писателя, рисунков художника, исходного кода программиста, музыкальных ритмов, созданных музыкантом. Объект авторского права попадает под действие закона об авторском праве сразу после его создания. Закон об авторском праве не требует обязательного предупреждения и/или использования знака авторского права (©), но рекомендуется делать это для того, чтобы другие не могли заявить о своей невинности, скопировав чужие работы.

Защита законодательства об авторских правах не распространяется на любые методы выполнения операций, процессы, концепции или процедуры – оно направлено только на защиту от несанкционированного копирования и распространения авторских работ. В отличие от патента, авторское право защищает форму выражения, а не сам объект. С этой точки зрения, авторское право обеспечивает меньший уровень защиты по сравнению с патентом, однако продолжительность действия защиты авторского права больше (законодательство обеспечивает защиту авторских прав человека в течение всей его жизни плюс 50 лет).

Компьютерные программы могут защищаться законодательством об авторском праве также, как литературные произведения. Законодательство защищает как исходный, так и объектный код, который может представлять из себя операционную систему, приложение или базу данных. Закон может защищать не только код, но и различные структуры, элементы интерфейса и т.п.

4.3. Торговая марка

Торговая марка (trademark) несколько отличается от авторского права, поскольку она используется для защиты слова, названия, символа, звука, формы, цвета или их сочетания. Торговые марки используются компаниями для своего представления группе людей или всему миру, путем создания узнаваемого бренда. Маркетинговые отделы компаний усердно работают, чтобы придумать что-то новое, что позволит компании быть замеченой и выделиться из толпы конкурентов и множества других торговых марок. Результат этой работы должен быть защищен, должны быть исключены возможности его копирования другими.

Компании не могут использовать в качестве торговых марок числа и частоиспользуемые слова. Однако торговой маркой может быть уникальный цвет, узнаваемая упаковка и т.п.

4.4. Патент

Патенты (patent) дают компаниям и частным лицам законную возможность владения изобретением, запрещая без их разрешения другим использовать и копировать изобретение, защищенное патентом. Патент обеспечивает наибольший уровень защиты интеллектуальной собственности. При этом изобретение должно вносить что-то новое, полезное и не быть очевидным, поэтому компания, например, не может запатентовать воздух. Иначе мы вынуждены были бы платить ей за каждый вдох.

После утверждения поданной изобретателем заявки на выдачу патента, изобретателю предоставляется исключительное право собственности, запрещающее изготовление, использование или продажу изобретения другими в течение определенного периода времени. Например, фармацевтическая компания может получить патент на созданное ей лекарство, который будет означать, что эта компания является единственной, кто может производить и продавать это лекарство до указанного в патенте срока. После этого срока, любые компании могут производить и продавать это лекарство (это часто приводит к существенному снижению цен на медикаменты после окончания срока действия патента).

То же самое относится и к алгоритмам программ. Изобретатель алгоритма может получить патент, что даст ему полный контроль над использованием этого алгоритма в любых продуктах. Если кто-то хочет воспользоваться этим алгоритмом, он должен обратиться к изобретателю и договориться с ним об условиях использования алгоритма (обычно такими условиями является единовременная плата или процент от каждого проданного экземпляра продукта, содержащего этот алгоритм).

4.5. Внутренняя защита интеллектуальной собственности

Компании необходимо убедиться, что определенные ее ресурсы защищены рассмотренными выше законами. Однако не менее важно принять и ряд мер внутри самой компании для надлежащей защиты этих ресурсов.

Ресурсы, защищенные одним таких законов, должны быть идентифицированы и интегрированы в схему классификации данных компании. Это должно выполняться персоналом ИТ под контролем руководства. Для таких ресурсов должна быть обеспечена надлежащая защита от несанкционированного доступа, созданы безопасные условия хранения, должен вестись аудит их использования. Лица, которым разрешен доступ к этим ресурсам, должны быть четко идентифицированы, уровень их доступа должен быть детально определен. Любые попытки доступа к этим ресурсам должны контролироваться, ресурсы должны храниться на защищенном сервере с необходимыми механизмами безопасности.

Работники должны быть информированы о степени секретности или конфиденциальности ресурсов, с которыми они работают, об ожидаемом от них поведении в отношении этих ресурсов.

Если компания не выполнила хотя бы один из этих шагов, на нее не распространяется защита рассмотренного ранее законодательства, поскольку она не проявила должную заботу и не обеспечила надлежащую защиту своих ресурсов, которые (по ее утверждению) так важны для ее конкурентоспособности и успешного существования на рынке.

4.6. Компьютерное пиратство

Компьютерное пиратство (software piracy) – это использование или копирование интеллектуальной или творческой работы автора без разрешения с его стороны и компенсации. Это является посягательством на права собственности, поэтому в случае поимки пирата, ему может быть предъявлен иск о возмещении ущерба и/или он может быть привлечен к уголовной ответственности.

Производитель программного обеспечения, как правило, выдает лицензии на его использование, а не продает целиком. При этом в лицензионном соглашении предусматриваются требования, касающиеся порядка использования и защиты программного обеспечения, а также документации к нему. Если частное лицо или компания нарушает эти требования, лицензия может быть аннулирована, а на нарушителя может быть заведено уголовное дело (в отдельных случаях). Риском для производителя, который разработал и выдает лицензии на свое программное обеспечение, является потеря прибыли, которую он мог бы получить. Следует отметить, что в наше время многие компании нарушают лицензионные соглашения на используемое программное обеспечение, а сотрудники компаний не редко используют дома программное обеспечение, купленное компанией.

Существует четыре категории лицензий на программное обеспечение. **Бесплатное программное обеспечение** (freeware) свободно распространяется, оно может бесплатно использоваться, копироваться, анализироваться, изменяться и распространяться в обновленном виде без каких-либо ограничений. **Условно-бесплатное программное обеспечение** (shareware) или **пробные версии программного обеспечения** (trialware) используется производителями для продажи своего программного обеспечения. Они предоставляют пользователям возможность бесплатно получить пробную версию

программного обеспечения. После испытания программы, пользователь должен приобрести ее копию или отказаться от ее использования. **Коммерческое программное обеспечение** (commercial software) – это программное обеспечение, которое продается и используется в коммерческой деятельности. И, наконец, **учебное программное обеспечение** (academic software) – это программное обеспечение, используемое исключительно в учебных целях и поэтому продаваемое по сниженной стоимости. Учебным программным обеспечением может быть программное обеспечение с открытым исходным кодом (open source), бесплатное или коммерческое программное обеспечение.

Некоторые производители программного обеспечения продают групповые лицензионные соглашения (bulk license), позволяющие нескольким пользователям использовать продукт одновременно. Эти соглашения (master agreement) определяют разрешенный порядок использования программного обеспечения и различные ограничения. Другой распространенной формой лицензирования программного обеспечения является Лицензионное соглашение для конечного пользователя (EULA – End User Licensing Agreement). В нем более детально указываются условия и ограничения, по сравнению с обычным лицензионным соглашением (master agreement). Другие производители применяют систему мониторинга (часто при этом используются решения третьих сторон), контролирующую, чтобы клиент не превысил купленное им количество лицензий. Офицер безопасности должен хорошо знать требования лицензионных соглашений на используемое в компании программное обеспечение. Он обязан убедиться, что все требования лицензионных соглашений соблюдаются и внедрить необходимые защитные меры, позволяющие ему выявлять их нарушение. В случае обвинения компании в незаконном копировании программного обеспечения или использовании большего количества копий, чем предусмотрено лицензией, именно офицер безопасности будет в первую очередь нести за это ответственность.

Благодаря легкости использования сети Интернет и постоянного повышения скоростей доступа, у пользователей постоянно растет соблазн загрузить и использовать пиратскую версию программы. По оценкам BSA и IDC (International Data Corporation) в мире в 36% случаев программное обеспечение используется нелегально.

Не в каждой стране компьютерное пиратство считается преступлением, но таких стран становится все меньше, благодаря усилиям ряда международных организаций. Для обеспечения соблюдения своих имущественных прав на программное обеспечение, группа крупных компаний организовала Ассоциацию по защите программного обеспечения (SPA – Software Protection Association). Изначально эта ассоциация была создана для защиты именно этой группы компаний, однако в настоящее время она предоставляет свои услуги и другим компаниям, желающим защитить свои интересы. Пиратство является большой проблемой для производителей программного обеспечения, поскольку большинство из них от лицензионных сборов получают основную часть своих доходов.

Также были созданы и другие международные группы по борьбе с пиратством, в том числе Федерация по борьбе с программным пиратством (FAST – Federation Against Software Theft) со штаб-квартирой в Лондоне, и Альянс производителей программного обеспечения для коммерческих организаций (BSA – Business Software Alliance), базирующийся в Вашингтоне. Они выполняют функции, аналогичные SPA, обеспечивая защиту программного обеспечения по всему миру.

Одним из вариантов правонарушений в отношении программного обеспечения, является декомпиляция скомпилированного программного кода. Обычно это делается для получения исходного кода программы, чтобы проанализировать его и понять, как работает приложение. Исходный код коммерческой программы, как правило, является конфиденциальным, поскольку его использование может позволить понять и использовать в своих целях детали функционирования программного обеспечения. Другой возможной целью проведения

обратного инжиниринга кода программного обеспечения является попытка обнаружить проблемы безопасности, которые впоследствии могут быть использованы. Примером может быть выявление уязвимости, позволяющей провести атаку переполнения буфера.

Декомпилируя объектный код в исходный код, исследователи могут найти проблемы безопасности и воспользоваться ими, либо внести изменения в код программы, чтобы реализовать в ней определенную функциональность, которую не предусмотрел разработчик. В качестве примера можно привести случай, когда человек декомпилировал программу, предназначенную для чтения электронных книг, в которой был реализован механизм защиты от несанкционированного копирования и использования книг. Производитель не хотел, чтобы появилась возможность обхода защиты, реализованной в его продукте, поэтому он закодировал отдельные участки кода программного обеспечения. Декомпилировав объектный код, этот человек выяснил, как можно создать декодер. Это помогло ему преодолеть устанавливаемые программой ограничения и получить возможность свободно копировать и использовать электронные книги в нарушение авторских прав авторов и издателей.

Этот человек был арестован и обвинен в нарушении Закона об авторском праве в цифровом тысячелетии (DMCA – Digital Millennium Copyright Act), в соответствии с которым изготовление программ для обхода механизмов защиты авторских прав, является незаконным. Введение этого закона вызвало множество дискуссий и споров из-за его возможного негативного воздействия на свободу слова и законные научные исследования.

Любопытно, что многие компьютерщики начали активно протестовать против ареста этого человека, в связи с чем обвинившая его компания (Adobe) решила по-быстрому снять все свои обвинения.

Ссылки по теме:

- United States Copyright Office
- Electronic Frontier Foundation, Intellectual Property Online: Patent, Trademark, Copyright
- Caltech Office of the Intellectual Property Counsel
- Find Law
- TracReports

5. Неприкосновенность частной жизни

В мире растут угрозы нарушения прав на неприкосновенность частной жизни (privacy) по мере увеличения зависимости людей от технологий. Среди подходов к защите неприкосновенности частной жизни можно выделить общий подход и регулирование отдельных отраслей. Общим подходом является принятие горизонтальных законов. Они относятся ко всем отраслям одновременно, в том числе к правительству и государственным службам. Регулирование отдельных отраслей выполняется с помощью принятия вертикальных законов, которые устанавливают требования к отдельным вертикалям, например, к финансовому сектору, здравоохранению и т.п. В обоих случаях это направлено на достижение двух целей. Во-первых, эти инициативы направлены на защиту персональных данных. Во-вторых, они обеспечивают баланс между интересами людей и интересами государства и бизнеса, которые собирают и используют персональные данные этих людей для поддержания безопасности государства и бизнеса.

Рядом стран были приняты законы о защите неприкосновенности частной жизни. Например, в США были приняты новые законы, такие как Gramm-Leach-Bliley Act (1999 год) и HIPAA (Health Insurance Portability and Accountability Act), несмотря на то, что на тот момент действовал Закон о конфиденциальной информации (Federal Privacy Act) 1974 года. Это является примером вертикального подхода к обеспечению защиты неприкосновенности

частной жизни. В качестве примера горизонтального подхода можно привести принятый в Канаде Закон о защите персональной информации и электронных документов (Personal Information Protection and Electronic Documents Act) или Закон о защите частной жизни (Privacy Act), введенный в 1993 году в Новой Зеландии.

Закон о конфиденциальной информации был направлен на защиту персональных данных граждан США, которые собирали государственные учреждения. В нем указано, что сбор любых персональных данных может производиться только на законных основаниях, а собранные данные могут использоваться только для целей, для которых они собирались, и храниться в течение обоснованного периода времени. Если учреждение собирает персональные данные, человек имеет право запросить у него собранные о нем данные. Сейчас подобные законы существуют во многих странах мира.

Технологии постоянно развиваются, растет объем данных в информационных хранилищах, развиваются технологии сбора, анализа и распространения данных. Компании-агрегаторы данных собирают большой объем детальной личной информации миллионов людей, хотя большинство людей никогда не слышали об этих компаниях, никогда не открывали у них счета, не давали им разрешения на получение своих персональных данных. Задачей таких компаний является сбор и хранение подробной информации о людях, которую они затем успешно продают. Например, у одной из таких компаний ChoicePoint хранятся персональные данные около 19 миллионов людей.

Казалось бы, это и правильнее – хранить всю информацию в одном месте. Из такого централизованного и надежного источника ее было бы легче получить, она будет достоверна, но... слишком многие захотят получить такой огромный объем личных данных людей. Взлом всего лишь одной такой системы, пусть и хорошо защищенной, окупит любые затраченные на это усилия. В США работает компания LexisNexis, которая занимается сбором и продажей личных и финансовых данных на американских потребителей. В 2005 году компания заявила о краже личной информации 310 000 американцев.

Рост потребности в законодательном регулировании вопросов по защите неприкосновенности частной жизни. Следующие аспекты повысили потребность в разработке дополнительных законодательных требований в области защиты неприкосновенности частной жизни:

- **Улучшение технологий сбора, объединения (агрегирования) и поиска данных**
 - Постоянно появляются новые крупные хранилища персональных данных
- **Стирание границ (глобализация)**
 - Персональные данные передаются между различными странами по множеству причин
 - Глобализация бизнеса
- **Улучшение технологий**
 - Сбор, глубокий анализ, распространение критичной информации

5.1. Законодательство и требования

Законодательство и требования в области компьютерной и информационной безопасности охватывают множество различных областей. В частности, необходимым является регулирование таких областей, как защита персональных данных, злоупотребления компьютерами, защита авторских прав на программное обеспечение, защита конфиденциальных данных, контроль за применением криптографии. Соответствующие требования могут предъявляться как к государственным учреждениям, так и к коммерческим компаниям в рамках охраны окружающей среды, защиты интеллектуальной собственности, обеспечения национальной безопасности, неприкосновенности частной жизни, общественного порядка, охраны здоровья, предотвращения мошенничества.

Специалисты по безопасности должны прикладывать большие усилия, чтобы идти в ногу со временем, и заранее узнавать, как работает последний сетевой червь, новый вариант DoS-атаки, и что надо сделать, чтобы защититься от всего этого. Они должны отслеживать выход новых продуктов по безопасности, анализировать их отличие от существующих продуктов. Им необходимо быть в курсе новых технологий, своевременно узнавать о выходе исправлений и патчей. Они должны знать о новых методах шифрования, механизмах управления доступом, средствах обеспечения безопасности телекоммуникаций, новых методах социальной инженерии и физической безопасности. Кроме того, они должны хорошо знать действующие законы и требования регуляторов, применимые к отрасли, в которой работает их компания. Количество требований, предъявляемых к компаниям, постоянно растет, а их несоблюдение ведет к все более серьезным последствиям, среди которых могут быть штрафы, прекращение деятельности компании, лишение свободы должностных лиц компании.

Разработанные правительством и государственными учреждениями законы и требования обычно не содержат подробных инструкций, следование которым позволит защитить компьютеры и информационные активы компании. Внутренняя среда каждой компании уникальна по своей топологии, применяемым технологиям, инфраструктуре, к ней предъявляются различные бизнес-требования, она имеет различную функциональность, в ней работает различный персонал. А поскольку технологии меняются в столь быстром темпе, законы и правила, устанавливающие детальные требования, быстро теряли бы свою актуальность. Поэтому они разрабатывают высокоуровневые требования, над реализацией которых потом ломают голову компании. Именно здесь приходят на помощь специалисты по безопасности. В прошлом, специалисты по безопасности должны были знать, как выполнять тесты на проникновение, как настраивать межсетевые экраны, — они имели дело только с техническими вопросами обеспечения безопасности. Сегодня специалисты по безопасности вышли из серверных комнат, теперь они принимают гораздо более активное участие в решении бизнес-ориентированных вопросов. Специалист по безопасности должен знать законы и требования, которым должна соответствовать компания, он должен понимать, какой должен быть реализован контроль для обеспечения соответствия им. Современный специалист по безопасности должен стоять одной ногой в техническом мире, а другой — в мире бизнеса.

Со временем экзамен CISSP становится все менее ориентированным на американскую специфику, сейчас он носит значительно более глобальный характер. Из теста исключены специфичные вопросы, касающиеся исключительно законодательства США. В нескольких последующих разделах в качестве примера приведены американские законы и требования, однако почти каждая страна имеет похожие собственные законы (либо разрабатывает их в настоящее время). Для прохождения экзамена CISSP знание самих этих законов не требуется, достаточно понимания причин их разработки и целей.

Закон Сарбейнза-Оксли

Закон Сарбейнза-Оксли (SOX - Sarbanes-Oxley Act) был разработан после ряда громких корпоративных скандалов и мошенничеств, в которых инвесторы потеряли миллиарды долларов, что угрожало всей экономике в целом.

SOX применим к любой компании, ценные бумаги которой торгуются на американской бирже. В основном этот закон содержит требования в отношении методов ведения учета и отчетности о финансовом положении компании. Однако некоторые части, в частности Раздел 404, имеют прямое отношение к информационной безопасности.

В SOX определено, как компании должны отслеживать, управлять и отчитываться по своей финансовой информации. Это включает защиту данных, обеспечение гарантий их целостности и достоверности. Большинство компаний полагаются на компьютерное оборудование и электронные хранилища информации для проведения транзакций и хранения

архива данных, поэтому они обязаны внедрить соответствующие процессы и контроли для обеспечения защиты данных.

За несоответствие требованиям SOX на компанию могут быть наложены штрафы, а руководству компании (включая генерального директора, финансового директора и других) может грозить лишение свободы.

Закон о преемственности страхования и отчетности в области здравоохранения

Закон о преемственности страхования и отчетности в области здравоохранения (HIPAA - Health Insurance Portability and Accountability Act) – это обязательные государственные стандарты и процедуры, определяющие требования к порядку хранения, использования и передачи личной медицинской информации и сведений о состоянии здоровья. Эти требования предоставляют основу и являются руководством по обеспечению безопасности, целостности и конфиденциальности медицинской информации. HIPAA содержит требования по управлению безопасностью для любой организации, которая создает, использует, передает (предоставляет доступ) или уничтожает медицинскую информацию.

Сведения о состоянии здоровья людей могут неправомерно использоваться по многим причинам. Информация с бумаги переносится в электронные системы, что упрощает сопровождение этой информации, ее использование и передачу. Однако это также ведет к появлению более простых способов для несанкционированного использования этой информации. Традиционно, медицинские учреждения отставали от других отраслей бизнеса в вопросах обеспечения информационной и сетевой безопасности, поскольку у них не было бизнес-потребностей для расхода дополнительных средств и ресурсов на внедрение этих вещей. Но не сейчас.

В HIPAA предусмотрены крупные штрафы за несоответствие требованиям. В случае использования медицинской информации с нарушением указанных в HIPAA требований по обеспечению ее конфиденциальности (даже если это произошло по ошибке), на компанию налагается денежный штраф от 100 долларов за каждое выявленное нарушение, до 25 000 долларов в год за нарушение стандарта. Если защита медицинской информации не была обеспечена или была нарушена сознательно, штраф может достичь 50 000 долларов, а руководителю компании грозит лишение свободы на один год. Если защита медицинской информации была нарушена вследствие обмана, штраф возрастает до 250 000 долларов, а виновному грозит лишение свободы на срок до 10 лет. Это серьезный бизнес.

Закон Грэма-Лича-Блилей

Закон Грэма-Лича-Блилей (GLBA - Gramm-Leach-Bliley Act) был принят для того, чтобы защитить информацию клиентов финансовых организаций, от кражи, несанкционированного использования и злоупотреблений. Он требует, чтобы финансовые организации разрабатывали внутренние положения по обеспечению конфиденциальности и предоставляли своим клиентам возможность запрета финансовой организации обмениваться относящейся к этим клиентам информацией с неаффилированными третьими сторонами. В законе указано, что Совет Директоров несет ответственность за различные проблемы безопасности в работе финансовой организации, предусмотрена необходимость реализации управления рисками, проведения обучения персонала по вопросам информационной безопасности, а также требования по надлежащему тестированию реализованных мер безопасности. Также в нем указаны требования по разработке политики безопасности.

Закон о борьбе с компьютерным мошенничеством и злоупотреблениями

Закон о борьбе с компьютерным мошенничеством и злоупотреблениями (CFAA - Computer Fraud and Abuse Act), разработанный в 1986 году и доработанный в 1996 году, является основным антихакерским законом США. Он запрещает семь видов деятельности, определяя их в качестве преступлений:

- Несанкционированный доступ или превышение полномочий доступа к компьютерам правительственных учреждений для получения классифицированной информации
- Несанкционированный доступ или превышение полномочий доступа к компьютерам финансовых организаций, правительственных учреждений и любым другим защищенным компьютерам, участвующим во внутреннем и международном информационном обмене
- Несанкционированный доступ к компьютерам правительственных учреждений (или компьютерам, работающим в интересах правительства), оказывающий негативное воздействие на их работу
- Несанкционированный доступ или превышение полномочий доступа к защищенным компьютерам с целью обмана
- Умышленная несанкционированная передача программы, информации, кода или команды с целью нанесения ущерба защищенному компьютеру
- Умышленный перехват компьютерных паролей с целью мошенничества
- Передача сообщений, содержащих угрозы, способные причинить ущерб защищенному компьютеру

Классификация этих действий варьируется от проступков до преступлений, соответствующим образом устанавливается размер штрафа за их совершение. Максимальной мерой наказания за совершение этих действий является лишение свободы.

Закон о защите персональных данных

В середине 1960-х годов было внесено предложение об организации хранения правительством США в едином банке данных информации о каждом американском гражданине, включая информацию Управления социального страхования, Бюро переписи населения, Налоговой службы, Бюро трудовой статистики и других правительственных учреждений. Комитет, который внес это предложение, видел в этом эффективный вариант сбора и централизованного хранения данных. Однако другие увидели в этом шаг в сторону ограничения права на неприкосновенность частной жизни. Единый государственный банк данных так и не был создан из-за сильного противодействия оппозиции.

Для контроля деятельности правительства в части сбора информации о гражданах США, большинство файлов общедоступны и являются открытой информацией в соответствии с Законом о свободе информации (Freedom of Information Act). Исключение составляют только информация, доступ к которой ограничивается в соответствии с законодательством. Закон о защите персональных данных (Federal Privacy Act) 1974 года распространяется на записи и документы, собираемые и обрабатываемые различными государственными учреждениями, например, органами исполнительной власти, государственными корпорациями, независимыми регулирующими органами, а также корпорациями, контролируруемыми правительством. Этот закон не распространяется на Конгресс, судебные и территориальные подразделения.

Государственные учреждения могут собирать и обрабатывать информацию об образовании людей, истории их болезни, финансовую информацию, сведения о судимости, занятости и другую подобную информацию только если она действительно необходима для реализации целей, для которых были созданы эти учреждения. Закон о защите персональных данных содержит запрет на раскрытие такой информации государственными учреждениями без письменного разрешения от человека, к которому она относится. В случае распространения государственным учреждением персональных данных человека без его согласия, он может подать в суд на это учреждение за нарушение его прав на частную жизнь. Однако, как и в большинстве других законов, в Законе о защите персональных данных существует ряд

исключений.

Поскольку информация хранится в компьютерных системах, всегда существует риск ее утечки. Для защиты от этого, компьютерные системы государственных учреждений должны быть защищены необходимыми механизмами безопасности.

Базель II

Банком международных расчетов (Bank for International Settlements) были разработаны инструменты, позволяющие защитить банки от чрезмерного роста, который может привести к их банкротству. В первоначальном Базельском соглашении по капиталу (Basel Capital Accord) были установлены требования к минимальному размеру капитала, который финансовые организации обязаны держать под рукой.

Базель II вступил в силу в ноябре 2006 года. В нем применен более совершенный подход к определению реальной подверженности риску каждой финансовой организации, он учитывает предпринятые меры по снижению риска, давая финансовым организациям стимул для инвестиций в меры и средства безопасности.

Базель II состоит из трех основных компонентов (pillar). Первый компонент представляет расчет минимальных требований к капиталу на основе результатов оценки кредитного, рыночного и операционного рисков. Второй компонент содержит основные принципы для надзорного процесса, управления рисками и безопасностью. Третьим компонентом является рыночная дисциплина, которая содержит комплекс требований о раскрытии информации, позволяющей участникам рынка оценить подверженность организации риску, убедиться в достаточности ее капитала.

Информационная безопасность является неотъемлемой частью Базель II. Финансовые организации стремятся снизить необходимый размер капитала, который всегда должен быть у них под рукой. А для этого они должны постоянно оценивать свою подверженность риску, внедрять защитные меры для снижения рисков.

PCI DSS

Кражи личных данных и мошенничество с банковскими картами происходят все чаще. Хотя это происходило и раньше, распространение сети Интернет и компьютерных технологий дало злоумышленникам возможность украсть миллионы записей за один раз.

Индустрия банковских карт приняла активные меры для борьбы с этой проблемой и поддержки доверия клиентов к банковским картам, как к безопасному инструменту выполнения финансовых операций. Visa выступила с инициативой создания программы обеспечения информационной безопасности держателей карт (CISP – Cardholder Information Security Protection), другие платежные системы начали разрабатывать свои собственные аналогичные программы.

В конце концов, платежные системы объединили свои усилия и разработали стандарт PCI DSS (Payment Card Industry Data Security Standards). Для поддержки и обеспечения соблюдения требований PCI DSS был организован PCI Security Standards Council.

PCI DSS распространяется на любой процесс или объект, связанный с передачей, хранением или использованием данных банковских карт. Уровень требований, предъявляемых к компании, обрабатывающей данные банковских карт, зависит от ее размеров и объема операций с картами. От этих же параметров зависят и штрафы за нарушение требований. Поскольку банковскими картами пользуются миллионы людей, а карты принимаются к оплате почти в любой торговой организации, очень многие организации по всему миру обязаны соответствовать требованиям PCI DSS.

PCI DSS содержит 12 основных требований, разбитых на шесть основных категорий. Этими шестью категориями PCI DSS являются: Обеспечение и поддержка безопасности сети,

Защита данных держателей карт, Поддержка программы управления уязвимостями, Реализация строгого контроля доступа, Постоянный мониторинг и тестирование сети, Поддержка политики информационной безопасности. Основными требованиями PCI DSS являются:

- Установить, настроить и поддерживать межсетевой экран для защиты данных держателей карт
- Изменить настройки безопасности и пароли, установленные производителем «по умолчанию»
- Обеспечить защиту данных держателей карт при хранении
- Обеспечить шифрование данных держателей карт при их передаче по открытым сетям
- Использовать и регулярно обновлять антивирусное программное обеспечение
- Соблюдать меры безопасности при разработке и поддержке систем и приложений
- Ограничить доступ к данным держателей карт в соответствии со служебной необходимостью
- Каждый человек, имеющий доступ к компьютеру, должен иметь уникальный идентификатор
- Ограничить физический доступ к данным держателей карт
- Отслеживать и контролировать любой доступ к сетевым ресурсам и данным держателей карт
- Регулярно тестировать системы и процессы обеспечения безопасности
- Поддерживать политику, регламентирующую вопросы информационной безопасности

PCI DSS является инициативой коммерческой индустрии. Это не закон. Несоблюдение или нарушение требований PCI DSS может привести к финансовым штрафам или отзыву членства организации в международной платежной системе, но не к лишению свободы. Однако следует отметить, что Миннесота недавно стала первым американским штатом, утвердившим PCI Compliance в качестве закона, а другие штаты и правительство США рассматривают аналогичные меры.

Закон о компьютерной безопасности

Закон о компьютерной безопасности (Computer Security Act) от 1987 года требует, чтобы американские государственные учреждения провели идентификацию компьютерных систем, содержащих критичную информацию. Каждое государственное учреждение должно разработать политику безопасности и план для каждой из этих систем, организовать периодическое обучение для лиц, которые поддерживают и управляют этими системами, а также для их пользователей. Сотрудники государственного учреждения должны быть осведомлены о требованиях безопасности, а также об установленных правилах по использованию компьютеров и этих систем.

Поскольку правительство США работает с большим количеством важной, конфиденциальной и секретной информации, ему требуется уверенность в том, что все сотрудники должным образом осведомлены о действующих требованиях безопасности, а все системы соответствуют необходимому уровню защиты.

Закон об экономическом шпионаже

До 1996 года в США не было каких-либо руководящих принципов, которые могли бы быть использованы при расследовании случаев промышленного и корпоративного шпионажа. Закон об экономическом шпионаже (Economic Espionage Act) 1996 года содержит

необходимые для рассмотрения подобных случаев принципы, он делит коммерческую тайну на техническую, деловую, инженерную, научную и финансовую. В нем говорится о том, что защищаемые активы не обязательно должны быть материальными. Этот Закон позволяет ФБР расследовать случаи промышленного и корпоративного шпионажа.

5.2. Вопросы неприкосновенности частной жизни сотрудников

Компания должна продумать ряд вопросов по защите персональных данных своих сотрудников. Компания должна понимать, что в каждом государстве и регионе могут быть свои особенности законодательства по защите персональных данных, которые она должна проанализировать, чтобы понять, что она может делать с персональными данными сотрудников, а что - нет.

Проверка кандидатов на работу в компании. В Домене 01 мы рассматривали вопрос, почему так важно надлежащим образом проверять людей перед их приемом на работу в компанию. Это необходимо для собственной защиты компании и позволяет найти именно того сотрудника, который нужен компании для выполнения определенной работы. В этом Домене этот же вопрос рассматривается с другой стороны – в части прав человека на неприкосновенность частной жизни.

Существуют ограничения по видам и объемам информации, которую компания может получить в отношении кандидата на вакантную должность. Такие ограничения могут быть различными в разных странах и регионах, поэтому специалисты по подбору персонала должны проконсультироваться по этому вопросу с юристами. Обычно кадровые службы создают шаблоны для специалистов по подбору персонала, которых те должны придерживаться при проведении интервью с кандидатом и проверки кандидата.

Если компания использует средства для перехвата набираемой на клавиатуре информации, электронной почты, интернет-трафика и иными способами следит за действиями своих сотрудников, она должна предпринять шаги для уведомления об этом сотрудников. Сотрудники должны знать, что компания может использовать такие способы мониторинга, это не должно оказаться для них сюрпризом. Уведомление сотрудников необходимо для обеспечения законности такого контроля их действий.

Проводимый мониторинг должен иметь отношение только к работе, т.е., например, руководитель может прослушивать разговоры своих подчиненных с клиентами, но он не имеет права прослушивать их личные разговоры, не связанные с работой. Мониторинг должен применяться ко всем сотрудникам в равной степени, а не только к одному – двум сотрудникам.

Если компания считает необходимым вести мониторинг переписки сотрудников по электронной почте, она должна объяснить это сотрудникам сначала посредством политики безопасности, а затем с помощью постоянных напоминаний, например, в виде баннера на внутреннем веб-сайте компании или в процессе регулярного повышения осведомленности. Лучше всего, чтобы сотрудники были ознакомлены под роспись с документом, в котором описаны виды мониторинга, применяемого компанией для контроля действий сотрудников, указано, что считается приемлемым поведением, и каковы последствия несоблюдения действующих в компании требований. Подписывая такой документ, сотрудник отказывается от части своих прав.

Если компания хочет иметь возможность контроля электронной почты сотрудников, она должна отразить этот момент в своей политике безопасности и стандартах. Компания должна указать, кто уполномочен читать сообщения сотрудника и при каких обстоятельствах может применяться мониторинг электронной почты, из каких источников будут перехватываться передаваемые сообщения (почтовый сервер компании, внешние почтовые сервисы, компьютер сотрудника). Если компания будет выполнять подобный мониторинг тайно, не уведомив о нем сотрудников, это может закончиться для нее судебными исками. Хотя специалисты ИТ и подразделений безопасности имеют доступ ко многим компонентам компьютерных систем и сетей компании, это не означает, что у них есть моральные права и

законные основания, чтобы вмешиваться в личную жизнь сотрудников компании. Они могут выполнять только те задачи, которые необходимы для реализации положений политики безопасности и ничего более.

Было немало случаев, когда компании увольняли сотрудников за неправильные действия (просмотр порносайтов, отправку по электронной почте конфиденциальной информации компании ее конкурентам и т.п.), а сотрудники подавали на компанию в суд за неправомерное увольнение. Если компания не указала в своей политике безопасности, что такие действия запрещены, и не предприняла достаточных мер для информирования сотрудников (путем повышения их осведомленности по вопросам безопасности, размещения баннеров на внутреннем сайте и т.п.) о том, что считается приемлемым, а что неприемлемым, какие последствия могут быть за выполнение запрещенных действий, в таком случае у сотрудника есть хорошие шансы выиграть судебное дело и получить компенсацию от компании. Чтобы избежать этого, компания должна учитывать эти вопросы в своих политиках, стандартах, сообщать о них в рамках мероприятий по повышению осведомленности сотрудников по вопросам безопасности. Если это не было сделано, адвокат сотрудника будет утверждать, что сотрудник имел право на личную жизнь.

Защита собственных персональных данных. У пользователей также есть обязанности по защите своих персональных данных в своих системах. Они должны руководствоваться здравым смыслом и лучшими практиками. Защитные меры включают шифрование критичных персональных данных, использование межсетевого экрана, антивирусной программы, регулярную установку патчей. При удалении информации, содержащей критичные персональные данные (например, номер банковской карты), следует использовать специальные утилиты, удаляющие информацию без возможности ее восстановления. Пользователи должны понимать, что как только их данные станут доступны третьей стороне, они потеряют контроль над ними.

6. Обязательства и последствия их нарушения

Законодательные и правоохранительные органы, суды развивают и совершенствуют свои подходы к компьютерным преступлениям, также как и многие компании. Компании должны развивать не только свои превентивные, детективные и корректирующие подходы, но и подходы к своим обязательствам и ответственности. Поскольку компьютерные преступления совершаются все чаще и становятся все изощреннее, растет ущерб от них и продолжительность их воздействия. Во многих случаях злоумышленников не могут поймать, что вызывает все больше недовольства.

Это же справедливо и для других видов угроз, с которыми сталкиваются современные компании. Если здание компании построено из материалов, которые могут полностью сгореть, поджигатель будет только одной маленькой деталью этой трагедии. Компания обязана установить систему выявления пожара, систему пожаротушения, сигнализацию, она должна использовать огнеупорные строительные материалы, должна предусмотреть пожарные выходы, закупить огнетушители, сделать резервные копии всей важной информации, которая может быть повреждена в результате пожара. Если здание компании полностью сгорело, и огонь уничтожил все данные (сведения о клиентах, данные бухгалтерского учета и другую информацию, необходимую для восстановления бизнеса), это означает, что компания не проявила должной осмотрительности для защиты от таких случаев (например, с помощью организации резервного копирования данных и хранения копий на удаленной площадке). В таком случае сотрудники, акционеры, клиенты и другие пострадавшие могут подать в суд на компанию. Однако, если компания сделала все, что должна была сделать по перечисленным выше пунктам, к ней не смогут предъявить претензий за проявление халатности.

В контексте обеспечения безопасности, **должная забота** (due care) означает, что компания предпринимает все разумные меры для предотвращения нарушений безопасности, реализовала надлежащий контроль и внедрила необходимые защитные меры для снижения уровня возможного ущерба. Должная забота – это применение на практике здравого смысла

и разумного управления, ответственное выполнение своих обязанностей. **Должная осмотрительность** (due diligence) означает, что компания надлежащим образом анализирует все свои вероятные недостатки и уязвимости.

Чтобы понять, как обеспечить надлежащую защиту компании, необходимо сначала выяснить, от чего вы будете ее защищать. Именно об этом говорит должная осмотрительность – анализ и оценка текущего уровня уязвимостей для понимания истинного уровня рисков, перед лицом которых стоит компания. Только после этого можно разрабатывать и внедрять защитные меры и средства.

Аналогичного уровня ответственности начинают ожидать от компаний и в отношении компьютерных преступлений и защиты информационных ресурсов. Безопасность реализуется для защиты ценных ресурсов компании, это необходимо для обеспечения гарантий защиты миссии компании посредством защиты ее материальных и нематериальных ресурсов, репутации, персонала, клиентов, акционеров, а также юридического статуса. Безопасность является средством достижения целей компании, а не вещью в себе. Безопасность обеспечивается не для того, чтобы она просто была. Для реализации необходимого компании уровня безопасности требуется полное понимание целей, надлежащее планирование, постановка реально выполнимых задач.

Высшее руководство обязано защитить компанию от множества действий, которые могут негативно повлиять на нее, в том числе оно обязано обеспечить защиту от вредоносного кода, стихийных бедствий, защитить персональные данные сотрудников и клиентов, соблюдать требования законодательства и многое другое.

Затраты и выгоды обеспечения безопасности должны быть оценены как в финансовых, так и нефинансовых единицах, что позволит избежать излишних расходов на безопасность, которые будут превышать получаемые в результате преимущества. Уровень безопасности должен быть пропорционален оцененным рискам, основанным на степени тяжести последствий и вероятности реализации рисков. Механизмы безопасности должны быть реализованы для снижения частоты инцидентов, связанных с нарушением безопасности, а также величины потерь от них.

Высшее руководство должно решить, какой уровень риска оно готово принять в отношении компьютерной и информационной безопасности, а также ответственно подойти к выбору и внедрению экономички целесообразных мер безопасности (эти вопросы подробно обсуждались в Домене 01). Эти риски могут выходить за границы компании. Многие компании работают с третьими сторонами, с которыми они должны совместно использовать критичные данные. Основная компания по-прежнему несет ответственность за защиту этих данных, даже если они обрабатываются в сети другой компании. Именно поэтому появляется все больше правил, обязывающих компании оценивать предпринимаемые третьей стороной меры по обеспечению безопасности.

При совместной работе компаний, особое внимание следует уделить тому, чтобы каждая сторона взяла на себя обязательства по обеспечению необходимого уровня защиты. Эти обязательства, а также ответственность за их нарушение, должны быть четко указаны в договоре, подписанном каждым из участников. Целесообразно провести аудит и тестирование, чтобы убедиться, что каждая сторона действительно выполняет взятые на себя обязательства.

Если одна из таких компаний не обеспечивает необходимый уровень защиты, что оказывает негативное влияние на ее партнера, с которым она работает, пострадавшая компания может подать на нее в суд. Предположим, например, что компании А и Б организовали экстрасеть для связи друг с другом. Компания А не внедрила у себя систему антивирусной защиты и однажды в ее сеть проникает компьютерный вирус, который через созданную экстрасеть распространяется и на сеть компании Б. Вирус уничтожает данные, имеющие критическое

значение для работы компании Б, что приводит к нарушению ее работы. В этом случае компания Б может подать в суд на компанию А за ее небрежность. Обе компании должны убедиться, что они на практике реализуют все разумные меры по выполнению взятых на себя обязательств, и не окажут негативного воздействия на компанию-партнера.

ПРИМЕЧАНИЕ. Обязательства обычно описываются в виде перечня обязанностей, ожидаемых действий и поведения определенной стороны. Для описания обязательств может использоваться и более общий и открытый подход, позволяющий стороне самостоятельно решить, как она будет выполнять свои обязательства. Ведение журналирования действий и событий позволяет четко определить, какая из сторон несет ответственность за определенные действия или бездействие.

К каждой компании предъявляются различные требования в отношении списка обязательств, о выполнении которых она должна позаботиться. Если компания не выполнит эти обязательства, ей может быть предъявлено обвинение в халатности (если это приведет к возникновению ущерба). Чтобы в суде доказать обвинения в халатности, истец должен доказать, что ответчик не выполнил свою **юридическую обязанность** (legally recognized obligation), чтобы защитить истца от необоснованных рисков, и что именно это стало **основной причиной** убытков истца. Наказание за халатность может определяться в рамках гражданского или уголовного права в зависимости от последствий. Наказанием могут быть штрафы, возмещение убытков пострадавшим, либо лишение свободы ответственных лиц компании, виновных в нарушении закона.

Ниже рассмотрены несколько примеров ситуаций, в которых компании могут быть привлечены к ответственности за халатное отношение к своим обязанностям.

6.1. Персональные данные

Рассмотрим следующую ситуацию. Компания Medical Information Inc. обрабатывает и хранит медицинскую информацию, но у нее нет четких правил по распространению и совместному использованию этой информации.

Человек обращается к Medical Information Inc., представляется врачом и просит предоставить ему информацию о здоровье пациента Дона Хэмми. Секретарь, не задавая никаких вопросов, сообщает ему, что у Дона Хэмми опухоль головного мозга. Через неделю Дону Хэмми отказывают в трудоустройстве на работу в компанию, в которой он проходил собеседование. Дон Хэмми догадывается, что работодатель обратился в Medical Information Inc. и получил сведения о его заболевании.

Какие обязательства нарушила компания Medical Information Inc. и потенциальный работодатель Дона? Если он обратится в суд, суд будет рассматривать следующие вопросы:

- **Юридические обязанности**
 - Medical Information Inc. не имеет политик и процедур по вопросам защиты информации пациентов
 - Работодатель не имеет права задавать подобные вопросы и использовать медицинскую информацию о потенциальных работниках
- **Несоблюдение требуемых норм**
 - Критичная информация была передана неустановленному лицу сотрудником Medical Information Inc.
 - Работодатель запросил информацию, которую он не вправе запрашивать
- **Непосредственные причины нанесения вреда**
 - Информация о Доне Хэмми, переданная Medical Information Inc., привела к большим проблемам для него, не позволив получить определенную работу
 - Работодатель принял решение на основе информации, которую он не имел

права получать. Незаконные действия работодателя, выразившиеся в анализе им личной медицинской информации Дона Хэмми, привели к его отказу от трудоустройства Дона Хэмми.

Судебное дело разбиралось очень долго, но в итоге Дон Хэмми выиграл суд против обеих компаний, на полученные в качестве компенсации деньги вылечился от опухоли головного мозга, купил себе остров, и ему никогда не приходилось работать снова.

6.2. Атака хакеров

Рассмотрим другую ситуацию. Финансовая компания Cheapo Inc. внедряет программное обеспечение для удаленного банковского обслуживания, чтобы предложить своим клиентам возможность удаленного управления своими банковскими счетами. Но она не обеспечивает мер безопасности, необходимых для проведения электронных платежей через Интернет.

В первые же две недели работы системы счета 22 клиентов были взломаны и с них было похищено в общей сложности \$439 344,09 долларов США.

Какие обязательства нарушила компания Cheapo Inc.? Если дело будет передано в суд, судом будут рассмотрены следующие вопросы:

- **Юридические обязанности**

- Cheapo Inc. не внедрила межсетевой экран и систему выявления вторжений для защиты базы данных, содержащей информацию о счетах клиентов, и не обеспечила шифрование передаваемой клиентами платежной информации
- Cheapo Inc. не выполняла эффективной защиты денежных средств своих клиентов

- **Несоблюдение требуемых норм**

- Отсутствие у Cheapo Inc. политики и программы безопасности, а также необходимых защитных мер, нарушает 12 требований законодательства, относящегося к работе финансовых компаний

- **Непосредственные причины нанесения вреда**

- Отсутствие должной заботы и невыполнение Cheapo Inc. требований по обеспечению безопасности системы дистанционного банковского обслуживания стало причиной того, что 22 клиента потеряли \$439 344,09 долларов США.

Против Cheapo Inc. был подан коллективный судебный иск, и большинство клиентов получило назад свои деньги. Чтобы рассчитаться с долгами, компания Cheapo Inc. была вынуждена продать здание своего центрального офиса.

Приведенные выше сценарии в упрощенной форме показывают, что невыполнение обязательств по обеспечению компьютерной и информационной безопасности может привести компанию и ее должностных лиц к судебному преследованию. При этом Совет Директоров компании также может нарушить свои обязательства по отношению к акционерам, клиентам и сотрудникам, не обеспечив должной заботы.

Ссылки по теме:

- U.S. Department of Justice
- Computer Fraud and Abuse Act
- White Collar Prof Blog

7. Расследования

Число компьютерных преступлений постоянно растет, поэтому специалистам по безопасности важно понимать, как должно проводиться расследование таких преступлений. Это включает в себя понимание требований законодательства к конкретным ситуациям, системы охраны вещественных доказательств (chain of custody), понимание того, какие доказательства приемлемы для использования в суде, какие следует применять процедуры реагирования на инциденты, процессы эскалации.

При расследовании компьютерного преступления очень важно правильно выполнить все необходимые для этого процедуры, предусмотренные законодательством. Это необходимо, чтобы собранные доказательства могли быть приняты в суде и проверены. Любое нарушение процедуры сбора доказательств может привести к тому, что доказательства не будут приняты судом и это разрушит все дело. Специалист по безопасности должен понимать, что расследование касается не только потенциальных уликов на жестком диске – оно будет проводиться в отношении всего окружения: люди, сеть, любые подключенные внутренние и внешние системы.

7.1. Реагирование на инциденты

О многих компьютерных преступлениях ничего не известно, поскольку часто жертвы таких преступлений либо не знают о том, что они стали жертвами, либо они хотят просто исправить проблему безопасности, которой воспользовался хакер, и сохранить в тайне сведения о произошедшем инциденте, чтобы избежать ущерба репутации компании. Это не позволяет рассчитать реальную статистику компьютерных преступлений – сколько их происходит ежедневно, каков причиняемый ущерб, какие методы атак применяются злоумышленниками.

Мы часто используем термины «событие» (event) и «инцидент» (incident) как взаимозаменяемые, но на самом деле между ними есть важное различие. *Событие* – это негативное происшествие, которое можно выявить, проверить и задокументировать, тогда как *инцидент* представляет собой последовательность событий, которая негативно влияет на компанию и / или воздействует на состояние ее безопасности. Именно поэтому мы называем реагирование на подобные проблемы «реагированием на инциденты» (incident response) и «обработкой инцидентов» (incident handling), именно инциденты вызывают нарушения безопасности и ведут к негативному влиянию на компанию.

Существует множество различных видов инцидентов (вирусы, инсайдерские атаки, утечки или повреждения информации и т.п.), часто они бывают следствием человеческих ошибок. На практике, люди, в обязанности которых входит реагирование на инциденты, довольно часто получают вызовы, потому что "система работает странно". Причиной может быть недавняя установка патча, который испортил что-нибудь, неправильная настройка системы администратором или ошибки программиста.

Если в компании происходит компьютерное преступление, она должна сохранить доказательства и окружение в неизменном состоянии и обратиться к специалистам (правоохранительным органам), уполномоченным на расследование таких происшествий. Человек, не знакомый с правильной организацией проведения процедуры сбора данных и уликов на месте преступления, может уничтожить важные доказательства, что может привести к невозможности судебного преследования злоумышленника. У компании должны быть заранее проработанные процедуры по различным вопросам в области компьютерной безопасности, в частности, процедурам обеспечения непрерывности, процедурам восстановления после аварий, процедурам резервного копирования данных. Помимо них, должна быть разработана процедура расследования компьютерных инцидентов, поскольку количество таких инцидентов многократно возрастает из года в год. Среди таких инцидентов сетевые атаки, атаки на информационные системы, спам, фишинг, вредоносное программное

обеспечение, DDoS-атаки и т. п.

К сожалению, многие компании не имеют представления о том, к кому обращаться и что делать в случае, если они станут жертвой компьютерного преступления. Поэтому любая компания должна заранее разработать политику реагирования на инциденты и соответствующие процедуры, предназначенные именно для таких случаев. Эта политика должна разрабатываться и поддерживаться юридическим департаментом (либо, как минимум, с его активным участием).

Политика реагирования на инциденты должна быть краткой и ясной. Например, она должна указывать, следует ли отключать системы, чтобы попытаться сохранить доказательства, либо системы должны продолжать работать, невзирая на риск уничтожения доказательств. Каждой системе и функции должен быть присвоен приоритет. Например, если файловый сервер компании заражен вирусом, его нужно отключить от сети, но не выключать. Однако если заражен почтовый сервер, он должен продолжать работать, поскольку он имеет более важное значение для компании, чем файл-сервер. Лучше заранее продумать все подобные компромиссы, прежде чем произойдет такая ситуация. Это позволит принять значительно более обдуманное и логичное решение, по сравнению с эмоциональным решением, принятым в спешке и хаосе критической ситуации.

В любой компании должна быть организована **команда реагирования на инциденты** (incident response team), которая должна быть готова к работе с широким спектром возможных инцидентов. Такая команда нужна для того, чтобы в компании всегда была группа имеющих необходимые навыки людей, которые готовы в короткие сроки приступить к выполнению определенного набора процедур в случае инцидента. Среди прочего, эта команда должна уметь скоординировано работать с правоохранительными органами. Команда реагирования на инциденты является одним из важнейших элементов программы безопасности в целом. В составе команды должны быть представители бизнес-подразделений, юридического департамента, кадровой службы, службы физической (корпоративной) безопасности, службы по связям с общественностью, подразделения информационной безопасности, департамента ИТ и кто-то из высшего руководства компании.

Существует три различных типа команды реагирования на инциденты. *Виртуальная команда* (virtual team) состоит из специалистов, которые в обычное время выполняют другие обязанности в компании. Время реагирования такой команды обычно больше, чем для команд других типов, к тому же в случае инцидента члены виртуальной команды будут вынуждены пренебречь своими повседневными обязанностями в компании. В результате, виртуальная команда может оказаться дорогим решением. Однако не каждая компания может позволить себе организовать *постоянную команду* (permanent team), состоящую из людей, единственной функцией которых является реагирование на инциденты. Третьим типом является *смешанная команда* (hybrid team). Ядро такой команды состоит из людей, выполняющих исключительно работу в части реагирования на инциденты, остальные члены команды привлекаются к ее работе в случае необходимости.

У команды реагирования на инциденты всегда под рукой должны быть следующие документы (сведения):

- Перечень внешних организаций и ресурсов, с которыми нужно взаимодействовать или перед которыми нужно отчитываться в случае инцидента
- Описание ролей и обязанностей
- Дерево вызовов для взаимодействия с этими ролями и внешними организациями (лицами)
- Список экспертов по компьютерной криминалистике, к которым можно обратиться

при необходимости

- Описание шагов, которые необходимо выполнить для обеспечения сохранности и защиты доказательств
- Перечень пунктов, которые должны быть включены в отчет для руководства и (возможно) для суда
- Описание того, что следует делать с различными системами в такой ситуации (например, должны ли системы быть отключены от сети Интернет, внутренней сети или полностью выключены).

При получении информации о потенциальном преступлении, команда реагирования на инциденты должна приступить к выполнению предопределенного набора шагов, что позволит обеспечить использование единообразного подхода и гарантирует, что ни один из шагов не будет пропущен. Сначала команда реагирования на инциденты должна проанализировать полученную информацию и определить, действительно ли было совершено преступление. Если преступление было действительно совершено, об этом должно быть немедленно проинформировано высшее руководство компании. Если подозреваемый является сотрудником компании, немедленно должен быть вызван представитель кадровой службы. Лучше как можно скорее начать документирование событий. Если кто-то сможет задокументировать, что происходило в момент совершения преступления, это может послужить в дальнейшем хорошей основой для доказательств. Кроме того, компания должна сразу решить, будет ли она проводить внутреннее расследование или обратится к правоохранительным органам. Во втором случае нужно позаботиться о том, чтобы сохранить атакованную систему в том состоянии, в котором она находилась на момент обнаружения преступления, чтобы сохранить максимально возможное количество улик. Если же компания решает проводить внутреннее расследование, она должна решить целый ряд вопросов, учитывая при этом отдельные важные моменты (подробнее о проведении расследований компьютерных преступлений будет рассказано в следующем разделе).

Компьютерные сети и бизнес-процессы стоят перед лицом множества различных угроз, каждая из которых требует определенной реакции. Тем не менее, команде реагирования на инциденты следует разработать и описать основные шаги, применимые к обработке любых инцидентов. Это гораздо лучше реактивного подхода, при котором о дальнейших действиях начинают думать, когда инцидент уже произошел – такие решения обычно эмоциональны, недостаточно продуманы и не скоординированы. Четко описанный процесс обработки инцидентов значительно более экономически эффективен, он обеспечивает более быстрое восстановление, а также единый подход, позволяющий рассчитывать на определенные результаты.

Обработка инцидентов должна быть тесно связана с планированием восстановления после аварий, она должна являться частью этого плана (приложением), поскольку оба этих плана предназначены для реагирования на определенные инциденты, которые требуют быстрой реакции для максимально быстрого возвращения компании к нормальной работе. Обработка инцидентов – это план восстановления, относящийся к техническим угрозам. Основной задачей обработки инцидента является минимизация ущерба, вызванного инцидентом, а также предотвращение дальнейшего ущерба. Обычно этапами обработки инцидента является: выявление проблемы, определение причин ее возникновения, решение проблемы и документирование всего этого процесса.

Не имея эффективной программы обработки инцидентов, часто люди даже с лучшими намерениями, могут только ухудшить ситуацию, повредив доказательства, разрушив системы и данные, либо допустив распространение вредоносной программы. Злоумышленники часто ставят ловушки в скомпрометированной системе, позволяющие,

например, удалить определенные файлы, имеющие критическое значение, при выполнении пользователем какого-нибудь простого действия, например, вывода списка файлов в каталоге. Скомпрометированной системе нельзя доверять, поскольку злоумышленник мог переопределить внутренние команды или заменить системные файлы, вызов любой команды или программы на такой системе может привести к неожиданным последствиям. Злоумышленник может установить на скомпрометированную систему бэкдор, который позволит злоумышленнику вернуться в эту систему в любой момент, либо логическую бомбу, которая будет тихо ждать пользователя, чтобы уничтожить важные улики или начать следить за действиями пользователя.

Вопросы обработки инцидентов должны учитываться при проведении обучения и повышения осведомленности персонала компании по вопросам безопасности. В частности, до сведения сотрудников должны доводиться отдельные данные по ранее произошедшим инцидентам, чтобы они понимали, с чем может столкнуться компания, что нужно делать в таких случаях и как избежать повторения подобных инцидентов.

Сотрудники должны знать, каким образом и кого нужно уведомлять о произошедшем инциденте. Для этого политика реагирования на инциденты должна содержать детальное описание процесса эскалации, чтобы сотрудники понимали, когда о преступлении нужно сообщать руководству, а когда внешним организациям или правоохранительным органам. Этот процесс должен быть централизованным, удобным и легко выполнимым (иначе сотрудники не будут никого беспокоить). Многие сотрудники очень неохотно сообщают об инцидентах, потому что они боятся наказания, либо не хотят быть причастны к этому. Нет ничего хуже, чем получить по рукам, пытаясь сделать что-то правильное и важное. Сотрудники должны чувствовать себя вполне комфортно, участвуя в этом процессе, а не опасаться за последствия, сообщив о подозрительных действиях.

Политика реагирования на инциденты должна указывать, как сотрудники должны отвечать на вопросы внешних организаций, таких как средства массовой информации, правительственные учреждения, правоохранительные органы. Это достаточно сложный вопрос, он зависит, в частности, от юрисдикции, характера преступления и доказательств. Одна лишь юрисдикция может зависеть от страны, региона или надзорного органа. Учитывая нежелательность широкого распространения информации об инциденте в компании, в процессе такого взаимодействия должны участвовать сотрудники подразделения по связям с общественностью, кадровой службы или другие заранее подготовленные люди, уполномоченные на публичное обсуждение произошедших в компании инцидентов. Если при публичном сообщении информации о произошедшем инциденте что-то было сделано неправильно, это может усугубить негативные последствия. Например, в современном информационном обществе ответ «без комментариев» или отрицание может привести к негативной реакции и распространению порочащих компанию слухов. Однако если все сделано правильно, это может дать компании возможность вернуть доверие общественности. Во многих странах либо уже существуют, либо планируются законы, требующие, чтобы компании сообщали общественности информацию об инцидентах безопасности (произошедших или подозреваемых), связанных с персональными данными или иной идентификационной информацией людей. Поэтому открытость и откровенность компании при взаимодействии с третьими сторонами все чаще находится в интересах компании.

Программа обработки инцидентов должна предусматривать работу с внешними учреждениями и контрагентами. Члены команды должны подписаться на рассылку CERT (Computer Emergency Response Team), чтобы получать актуальную информацию о новых угрозах и иметь возможность выявить признаки преступной деятельности, прежде чем преступление реально произойдет. CERT – это организация, которая осуществляет мониторинг и консультирует пользователей и компании о мерах по обеспечению безопасности и нарушениях безопасности.

7.2. Процедуры реагирования на инциденты

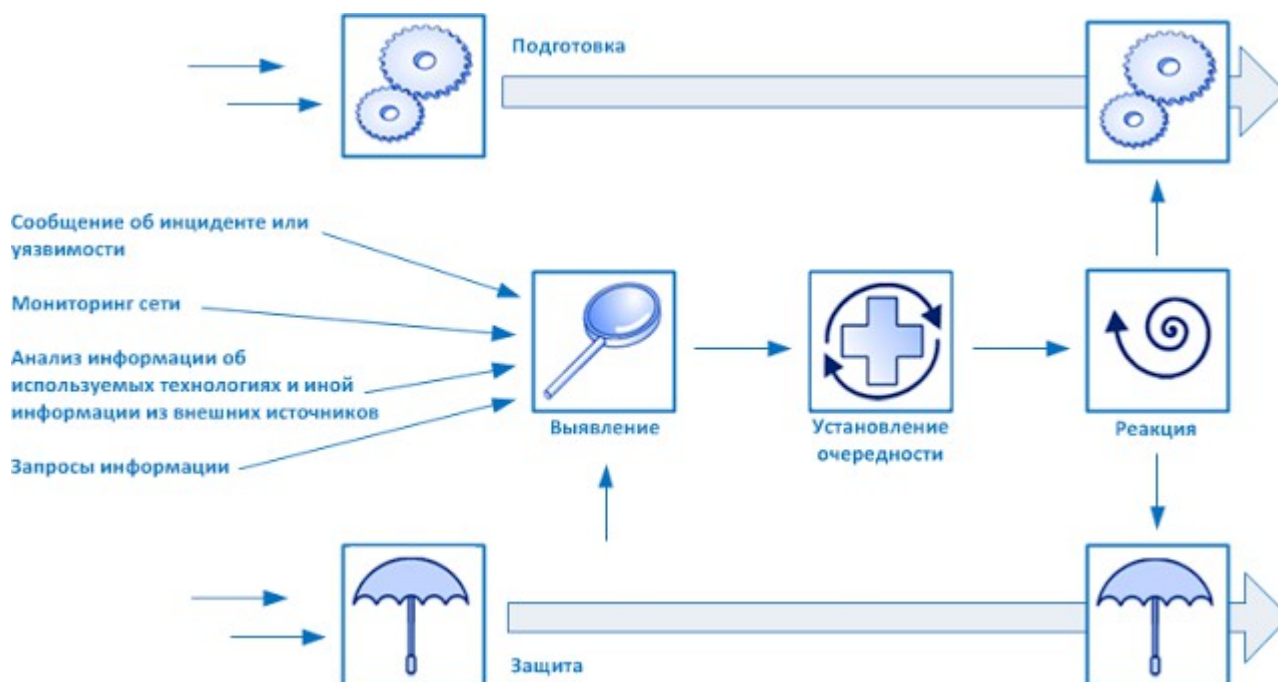
В предыдущих разделах было неоднократно сказано, что в компании должен быть предварительно разработанный набор процедур, которым будет следовать команды при возникновении инцидента. Но что это за процедуры? Хотя различные компании по-разному описывают эти процедуры (или этапы), по сути, они выполняют практически одно и то же. Реагирование на инциденты – это динамический процесс. Отдельные его этапы выполняются параллельно, а другие – последовательно, в случае, если следующий этап зависит от результатов предыдущего. Очень важно, чтобы компания использовала методический подход, который обеспечит надлежащее документирование событий и действий, что может иметь важное значение на более поздних этапах процесса реагирования на инциденты или, если дело дойдет до суда, когда вас спросят, следовали ли вы стандартным процедурам, не пропускали ли вы отдельные шаги. Документ с перечнем шагов и вашими отметками об их выполнении будет приемлемым доказательством в суде.

Вам следует понимать следующий набор процедур для реагирования на инциденты:

- Установление очередности
- Локализация
- Расследование
- Анализ
- Отслеживание
- Восстановление

Первым этапом реагирования на инцидент сразу после сообщения о нем сотрудниками или выявления автоматизированной системой контроля безопасности должно быть **установление очередности** (triage). Установление очередности в данном контексте очень похоже на то, что делают врачи, когда им доставляют травмированного человека. Сначала они выясняют, действительно ли этот человек травмирован? Больно ли ему? Насколько сильно он пострадал? В каком лечении этот человек нуждается (операция, наложение швов или просто пластырь под зад)?

То же самое происходит в компьютерном мире. Сразу после получения информации об инциденте, мы анализируем его тяжесть и устанавливаем приоритеты в отношении этого инцидента. Процесс начинается с тщательного анализа информации о произошедшем событии для выяснения, действительно ли произошел инцидент и нужно ли приступать к процедуре его обработки. На одного из членов команды реагирования на инциденты должна быть возложена обязанность по анализу полученной информации об инциденте для выяснения, является ли она достоверной или ложной (например, вызванной ложным срабатыванием системы защиты). Если выясняется, что информация была ложной, сведения об этом регистрируются и процесс реагирования на инцидент на этом завершается. Однако если выясняется, что инцидент действительно произошел, он должен быть идентифицирован и классифицирован. Классификация инцидента должна быть произведена на основе уровня потенциального риска, который зависит от типа инцидента, его источника (внешний или внутренний), темпов его развития и возможности нанести ущерб. В свою очередь это определяет, какие уведомления требуются в рамках процесса эскалации, устанавливает границы и процедуры проведения расследования.



После выяснения серьезности инцидента, мы переходим к следующему этапу, которым является **локализация** (containment). К примеру, если врач выявил у больного туберкулез, он должен будет положить больного в изолятор, чтобы он никого не заразил. На этапе локализации должен быть минимизирован ущерб. В компьютерном мире, для этого может потребоваться отключить зараженный сервер от сети, внести изменения в настройки межсетевого экрана или полностью изолировать атакуемую систему от сети Интернет, чтобы остановить атакующего.

Правильная стратегия локализации дает команде реагирования на инциденты время для проведения надлежащего расследования и определения основной причины инцидента. Стратегии локализации должны быть основаны на категории атаки (внутренняя или внешняя), активах, которым нанесен вред, и их критичности. Какие стратегии локализации предпочтительны? Это зависит от конкретной ситуации. Стратегии локализации могут быть проактивными или реактивными. Какая из них лучше, зависит от окружения и категории атаки. В некоторых случаях, лучшим решением будет просто отключить систему от сети. Однако такой реактивный подход может привести к прерыванию обслуживания или ограничению функциональности критичных систем. При невозможности полной изоляции или локализации, можно использовать сегментацию сети для виртуальной изоляции системы (одной или нескольких). Граничные устройства также могут использоваться для предотвращения заражения одной системой других. Другой вариант реактивной стратегии предполагает анализ и внесение изменений в настройки правил межсетевого экрана или маршрутизатора с пакетной фильтрацией. Для снижения негативного воздействия также могут применяться списки контроля доступа. Однако по таким стратегиям локализации злоумышленник может понять, что его атака была замечена и предпринимаются контрмеры. Но что если вам нужно оставить пострадавшие системы подключенными к сети и делать вид, что вы не заметили атаку, чтобы собрать как можно больше информации для последующего анализа основных причин инцидента? Для этого вы можете использовать сети-приманки (или хосты-приманки), создав таким образом область, в которой злоумышленник может выполнять любые действия, с минимальным риском для компании. Для реализации такого решения следует проконсультироваться с юристами и высшим руководством компании, поскольку это может привести к правовым проблемам. Кроме того, компьютеры из сети-приманки могут быть использованы злоумышленником для атак на другие внутренние цели.

После того, как инцидент удалось локализовать, нужно собрать все относящиеся к этому инциденту данные, которые будут использоваться в процессе анализа и на последующих

этапах. Сбор данных производится на этапе **расследования**. Целью этого этапа, помимо сбора данных, является выяснение причин инцидента и максимально быстрое восстановление штатного функционирования систем и процессов. Руководство должно решить, будут ли для проведения расследования привлекаться сотрудники правоохранительных органов, будут ли доказательства собираться с целью судебного преследования, или достаточно будет просто закрыть соответствующую уязвимость. В штате большинства компаний нет команды экспертов по компьютерной криминалистике (forensics team), которые могли бы выполнить задачи по расследованию. Поэтому в случае подозрения о совершении преступления, если руководство компании не хочет обращаться к правоохранительным органам для проведения расследования, следует обратиться к внешним экспертам.

Мы более подробно рассмотрим вопросы проведения компьютерной криминалистической экспертизы в следующем разделе. Сейчас важно знать, что при расследовании необходимо соблюдать как требования политики компании, так и требования применимого законодательства.

ПРИМЕЧАНИЕ. Не путайте реагирование на инциденты (incident response) с компьютерной криминалистической экспертизой (computer forensics). Хотя оба они по сути являются расследованием, эти термины не являются синонимами. Компьютерная криминалистическая экспертиза имеет более высокий критерий доказанности (standard of proof) по сравнению с реагированием на инцидент. Компьютерная криминалистическая экспертиза проводится, когда требуются приемлемые для суда доказательства, в рамках экспертизы доказательства обрабатываются соответствующим образом.

После сбора всех относящихся к инциденту данных, нужно выяснить, что же произошло на самом деле, сложив вместе все имеющиеся части. Это является этапом **анализа**, на котором собираются дополнительные данные (журналы аудита, записи системы видеонаблюдения, отчеты о действиях людей, журналы работы систем), необходимые для выяснения, как произошел этот инцидент. Целью этого этапа является выяснение, кто это сделал, как он это сделал, когда он это сделал и почему. Руководство должно постоянно быть в курсе выполняемых мероприятий, поскольку именно ему нужно будет принимать важные решения о том, что теперь делать со всей этой неразберихой.

Члены команды, которые участвуют в проведении анализа, должны обладать различными навыками. Они также должны хорошо разбираться в системах, пострадавших в результате инцидента, уязвимостях систем и приложений, настройках сети и систем. Хотя образование имеет большое значение, ключевое значение для этих людей имеет практический опыт в сочетании со специализированным обучением. Одной из самых больших сложностей, с которыми сталкиваются такие специалисты, является динамичный характер журналов регистрации событий. Большинство интернет-провайдеров чистят или перезаписывают свои журналы регистрации событий через короткий промежуток времени, а время может быть потеряно момент инцидента. Может пройти не один час, прежде чем инцидент будет обнаружен. В некоторых странах существуют законы, обязывающие обеспечивать сохранность журналов регистрации событий в течение длительного времени. Однако такие законы создают сложности при хранении этих журналов (их объем может быть очень большим), а также ставят под угрозу неприкосновенность частной жизни.

После того как была собрана вся доступная информация и получены ответы на все вопросы, мы переходим к этапу **отслеживания** (tracking) (отслеживание может проводиться параллельно с анализом). Нужно определить, был ли источник этого инцидента внутренним или внешним, как злоумышленник смог проникнуть и получить доступ к активам. Если злоумышленник был внешним, команде следует связаться со своим интернет-провайдером, чтобы он оказал помощь в сборе данных и выявлении источника атаки. Часто это оказывается очень сложной задачей, поскольку нападающие перемещаются от одной системы к другой, что может потребовать обращаться за помощью сразу к нескольким

провайдером. В связи с этим, очень важно, чтобы у членов команды, выполняющих задачи по анализу и отслеживанию, были хорошие рабочие отношения с третьими сторонами, такими как интернет-провайдеры, другие команды реагирования, правоохранительные органы.

ПРИМЕЧАНИЕ. Все эти данные должны надлежащим образом документироваться и обрабатываться, чтобы они могли быть в дальнейшем использованы в качестве доказательств в суде.

После того, как нам удалось понять этот инцидент, мы переходим к этапу **восстановления** (recovery) (или **последующего контроля** (follow-up)), на котором мы проводим исправления, необходимые для того, чтобы подобные инциденты не повторялись. Для этого может потребоваться заблокировать определенные порты, отключить уязвимые службы или функции, переключить работу на другую площадку или установить патч. Более правильно называть это «процедурами последовательного восстановления» (following recovery procedures), поскольку беспорядочное внесение изменений в среду, может привести к дополнительным проблемам. В состав процедуры восстановления может входить восстановление системы с образа, восстановление данных с резервной копии, тестирование системы, выполнение правильных настроек.

Независимо от специфики процедур восстановления, перед тем, как поврежденная система будет возвращена в работу, вы должны сначала обеспечить, что она сможет противостоять другой такой атаке. Иначе хакеры быстро заметят, что уязвимая система снова вернулась в работу. Специально подготовленные сотрудники подразделения информационной безопасности должны предварительно провести проверку системы на наличие уязвимостей. Инструменты для тестирования уязвимостей, имитирующие выполнение реальной атаки, могут помочь повысить безопасность системы и защитить ее от различных атак, включая ту в результате которой она ранее была повреждена.

ПРИМЕЧАНИЕ. Никогда не следует доверять атакованной или зараженной системе, поскольку вы не знаете, какие на ней произошли изменения, и истинные масштабы повреждений. Некоторые вредоносные программы могут скрыто оставаться в системе. Система должна быть полностью восстановлена «с нуля», чтобы исключить все потенциальные угрозы.

Чему мы можем научиться в результате инцидента? Закрытие инцидента определяется характером или категорией этого инцидента, ожидаемых результатов реагирования на инцидент (например, возобновление работы бизнеса или восстановление системы), успех команды заключается в выяснении источника инцидента и основных причин его возникновения. Когда это определено и инцидент закрывается, целесообразно провести совещание команды с привлечением всех групп, подвергшихся влиянию инцидента, для ответа на следующие вопросы:

- Что произошло?
- Чему мы научились?
- Что мы сможем улучшить в следующий раз?

Команда должна еще раз взглянуть на этот инцидент и процесс его обработки, чтобы провести итоговый анализ. Информация по результатам этого совещания покажет, что нужно усовершенствовать в процессе реагирования на инциденты и какие изменения нужно внести в документы. Введение такого совещания в качестве одной из обязательных процедур процесса реагирования на инциденты, позволит команде собирать данные, которые в дальнейшем могут быть использованы в качестве метрик эффективности.

Вызывать ли полицию? Руководству необходимо принять решение, нужно ли привлекать правоохранительные органы для обработки инцидента, связанного с нарушением безопасности. Ниже перечислены вещи, которые нужно учитывать при принятии решения о привлечении правоохранительных органов:

- Компания теряет контроль над расследованием, им занимаются только правоохранительные органы
- Сохранение секретности не гарантируется, информация может стать достоянием широкой общественности

- Нужно учесть возможное влияние на репутацию (реакция на эту информацию клиентов, акционеров и т.д.).
- Сотрудники правоохранительных органов заберут все, что будет составлять доказательную базу. Все это будет недоступно компании в течение продолжительного времени (расследование может занять не один год, прежде чем дело дойдет до суда).

Нужно учесть и другие вопросы при разработке компанией процедур реагирования на инциденты, в том числе продумать, как инцидент будет объясняться средствам массовой информации, клиентам и акционерам. Для этого может потребоваться участие подразделения по связям с общественностью, кадровой службы (если в этом участвует персонал), департамента ИТ и юридического департамента. Компьютерные преступления могут иметь правовые последствия, которые не сразу заметны и к которым нужно подходить с осторожностью. Компания должна решить, как она будет доводить эту информацию до внешних лиц, чтобы они не восприняли ситуацию в совершенно ином свете.

Ссылки по теме:

- CERT Coordination Center
- 6 Steps to Incident Handling
- The University of Georgia Computer Security and Ethics
- Bibliography of Computer Security Incident Handling Documents, by Klaus-Peter Kossakowski, DFN-CERT (Germany)
- NIST Incident Handling Information

8. Компьютерная криминалистика и сбор доказательств

Компьютерная криминалистика (forensics) – это наука и искусство, она требует использовать специальные методы для восстановления, проверки подлинности и анализа электронных данных, связанных с компьютерными преступлениями. В ней объединяются компьютерные науки, информационные технологии и другие технические вопросы с законом. Возможно, вы слышали такие термины, как цифровая криминалистика (digital forensics), сетевая криминалистика (network forensics), обнаружение электронных данных (electronic data discovery), кибер-криминалистика (cyber forensics) и т.п. (ISC)² использует термин компьютерная криминалистика (computer forensics) в качестве синонима всех вышеперечисленных терминов. Компьютерная криминалистика применяется во всех случаях работы с доказательствами, представленными в цифровой или электронной форме, при хранении или передаче электронных данных. Одно время компьютерную криминалистику отделяли от сетевого анализа и анализа кода, но сейчас все это рассматривается в качестве *цифровых доказательств* (digital evidence).

Компьютерная криминалистика является достаточно новым направлением криминалистики. Из-за этого, а также из-за ее сложности, у многих компаний пока недостаточно навыков в этой области. Компьютерная криминалистика не относится непосредственно к компьютерному оборудованию или программному обеспечению. Это набор специальных процедур, предназначенных для реконструкции процесса работы на компьютере, анализа остаточных данных, проведения проверки подлинности данных с помощью средств технического анализа, исследования технических свойств данных. Это в корне отличается от работы системного администратора.

Люди, которые проводят компьютерные расследования, должны иметь необходимую квалификацию в этой области, они должны точно знать, что нужно искать. Перегрузка атакованной системы или даже простой просмотр файлов на ней, может разрушить доказательства, изменить метки времени на важных файлах, удалить следы, которые мог оставить преступник. Большинство цифровых доказательств имеет короткое время жизни,

поэтому они должны быть собраны как можно быстрее с учетом энергонезависимости мест их хранения. Другими словами, наиболее энергозависимые и недолговечные доказательства должны быть собраны первыми. В большинстве случаев, лучше всего отключить атакованную систему от сети, сделать дамп содержимого памяти, после чего выключить систему и сделать побитовый образ ее жесткого диска, а затем проводить экспертизу на основании этой копии. Работа с копией вместо реального диска системы позволит обеспечить сохранность доказательств в оригинальной системе даже в случае, если некоторые действия в процессе расследования разрушат или уничтожат данные. Сохранение в файл дампа содержимого оперативной памяти перед выполнением каких-либо других работ в системе или отключением питания, является очень важным шагом, поскольку в памяти могут находиться ценные улики. Это является одним из способов сбора очень энергозависимой информации. Однако это создает неоднозначную ситуацию, поскольку снятие дампа оперативной памяти или выполнение иного анализа работающей системы может привести к изменениям в системе. Какой бы метод не применялся экспертом для сбора цифровых доказательств, он обязан документировать все свои действия. Это является очень важным аспектом работы с доказательствами.

ПРИМЕЧАНИЕ. Команде экспертов по компьютерной криминалистике требуются специальные инструменты: блокнот для сбора доказательств, контейнеры, фотоаппарат, стикеры для пометки доказательств. Блокнот для сбора доказательств не должен быть простой тетрадкой, позволяющей незаметно вырывать страницы.

8.1. Международная организация по компьютерным доказательствам

Когда ранее в этом Доме мы рассматривали законодательство, мы отмечали, насколько важны стандартизированные отношения и подходы различных стран к компьютерным преступлениям, поскольку такие преступления часто переходят международные границы. То же самое верно и для компьютерной криминалистики. При работе с цифровыми доказательствами должен применяться единообразный подход, чтобы они могли использоваться в различных судах разных стран. Для этих целей была создана международная организация **IOCE** (International Organization on Computer Evidence – Международная организация по компьютерным доказательствам), задачей которой является разработка международных принципов, связанных с вопросами сбора и обработки цифровых доказательств, приемлемых для судов различных стран. В США есть похожая организация **SWDGE** (Scientific Working Group on Digital Evidence – Научная рабочая группа по цифровым доказательствам), целью которой является обеспечение единообразного подхода для всего сообщества экспертов по компьютерной криминалистике. Принципы, разработанные IOCE и SWDGE для стандартизированного подхода к работе с компьютерными доказательствами, имеют следующие характерные признаки:

- Согласованность со всеми правовыми системами
- Возможность использования общего языка
- Надежность
- Трансграничность
- Возможность подтверждения целостности доказательств
- Применимость к любым компьютерным доказательствам
- Применимость на любом уровне, в том числе индивидуальном, уровне государственных учреждений, государственном уровне

Принципы IOCE / SWDGE перечислены ниже:

1. При работе с цифровыми доказательствами должны быть применимы все основные процессуальные принципы и принципы компьютерной криминалистики.

2. Действия, выполняемые в процессе сбора цифровых доказательств, не должны изменять эти доказательства.
3. Допуск к оригинальным цифровым доказательствам может быть предоставлен при необходимости только лицу, прошедшему специальное обучение по работе с ними.
4. Все действия, связанные со сбором, использованием, хранением или передачей цифровых доказательств, должны быть надлежащим образом задокументированы, а документы должны быть сохранены и доступны для изучения.
5. Лицо несет ответственность за все действия в отношении цифровых доказательств, которые находятся в его распоряжении.
6. Любое учреждение, в обязанности которого входит сбор, использование, хранение или передача цифровых доказательств, несет ответственность за соблюдение этих принципов.

8.2. Мотивы, возможности и средства

Современная компьютерная преступность очень похожа на традиционную. Также как и при расследовании традиционного преступления, для понимания причин совершения компьютерного преступления необходимо понять мотивы, возможности и средства.

Мотив – это ответ на вопрос «кто» и «почему» совершает преступление. Мотивы могут быть обусловлены внешними или внутренними факторами. Человек может находиться в состоянии эмоционального возбуждения, испытывать страх или потребность в адреналине от совершения преступления - все это является факторами его внутреннего состояния. Примерами внешних факторов могут быть финансовые проблемы, болезнь члена семьи и т.п. Понимание мотива преступления является очень важным аспектом при определении круга людей, которые могли совершить его. Например, многие хакерские атаки на сайты с громкими именами осуществлялись лишь для того, чтобы факт нарушения их работы попал во все новости. Однако когда технологии достигнут достаточного уровня для успешного противодействия таким атакам, либо когда подобные факты перестанут привлекать такое внимание, люди перестанут проводить эти атаки, поскольку их мотивация снизится.

Возможность – это то, что делает возможным совершение преступления. Обычно возможности появляются тогда, когда существуют некоторые уязвимости или слабые места. Если у компании нет межсетевого экрана, хакеры имеют все возможности для выполнения своих действий в этой сети. Если компания не осуществляет управление доступом, не ведет или не контролирует журналы регистрации событий, сотрудники получают массу возможностей для хищения денег и обмана компании. После выяснения того, почему человек решил совершить преступление (мотив), нужно проанализировать, что именно дало возможность быть успешным этому преступлению (возможности).

Средства – это то, что нужно преступнику для обеспечения успеха своего преступления. Предположим, вам было поручено расследовать сложное хищение, произошедшее в финансовой компании. Если у вас после проведения предварительной работы осталось только трое подозреваемых, каждый из которых умел пользоваться мышью, клавиатурой и текстовым редактором, но один из них был программистом и системным аналитиком, вы можете сделать вывод, что именно этот человек имел гораздо больше средств для успешного совершения этого преступления, по сравнению с двумя другими.

8.3. Поведение компьютерных преступников

Как и традиционные преступники, компьютерные преступники имеют свои специфические методы работы (Modus Operandi). Другими словами, у каждого преступника есть свой, особый образ действий при совершении преступления, что может использоваться при расследовании для попытки установления личности преступника. Следовательно,

расследующему компьютерное преступление (в отличие от расследования традиционного преступления), необходимы хорошие знания информационных технологий. Например, методика работы (образ действий) компьютерного преступника может включать в себя использование определенных хакерских инструментов, нацеленность на определенные системы или сети, одинаковый стиль программирования, похожий текст отправляемых сообщений. В отдельных случаях, это позволяет выявить повторяющийся шаблон поведения. Знание методов работы компьютерных преступников и шаблонов их поведения может быть очень полезным в процессе расследования компьютерных преступлений.

Правоохранительные органы могут использовать эту информацию, например, для идентификации других преступлений, совершенных тем же преступником. Методы работы и шаблоны поведения могут давать информацию, которая будет очень полезной при проведении интервью и допросов, а также в суде.

Психологический анализ места преступления (определение профиля) также может проводиться с использованием сведений о методах работы и шаблонах поведения преступника. Определение профиля позволяет понять мысли преступника, что может помочь при определении его личности или, возможно, в определении использованного им инструмента для совершения преступления.

ПРИМЕЧАНИЕ. Принцип обмена Локарда также дает информацию, которую можно использовать при определении профиля. Этот принцип утверждает, что преступник что-то оставляет на месте преступления и что-то берет с собой. Этот принцип лежит в основе криминалистики. Даже для полностью цифрового преступления принцип обмена Локарда помогает в поисках преступника.

8.4. Специалисты по расследованию инцидентов

Специалисты по расследованию инцидентов (incident investigator, следователь) – это люди другой породы. Многие считают, что они пришли с другой планеты, но это пока не доказано. Они должны уметь выявлять подозрительную и необычную деятельность там, где другие ничего не замечают. Для этого им необходима профессиональная подготовка и большой опыт.

Специалист по расследованию инцидентов может выявить, например, такие подозрительные действия, как сканирование портов или попытки SQL-инъекций, найти важные улики в журнале регистрации событий. Необычную деятельность выявить сложнее. Она может заключаться в увеличении объема сетевого трафика, в участившемся пребывании сотрудника на рабочем месте в нерабочее время, выполнении нестандартных запросов к портам файлового сервера и т.п. Аналогично, если мать подростка почувствовала запах дыма от его пиджака, она может предположить, что он начал курить. Если он обычно каждый вечер играл на своем Xbox, а тут вдруг вечерами стал ходить «в библиотеку», его мать сразу заметит такое необычное поведение и может предположить, что у него появилась подруга.

Специалист по расследованию инцидентов должен хорошо разбираться в процедурах компьютерной экспертизы, сбора доказательств, знать, как проводить анализ ситуации, чтобы понять произошедшее, уметь найти важные улики в системных журналах.

Различные типы исследований, которые должен уметь выполнять специалист по расследованию инцидентов:

- **Сетевой анализ**
 - Анализ сетевого взаимодействия
 - Анализ журналов регистрации событий
 - Отслеживание маршрутов (path tracing)
- **Анализ носителей информации**
 - Работа с образами дисков

- Анализ времени создания, изменения, доступа (файлов)
- Анализ содержимого
- Анализ свободного дискового пространства
- Стеганография
- **Анализ программного обеспечения**
 - Обратный инжиниринг
 - Исследование вредоносного кода
 - Исследование эксплойтов

8.5. Процесс проведения компьютерной экспертизы

Чтобы компьютерная экспертиза (forensics) выполнялась стандартизованным образом, команда должна строго шаг за шагом следовать заранее определенным процедурам. Это позволит не упустить ничего важного и обеспечить приемлемость собранных доказательств для суда. У каждой команды или компании могут быть собственные процедуры, но все они, по существу, должны выполнять одни и те же вещи.

- Выявление
- Обеспечение сохранности
- Сбор
- Осмотр
- Анализ
- Предъявление
- Принятие решения

ПРИМЕЧАНИЕ. В процесс проведения компьютерной экспертизы включены все основные принципы криминалистики. В том числе определение места преступления, защита окружения от изменений и утраты улик, нахождение улик и потенциальных источников улик, сбор улик. При работе на месте преступления очень важно минимизировать влияние на окружение, однако нужно понимать, что полностью избежать такого влияния невозможно ни для традиционного, ни для компьютерного преступления. Важно минимизировать изменения и документировать все действия, причины их выполнения, а также их влияние на место преступления.

В процессе осмотра и анализа цифровых улик необходимо работать с образами носителей информации, содержащими *полные* копии данных с оригинальных носителей информации, а не с самими оригинальными носителями. Для этого должны формироваться побитовые копии каждого сектора оригинального носителя информации, включая удаленные файлы, свободное пространство и нераспределенные кластеры. Для создания таких образов могут использоваться специальные инструменты, такие как FTK Imager, EnCase, Safeback или утилита dd в Unix. Обычная функция копирования файлов не позволит создать копию всех областей диска, которые должны быть исследованы при выполнении компьютерной экспертизы.

Контроль места преступления. Независимо от того, является ли преступление традиционным или компьютерным, для обеспечения гарантий целостности улик очень важно контролировать все контакты с ними. Ниже приведены лишь некоторые из шагов, которые должны быть выполнены для защиты места преступления:

- Только уполномоченным лицам разрешается доступ к месту преступления. Эти лица должны иметь базовые навыки анализа места преступления.
- Внести в протокол информацию о тех, кто находится на месте преступления.
 - В суде целостность улик может быть подвергнута сомнению, если вокруг них крутилось слишком много людей.

- Внести в протокол информацию о том, кто последним взаимодействовал с системами.
- Если на месте преступления происходят изменения, необходимо документировать это. Изменения могут не оказывать негативного влияния на собранные улики, но они могут усложнять проведение расследования преступления.

С оригинальных носителей информации должны быть сделаны две копии: **основной образ** (контрольная копия, которая хранится в библиотеке) и **рабочий образ** (используется для проведения сбора и анализа доказательств).

Перед созданием этих образов, следователь должен убедиться, что новые носители информации были надлежащим образом очищены, т.е. не содержат никаких остаточных данных. Неоднократно возникали проблемы из-за того, что новый носитель информации содержал старые данные не очищенные производителем.

Следователь работает с дубликатом носителя информации, поскольку это обеспечивает сохранность оригинального доказательства, предотвращает его случайное изменение в ходе экспертизы и позволяет повторно создавать дубликаты в случае необходимости. В большинстве случаев данные на компьютере находятся на жестком диске и в энергозависимой памяти. Можно выделить следующие области хранения данных:

- Регистры и кэш-память
- Таблицы процессов и кэш ARP
- Содержимое системной памяти
- Временные файлы
- Данных на диске

Чтобы надлежащим образом собрать важные данные с компьютера или другого устройства, нужна большая осторожность и точность. Сотовые телефоны, коммуникаторы, USB-накопители, ноутбуки, карты памяти и другие устройства также могут содержать важные улики.

Получение доказательств с систем в процессе их функционирования, а также с сетевых хранилищ является более сложной задачей, поскольку вы не можете их отключить, чтобы сделать копию жесткого диска. Представьте реакцию руководителя ИТ-департамента, если вы попросите его выключить сервер базы данных или почтовый сервер. Образы подобных систем, а также систем, выполняющих шифрование информации «на лету», должны сниматься в процессе их работы.

Чтобы гарантировать неизменность и целостность исходного образа, нужно рассчитать хэш-функции для файлов и каталогов до и после проведения анализа.

ПРИМЕЧАНИЕ. В журналах регистрации событий должна храниться подробная информация обо всех действиях, системах, периферийных устройствах и их серийных номерах, а также информация о каждом действии экспертов. Это обеспечит возможность проверки доказательств и их применимость в суде. Также следует убедиться, что в компании документированы роли, которые выполняют системы.

ПРИМЕЧАНИЕ. В большинстве случаев, блокнот следователя не может использоваться в суде в качестве доказательства. Он может использоваться следователем, только чтобы освежить свою память о деталях произошедшего.

Инструменты для компьютерной криминалистики. При организации команды экспертов по компьютерной криминалистике, нужно позаботиться об обеспечении ее всеми необходимыми инструментами и вспомогательными средствами. Ниже перечислены наиболее часто используемые предметы, имеющиеся в наборах экспертов по компьютерной криминалистике:

- **Средства для документирования.** Метки, наклейки и формы со шкалой времени
- **Инструменты для разборки и извлечения.** Антистатические браслеты, щипцы, пинцеты, отвертки, кусачки и т.п.

- **Принадлежности для упаковки и транспортировки.** Антистатические мешки, сумки и ленты для доказательств, кабельные стяжки и т.п.

Следующим моментом, имеющим крайне важное значение, является обеспечение надлежащей **системы охраны вещественных доказательств** (chain of custody). Поскольку доказательства, связанные компьютерными преступлениями, могут быть очень "хрупкими" и легко могут стать непригодными для суда при неправильном обращении с ними, необходимо следовать строгим процедурам сбора, пометать каждый контейнер с доказательствами – без исключений. Кроме того, система охраны вещественных доказательств должна обеспечиваться на всех стадиях жизненного цикла доказательств, начиная с их выявления, заканчивая уничтожением, архивированием или возвращением владельцу.

Процесс изготовления копий данных должен соответствовать определенным стандартам, чтобы обеспечить гарантии качества и надежности. Для этой цели может использоваться специализированное программное обеспечение. Копии должны позволять провести независимую проверку их подлинности, а также должны быть защищены от неумелого обращения.

Каждый отдельный предмет, являющийся уликой, должен быть помечен каким-либо образом с указанием даты, времени, фамилии того, кто его собрал, а также номера дела, если оно уже заведено. При пометке носителей информации нужно позаботиться, чтобы нанесение этих пометок не привело к потере записанной на них информации. Каждый отдельный предмет должен быть запечатан в отдельный контейнер, который должен быть помечен с указанием такой же информации. Контейнер должен быть опечатан лентой для опечатывания доказательств. По возможности, на ленту должны быть нанесены надписи (например, личная подпись) таким образом, чтобы можно было легко выявить факт нарушения целостности этой ленты и вероятного несанкционированного доступа внутрь контейнера.

ПРИМЕЧАНИЕ. Система охраны вещественных доказательств требует, чтобы все доказательства были помечены с указанием информации о том, кто обеспечил их защиту, и кто это проконтролировал.

Провода и кабели должны быть помечены, должна быть сделана фотография системы с этими пометками, только после этого можно разбирать систему. Носители информации должны быть защищены от записи. В помещении для хранения доказательств не должно быть пыли, должна поддерживаться комнатная температура, низкая влажность, и, конечно же, вблизи не должно быть мощных магнитов или магнитных полей.

По возможности место преступления должно быть сфотографировано. Если компьютерная система была взломана физически (например, вскрыт системный блок компьютера), это также должно быть сфотографировано. Для работы с документами, бумагами и устройствами следует надеть тканевые перчатки, улики следует сначала поместить в контейнеры и опечатать их. Все носители информации, в том числе те, информация с которых была стерта, должны быть собраны, поскольку есть шансы, что информацию с них удастся восстановить.

Улики такого типа очень сложны в обращении, они могут быть легко стерты или уничтожены, поэтому их нахождение, обеспечение сохранности, сбор, осмотр, транспортировка и дальнейшая интерпретация являются очень важными. После того, как все улики надлежащим образом помечены, на каждом контейнере должны быть сделаны отметки ответственных за обеспечение охраны вещественных доказательств, а в общий протокол должны быть внесены записи обо всех событиях.

Чтобы преступление было успешно раскрыто, а преступник понес адекватное наказание, необходимы надежные доказательства. Компьютерная экспертиза – это искусство получения этих доказательств и их надлежащее сохранение, обеспечивающее их приемлемость для суда. Без правильного проведения компьютерной экспертизы, вероятность успешного предъявления доказательств компьютерного преступления в суде значительно снижается.

Наиболее распространенными причинами неправильного сбора доказательств являются: отсутствие команды реагирования на инциденты, отсутствие установленных процедур реагирования инциденты, плохо написанная политика, нарушения при организации системы охраны вещественных доказательств.

ПРИМЕЧАНИЕ. Система охраны вещественных доказательств – это история, которая показывает, как доказательства были собраны, сохранены, перевезены и проанализированы для предъявления в качестве доказательств в суде. Поскольку электронные доказательства могут быть легко изменены, только четко определенная система охраны вещественных доказательств может продемонстрировать, что этим доказательствам можно доверять.

Следующим шагом является анализ собранных улик. Следователи, расследующие компьютерные преступления, применяют следующие методы:

- Определение характеристик улики, в частности, может ли она применяться в качестве первичного или вторичного доказательства, ее источник, надежность и неизменность
- Сравнение улик из различных источников для определения хронологии событий
- Реконструкция событий, включая восстановление удаленных файлов и других действий в системе

Это может выполняться не только в лабораторных условиях, но и непосредственно на месте преступления, в полевых условиях – благодаря аппаратным копировщикам (write-blocker) и специализированному программному обеспечению для компьютерной криминалистики. Когда следователь анализирует улику в лаборатории, он проводит экспертизу «мертвых» данных (dead forensics), поскольку он работает только со статичными данными. «Живая» экспертиза (live forensics), которая выполняется в полевых условиях, имеет дело в том числе и с энергозависимыми данными (volatile data). Если не удастся найти улики, следует обратиться к опытному эксперту по компьютерной криминалистике, чтобы он помог обнаружить их.

В завершение, интерпретация результатов анализа улик должна быть представлена соответствующей стороне. Это может быть судья, адвокат, генеральный директор, совет директоров. Очень важно представить эти результаты в виде, который будет понятен нетехнической аудитории. Как специалист по безопасности, вы должны уметь объяснять такие вещи простым языком, используя метафоры и аналогии. Естественно, улики и сведения, которые являются секретной информацией или коммерческой тайной компании следует сообщать только уполномоченным на ознакомление с такой информацией сторонам. Среди них может быть юридический департамент или внешний юрист, помогающий в расследовании.

Руководство по компьютерной криминалистике Австралийской команды реагирования на компьютерные чрезвычайные происшествия

- Минимизировать работу с оригинальными данными и вероятность их повреждения
- Протоколировать все действия и пояснять изменения
- Следовать пяти правилам доказательства (Применимость, Подлинность, Полнота, Точность и Убедительность)
- Обращаться за квалифицированной помощью при работе с доказательствами и их анализе, если у вас недостаточно знаний, опыта или возможностей
- Придерживаться политики безопасности вашей компании и получать письменные разрешения для проведения расследования компьютерного преступления
- Быстро сохранять максимально точный образ системы
- Быть готовым давать показания в суде
- Удостовериться, что ваши действия повторимы
- Приоритезировать ваши действия по сбору улик - начинать следует с наименее

устойчивых, а заканчивать устойчивыми уликами

- Не запускать никаких программ на системе, которая является потенциальным доказательством
- Действовать этично и добросовестно при проведении расследования компьютерного преступления, не пытаться причинить какой-либо вред

Ссылки по теме:

- To Catch a Thief
- Computer Forensics Checklist
- Long List of Types of Forensics
- PDA and Cell Phone Forensics

8.6. Что является приемлемым для суда?

Компьютерные журналы регистрации событий очень важны во многих областях мира ИТ. Обычно они используются для выявления и устранения проблем, а также чтобы понять события, имевшие место в определенный момент времени. Если компьютерные журналы регистрации событий предполагается использовать в качестве доказательства в суде, они должны быть собраны в ходе обычной работы бизнеса. В большинстве случаев, компьютерные документы рассматриваются как **показания с чужих слов** (hearsay), т.е. являются вторичными доказательствами. Показания с чужих слов, обычно не принимаются в суде без основных доказательств, которые могут подтвердить их точность, достоверность и надежность. Основными доказательствами могут быть показания сотрудника компании, в обязанности которого входит настройка системы аудита событий и сбор журналов регистрации событий. Этот сотрудник должен выполнять эти обязанности в рамках своей повседневной работы в компании, а не делать это специально для суда. Ценность доказательства зависит от искренности и компетентности их источника.

Важно продемонстрировать, что журналы регистрации событий и все остальные доказательства не были подделаны или изменены каким-либо образом. Это может обеспечить система охраны вещественных доказательств (chain of custody). Существует ряд инструментов для расчета контрольной суммы или хэш-функции файлов журналов регистрации событий, которые позволят сразу же выявить любые изменения в этих файлах.

В процессе сбора доказательств может возникнуть проблема, вызванная ожиданиями сотрудников по сохранению неприкосновенности их частной жизни. Подозреваемый в совершении компьютерного преступления сотрудник может утверждать, что файлы на его компьютере являются его личными и они не могут быть доступны сотрудникам правоохранительных органов и суду. Поэтому очень важно, чтобы компания проводила регулярное обучение и повышение осведомленности персонала в области безопасности, требовала от сотрудников ознакомливаться под роспись с нормативными документами, устанавливающими требования по правильному использованию компьютеров и оборудования компании, использовала иные методы повышения осведомленности (например, сообщения, появляющиеся при регистрации сотрудника на компьютере, экранные заставки, баннеры на внутреннем сайте, плакаты и т.п.). Это является ключевым моментом при установлении того, что пользователь не имеет права на неприкосновенность частной жизни при использовании оборудования компании.

К примеру, CERT Advisory рекомендует при регистрации сотрудника на компьютере показывать на экране сообщение следующего содержания:

Эта система предназначена для использования только уполномоченными пользователями. Использование этой системы лицами, не имеющими соответствующих полномочий или с превышением своих полномочий, а также действия уполномоченных пользователей, выполняемые с нарушением действующих в компании требований, регистрируется системой мониторинга и

подлежат последующему анализу.

Продолжение использования этой системы означает безусловное согласие пользователя с таким мониторингом, а также с тем, что при необходимости данные мониторинга могут использоваться в качестве доказательств и передаваться компанией в правоохранительные органы.

Это явное предупреждение подкрепляет обоснованность и юридическую значимость претензий, которые могут быть предъявлены сотруднику или неуполномоченному лицу, поскольку продолжение использования системы после просмотра этого предупреждения означает, что человек признает политику безопасности компании и дает ей разрешение для выполнения мониторинга и использования результатов мониторинга в качестве доказательства.

Любое доказательство имеет свой жизненный цикл. Важно, чтобы участвующие в расследовании лица хорошо понимали все этапы жизненного цикла рассматриваемого доказательства и учитывали их в процессе расследования.

Жизненный цикл доказательства включает в себя:

- Выявление и сбор
- Обеспечение сохранности и транспортировка
- Предъявление в суде
- Возврат доказательства жертве или владельцу

В суде могут использоваться различные виды доказательств, в частности письменные, устные, компьютерные, визуальные и звуковые. Устные доказательства – это показания свидетеля. Визуальные или звуковые – это, как правило, доказательства, созданные во время самого преступления или сразу после него (например, видеозапись, запись на диктофон, фотоснимки).

В глазах закона не все доказательства равны, некоторые виды доказательств более значимы или имеют больший вес, чем другие. В следующих разделах кратко описываются способы, с помощью которых доказательства могут быть классифицированы и оценены.

Наилучшее доказательство

Наилучшее доказательство (best evidence) – это основное доказательство, используемое в суде, поскольку оно обеспечивает наибольшую надежность. Примером того, что может быть квалифицировано в качестве наилучшего доказательства, является оригинал подписанного договора. Устные доказательства не могут считаться наилучшим доказательством, поскольку они недостаточно надежны и не являются юридически значимым документом. При этом устные доказательства могут использоваться для интерпретации документов.

Вторичное доказательство

Вторичное доказательство (secondary evidence) менее надежно и не имеет такой доказательной силы, как наилучшее доказательство. Вторичными доказательствами могут быть, например, устные показания свидетеля, копии документов и т.п.

Прямое доказательство

Прямое доказательство (direct evidence) само может доказать некий факт, для этого не требуются какие-либо дополнительные доказательства. При использовании прямых доказательств не нужны предположения и домыслы. Примером прямых доказательств являются показания свидетеля, который видел совершение преступления. Хотя устные доказательства будут вторичными по своей природе (т.е. решение суда не может быть основано только на них), они являются прямыми доказательствами, что означает, что адвокату не обязательно предоставлять другие дополнительные доказательства. Прямое

доказательство часто основывается на информации, полученной с помощью пяти чувств свидетеля (слух, зрение, осязание, обоняние, вкус).

Неопровержимое доказательство

Неопровержимое доказательство (conclusive evidence) является бесспорным, оно не может быть подвергнуто сомнению. Неопровержимое доказательство имеет очень большую силу и не требует подтверждения.

Косвенное доказательство

Косвенное доказательство (circumstantial evidence) может доказать промежуточный факт, который затем может быть использован для вывода или предположения о существовании другого факта. На основании промежуточного факта судья или присяжные могут логично предположить наличие первичного факта. Например, если подозреваемый сказал другу, что он собирается атаковать сайт eBay, дело не может быть основано только на этом факте, поскольку такое доказательство является косвенным. Однако если через час после того, как подозреваемый сообщил об этом своему другу, сайт eBay был атакован, присяжные могут предположить на основе этого, что именно подозреваемый совершил это преступление.

Подтверждающее доказательство

Подтверждающее доказательство (corroborative evidence) – это вспомогательное, подкрепляющее доказательство, помогающее доказать идею или точку зрения. Само по себе подтверждающее доказательство использоваться не может, оно может служить дополнительным инструментом, помогающим доказать основные доказательства.

Доказательство, основанное на предположении

Когда свидетель дает показания, в соответствии с правилами (opinion rule) он должен сообщать только факты по данному вопросу, но не свое мнение об этих фактах. В данном случае речь не идет о показаниях эксперта, т.к. эксперт в основном привлекается именно для того, чтобы получить его профессиональное мнение по соответствующему вопросу. Большинство юристов вызывают экспертов для дачи показаний, позволяющих стороне защиты или обвинения лучше понять предмет, чтобы они могли, в свою очередь, помочь судье и присяжным лучше понять относящиеся к делу вопросы.

Доказательство, основанное на показаниях с чужих слов

Показания с чужих слов (hearsay evidence) – это устные или письменные доказательства, они исходят из вторых рук и непосредственно не могут обеспечить точность и надежность. Если свидетель в своих показаниях говорит, что он слышал, как кто-то говорил что-то, это не является никаким фактом, существует слишком много переменных, которые могут скрыть правду. Если документы были сформированы в рамках обычных бизнес-процедур, они могут быть приемлемыми для суда. Однако если документы были подготовлены специально для предъявления в суде, они могут быть отнесены к категории показаний с чужих слов.

При принятии решения о приемлемости доказательства для суда, рассматриваются следующие вопросы:

- Процедуры сбора и сопровождения доказательств
- Подтверждение того, каким образом удалось избежать ошибок
- Определение ответственного за сохранность доказательств и определение его навыков по работе с ними
- Разумное объяснение:
 - Почему были выполнены определенные действия
 - Почему определенные процедуры были пропущены

Важно, чтобы доказательства были достоверными, полными, достаточными, надежными и относящимися к рассматриваемому делу. Эти характеристики лежат в основе доказательств, их соблюдение обеспечивает приемлемость доказательств для суда.

Чтобы доказательство было **достоверным** (authentic) и **относящимся** к делу (relevant), оно должно разумно и здраво соотноситься с фактами. Если судья решает, что билеты человека на поезд не могут рассматриваться в качестве доказательства в суде по делу об убийстве, это означает, что судья постановил, что эти билеты не имеют отношения к делу. После этого адвокат не может упоминать о них в суде.

Чтобы доказательство было **полным** (complete), оно должно содержать все детали. Должны быть предъявлены все детали доказательства, даже оправдательные.

Чтобы доказательство было **достаточным** (sufficient) и **правдоподобным** (believable), оно должно быть достаточно убедительным, чтобы убедить разумного человека в справедливости доказательства. Это означает, что доказательство не может быть истолковано субъективно. Достаточным не может считаться доказательство, в котором легко усомниться.

Чтобы доказательство было **надежным** (reliable) и **точным** (accurate), оно должно соответствовать фактам. Доказательство не может быть надежным, если оно основано на мнении человека или копии документа, поскольку при этом возникает слишком много возможностей для ошибки. Надежное доказательство – это доказательство, основанное на фактах, оно не является косвенным.

Если какое-либо доказательство признано подлинным, полным, обоснованным, надежным и относящимся к делу, оно также должно быть юридически допустимо, т.е. должно быть получено законным путем. Доказательство не должно быть получено в результате незаконного обыска и изъятия, несанкционированного контроля, или получено под принуждением. Иначе доказательство будет аннулировано, как только оно попадет в суд.

8.7. Наблюдение, обыск и изъятие

Существует два основных вида наблюдения, применяемого для выявления компьютерных преступлений: физическое и компьютерное наблюдение. Для **физического наблюдения** (physical surveillance) используются камеры безопасности, охранники, система видеонаблюдения, которые могут зафиксировать доказательства. Физическое наблюдение также может осуществляться с использованием агента под прикрытием, который может собрать информацию о деятельности подозреваемого, его семье и друзьях, личных привычках, что может дать очень важные сведения для расследуемого дела.

Компьютерное наблюдение (computer surveillance) – это аудит журналов регистрации событий, пассивный перехват информации с использованием сетевых снифферов, клавиатурных перехватчиков, систем перехвата информации, передаваемой по каналам связи и т.п. В большинстве стран активный мониторинг может проводиться только при наличии ордера на обыск, а для легального контроля за действиями сотрудника, требуется заранее предупредить его о том, что его действия могут контролироваться таким образом.

Деятельность по обыску (search) и изъятию (seizure) доказательств может быть очень сложной, это зависит от того, что нужно найти и где. Например, американские граждане находятся под защитой Четвертой поправки, запрещающей незаконный обыск и изъятие, поэтому правоохранительные органы должны иметь достаточные основания и заранее запросить соответствующий ордер на обыск от судьи или суда. При этом обыск может проводиться лишь в тех местах, которые указаны в ордере. Четвертая поправка не распространяется на деятельность частных лиц, если только они не действуют в качестве агентов полиции. Таким образом, например, если руководитель компании предупредил всех сотрудников, что по решению руководства все файлы с их рабочих компьютеров могут быть

удалены в любое время, и этот руководитель не является сотрудником (или агентом) полиции, сотрудники компании не могут утверждать, что их права, предусмотренные Четвертой поправкой, были нарушены. Руководитель таким решением возможно нарушит некоторые законодательные акты по защите неприкосновенности частной жизни, но не Четвертую поправку.

В некоторых случаях, сотрудник правоохранительных органов может изъять улики вне областей, указанных в ордере, например, если подозреваемый пытается уничтожить улики. Иными словами, если существует вероятность, что потенциальные доказательства могут быть уничтожены, сотрудник правоохранительных органов имеет право изъять их, чтобы предотвратить их уничтожение. Это называется *срочными обстоятельствами* (exigent circumstances), по данному факту в дальнейшем судья должен будет принять решение, было ли изъятие правильным и законным, что необходимо для решения вопроса о применимости собранных таким образом доказательств для суда. Например, если у сотрудника полиции есть ордер на обыск в гостиной подозреваемого (но не в других помещениях), а он видит, что подозреваемый прячет кокаин в туалете, сотрудник полиции может изъять кокаин, хотя туалет не был указан в ордере на обыск.

После сбора доказательств, необходимо обеспечить их охрану для обеспечения гарантий их целостности.

Существует очень тонкая грань между заманиванием и провокацией, когда речь идет о фиксации действий подозреваемого. *Заманивание* (enticement) является законным и этичным, тогда как *провокации* (entrapment) не являются ни законными, ни этичными. В мире компьютерных преступлений, хост-приманка (honeypot) является хорошим примером для демонстрации разницы между заманиванием и провокацией. Компании размещают специальные системы в своих экранированных подсетях, которые либо эмулируют сервисы, которыми часто пытаются воспользоваться злоумышленники, либо эти сервисы на них включены реально. Это делается с расчетом на то, что если злоумышленник взломает сеть компании, он пойдет напрямик к приманке, а не к реальным системам, используемым для работы компании. Система-приманка имеет множество открытых портов и работающих сервисов, содержащих уязвимости, которыми может воспользоваться злоумышленник, поэтому такая система с большой вероятностью привлечет внимание злоумышленника. С помощью системы-приманки компания может зафиксировать действия атакующего, а затем использовать собранную информацию для его судебного преследования.

Действия в приведенном примере законны, пока компания не переходит грань между заманиванием и провокацией. Примером провокации может быть ссылка на веб-странице, на которой указано, что при переходе по этой ссылке пользователь сможет бесплатно загрузить тысячи файлов MP3. Однако когда пользователь переходит по ссылке, он попадает на систему-приманку, компания регистрирует все его действия и пытается затем преследовать его в суде. В результате такой провокации компания не сможет доказать в суде, что подозреваемый имел намерение совершить преступление, это лишь покажет, что он был успешно обманут.

8.8. Проведение опросов и допросов

После проведения всех мероприятий по наблюдению, обыску и изъятию, вероятно, потребуется провести опрос свидетелей и допросы подозреваемых. Проведение опроса (interviewing) – это наука и искусство, поэтому он должен проводиться профессионалом, имеющим необходимую подготовку. Кроме того, нужно учитывать, что опрос свидетеля может проводиться только после консультации с юристом. Однако это не значит, что вы, как специалист по информационной безопасности, не будете участвовать в этом процессе. Вас могут попросить предоставить вводную информацию для проведения опроса или поприсутствовать в процессе опроса для пояснения технических вопросов. При необходимости, компания может быть назначено ответственное лицо, в обязанности

которого будет входить проведение опросов и допросов. Темы для обсуждения и вопросы должны быть подготовлены заранее, вопросы должны задаваться спокойно и систематично, поскольку целью опросов и допросов является получение доказательств для суда.

Сотрудник, проводящий допрос, должен занимать более высокую позицию в штатной структуре компании по сравнению с допрашиваемым. Вряд ли вице-президент компании испугается и будет откровенничать с рядовым специалистом. Допрос должен проводиться в приватной обстановке, подозреваемому должно быть достаточно комфортно и удобно. Если предполагается показывать подозреваемому улики, они должны показываться по очереди, либо храниться в папке. Нет необходимости перед допросом зачитывать подозреваемому его права, если допрос проводит не сотрудник правоохранительных органов.

Сотрудник, проводящий допрос, не должен допустить, чтобы подозреваемый обманул его, вынудил отказаться от важной информации, относящейся к расследованию, или сбежал до начала судебного процесса.

8.9. Несколько различных видов мошенничества

В различных категориях компьютерных преступлений используются различные методики. В следующих разделах мы рассмотрим некоторые виды компьютерного мошенничества и злоупотреблений.

"Салями"

При проведении *атаки «салями»* (salami attack) преступник совершает несколько мелких преступлений, рассчитывая на то, что общая картина останется незамеченной. Атака «салями» чаще всего происходит в бухгалтерии компании, наиболее распространенным примером такой атаки является незаконное списание небольших денежных сумм со множества различных счетов и их аккумуляция на отдельном счете для последующего снятия. При этом преступник надеется, что хищения очень мелких сумм останутся незамеченными. Например, сотрудник банка, имеющий возможность вносить изменения в программное обеспечение, используемое при выполнении банковских операций, может внести в программное обеспечение процедуру, которая будет ежемесячно списывать 1 рубль с расчетного счета каждого клиента банка и переводить его на счет этого сотрудника. Если в этом банке обслуживаются 50 000 банковских счетов клиентов, злоумышленник может получать доход около 600 000 рублей в год.

Подделка данных

Подделка данных (data diddling) – это внесение изменений в существующие данные. Чаще всего данные подделываются еще до того, как они загружаются в приложение, либо сразу после окончания их обработки и их вывода из приложения. Примером подделки данных может быть ввод заведомо неверной информации кредитным работником – клиент берет в банке кредит на сумму 1 000 000 рублей, а кредитный работник вводит сумму 1 500 000 рублей, а затем переводит 500 000 рублей со счета клиента на свой счет. Другим примером может быть кассир, который принимает у клиента 1 000 рублей, а чек пробивает на 800 рублей и 200 рублей забирает себе.

Заведомо ложная информация может вводиться в системы и приложения по множеству различных причин, но чаще всего это делается для того, чтобы завысить доходы и активы, либо занижить расходы и обязательства. Иногда руководители компаний делают это с целью обмана акционеров, кредиторов и партнеров, а руководители подразделений – с целью обмана руководства компании.

Этот вид преступлений достаточно распространен, хотя его несложно предотвратить с помощью правильного управления доступом, ведения учета, выполнения контрольных мероприятий, аудита, разделения обязанностей и установления лимитов на операции. Это один из примеров того, что сотрудники компании (инсайдеры) могут быть более опасны, чем

внешние преступники.

Чрезмерные привилегии

Чрезмерные привилегии (excessive privileges) являются очень распространенной проблемой безопасности, поскольку их чрезвычайно сложно контролировать в большой и сложной среде. Чрезмерными являются привилегии (права, разрешения) пользователя компьютера, которые не требуются ему для выполнения своих обязанностей. Если пользователю достаточно иметь возможность чтения и печати документов с файлового сервера, ему не должен предоставляться полный доступ к этому файловому серверу. Типичным примером является следующая ситуация. Руководитель одного из подразделений бухгалтерии для выполнения своих обязанностей имел полный доступ ко всем данным на сервере, включая финансовую информацию. Но затем он перешел из бухгалтерии в научно-исследовательский отдел, но ранее предоставленные ему права доступа не были аннулированы, поскольку большинство компаний не имеют предназначенных для таких случаев процедур. Это называется «расползание прав» (authorization creep). Теперь этот руководитель имеет полный доступ данным учета и к информации об исследованиях – это и есть чрезмерные привилегии. Если он будет недоволен какими-либо действиями компании, он сможет нанести компании гораздо больше ущерба, чем если бы его права доступа были надлежащим образом ограничены.

Сниффинг паролей

Сниффинг паролей – это перехват сетевого трафика для получения паролей, передаваемых между компьютерами. В Интернете в свободном доступе есть ряд инструментов, которые выполняют такую функциональность. Перехватить пароль не легко, т.к. он передается только в процессе аутентификации пользователя в домене или при получении доступа к ресурсу. Некоторые системы и приложения отправляют пароли по сети в открытом виде, но таких систем остается все меньше. Большинство современных систем и приложений вообще никогда не передают паролей. Например, рабочая станция пользователя может выполнять одностороннюю функцию хэширования введенного пользователем пароля и отправлять по сети только полученное в результате значение хэша для аутентификации на сервере. При этом на сервере хранится файл, содержащий хэш-значения паролей всех пользователей, а не сами пароли. Когда к нему обращается система для проверки пароля пользователя, сервер сравнивает полученное значение хэша со значением, хранящимся в этом файле.

Однако многие из инструментов, позволяющих перехватывать пароли, также могут проводить их взлом. Часто именно с этого и начинается компьютерное преступление.

IP-спуфинг

В локальных сетях и Интернете используются IP-адреса, они выполняют ту же функцию, которую в реальном мире выполняют номера домов и названия улиц, позволяющие найти дорогу из одного места в другое. Каждому подключенному к сети компьютеру присваивается IP-адрес, необходимый для того, чтобы пакеты знали, откуда они пришли и куда они направляются. Однако многие злоумышленники не хотят, чтобы кто-то мог определить их реальное местоположение, поэтому они изменяют свой IP-адрес в сетевых пакетах, чтобы выдавать себя за другого. Они могут делать это вручную, либо автоматически – с помощью специализированных программных инструментов. Это называется **IP-спуфингом** (IP spoofing). Большинство атак выполняются с поддельных IP-адресов, что дает жертвам мало шансов найти злоумышленника и систему, которую он использовал для атаки. Одной из причин, по которой выполнять IP-спуфинг так легко, является то, что протокол Интернета (IP) был разработан в то время, когда о безопасности редко задумывались. Тогда разработчики все внимание уделяли функциональности, они даже не могли себе представить те атаки, которые в наше время осуществляются с использованием разработанных ими протоколов.

В Домене 05 мы рассматривали протоколы IPv6 и IPSec в IPv4, которые позволяют эффективно бороться со спуфингом, но для этого необходимо перейти на использование этих протоколов. Такое изменение очень трудно провести, т.к. оно затрагивает работу миллионов людей.

ПРИМЕЧАНИЕ. Спуфинг также называют атакой маскардинга (masquerading attack). Маскардинг – это попытка выдать себя за кого-то другого.

Разгребание мусора

Разгребание мусора (dumpster diving) – это поиск в мусоре компании или конкретного человека, выброшенных документов, носителей информации и других ценных вещей, которые могут быть использованы для атаки на этого человека или компанию. Для этого злоумышленнику нужно получить физический доступ в помещение, однако, как правило, места, в которых хранится мусор не очень хорошо охраняются. Разгребание мусора неэтично, но оно не является противозаконным. Однако несанкционированное проникновение в помещение нарушает закон, а оно может быть необходимо для получения доступа к мусору. Законодательство, касающееся этого вопроса, может различаться в разных странах и регионах.

Люди, занимающиеся промышленным шпионажем, могут устраивать настоящие рейды на мусорные контейнеры компаний, чтобы найти служебную и конфиденциальную информацию. Также в мусорных контейнерах злоумышленники могут найти выброшенные документы с данными банковских карт, документы по организации работы внутренней компьютерной и телефонной сетей компании и многое другое.

Перехват побочных излучений

Побочные излучения и методы их перехвата злоумышленниками мы рассматривали в Домене 02 в разделе "Защита от утечки информации по техническим каналам". Обычно, каждое электрическое устройство излучает электрические волны в окружающую среду. Эти волны передают информацию, аналогично тому, как работают беспроводные технологии. Они могут передаваться на значительное расстояние, в зависимости от мощности сигнала, применяемых материалов и окружающих объектов. Злоумышленники могут использовать специальные устройства для перехвата этих волн для попытки получения доступа к информации, которая не была предназначена для них.

Для этого злоумышленникам нужны специальные инструменты, которые настраиваются на частоту, на которой передаются нужные волны. Кроме того, злоумышленник должен находиться в непосредственной близости от здания компании, в котором установлено устройство, излучающее волны. Компании, обрабатывающие настолько критичную информацию, что злоумышленники готовы будут перехватывать ее таким образом, преодолевая множество сложностей, как правило, используют в своей работе специализированное экранированное компьютерное оборудование, излучающее очень небольшое количество электрических сигналов. Также компании могут использовать специальные материалы в стенах здания, который не пропускает через себя эти типы электрических волн.

Обычно такие атаки происходят в кино и шпионских романах. Однако и в обычной жизни существует технология, которая позволяет производить подобные атаки без использования шпионской техники – это беспроводные сети. Если компания использует беспроводную сеть, она может воспользоваться специальными механизмами и настройками, предотвращающими перехват злоумышленниками трафика своих сотрудников. К сожалению, не все компании используют эти механизмы и настройки. Это позволяет любому пользователю с ноутбуком и беспроводной сетевой картой припарковаться на автостоянке компании и прослушивать ее сетевой трафик. Беспроводные технологии и их безопасность рассматривались нами в Домене 05.

Перехват информации, передаваемой по каналам связи

Большинство коммуникационных сигналов уязвимы к тому или иному виду прослушивания или перехвата. Обычно это можно делать незаметно, это называется пассивной атакой. Для перехвата передаваемой информации применяются такие инструменты, как сотовые сканеры, радиоприемники, микрофоны, магнитофоны, сетевые снифферы, а также устройства для прослушивания телефонных разговоров.

ПРИМЕЧАНИЕ. Пассивная атака не оказывает влияния на процесс передачи данных. Примером пассивной атаки может быть перехват (wiretapping) или прослушивание (eavesdropping). Активные атаки напротив, вмешиваются в процесс передачи данных. Примером может быть DoS-атака или проникновение во внутреннюю сеть.

Во многих странах существуют законы, запрещающие прослушивание переговоров других людей. Прослушивание может быть законным только с согласия самого прослушиваемого лица, либо по решению суда. Чтобы получить разрешение суда, сотрудники правоохранительных органов должны обосновать наличие у них оснований предполагать преступную деятельность и необходимость проведения прослушивания для доказательства этого. Такие ограничения устанавливаются в целях защиты прав человека на частную жизнь.

Ссылки по теме:

- Security Exploits
- Security Focus
- BlackHat Media Archives
- “Computer Crime and Security,” Virginia Tech Department of Computer Science, Professionalism in Computing Digital Library

9. Этика

Этика основана на различных вопросах и принципах. Она относится к различным ситуациям, по-разному интерпретируемым разными людьми. Из-за этого вопросы этики часто становятся предметом дискуссий. Однако отдельные варианты этики являются менее спорными, по сравнению с другими, поэтому их соблюдение проще требовать от людей. (ISC)² требует, чтобы все сертифицированные специалисты по безопасности безоговорочно соблюдали ее Кодекс Этики. Если CISSP умышленно нарушает Кодекс, его сертификат может быть отозван.

Ниже приведен краткий перечень его основных положений. Каждый кандидат CISSP должен ознакомиться с полной версией Кодекса Этики, прежде чем пытаться сдать экзамен.

- Действовать достойно, честно, справедливо, ответственно и законно, защищать общество
- Усердно работать, предоставлять компетентные услуги, продвигать профессию безопасности
- Содействовать проведению исследований, обучать, оценивать сертификацию.
- Избегать необоснованного страха и сомнений, не соглашаться с применением плохих практик
- Препятствовать применению небезопасных практик, сохранять и укреплять целостность общественной инфраструктуры
- Соблюдать и выполнять все договорные обязательства, явно выраженные или подразумеваемые, давать разумные советы
- Избегать любых конфликтов интересов, уважать доверие, которое другие возлагают

на вас, и выполнять только те работы, для выполнения которых у вас есть достаточно квалификации

- Поддерживать свои навыки в актуальном состоянии, не принимать участия в мероприятиях, которые могли бы повредить репутации других специалистов по безопасности

Между законом и этикой существует интересная связь. Чаще всего, законы основаны на этике, они внедряются для того, чтобы обеспечить соблюдение всеми соответствующих этических норм. Однако законы учитывают не все вопросы. Для того, чего нет в законодательстве, нужна этика. Некоторые действия могут не нарушать закон, однако это не обязательно означает, что они этичны.

Компании следует разработать руководство по компьютерной и деловой этике. Это руководство может быть частью справочника сотрудника компании, эта тема может затрагиваться в процессе обучения персонала.

Существуют этические заблуждения, которые многие пытаются использовать в компьютерном мире, чтобы оправдать свои неэтичные действия. Эти заблуждения вызваны различием во взглядах людей на одни и те же вопросы, различиями в интерпретации законов и правил. Ниже приведены примеры таких заблуждений:

- Хакеры хотят только учиться и совершенствовать свои навыки. Многие из них не получают никакой прибыли от своих действий, поэтому их деятельность не должна рассматриваться как незаконная или неэтичная.
- Первая поправка защищает и дает гражданам США право разрабатывать компьютерные вирусы.
- Информация должна использоваться совместно, свободно и открыто, поэтому обмен конфиденциальной информацией и коммерческой тайны должен быть законным и этичным.
- Хакинг не причиняет никому реального вреда.

9.1. Институт компьютерной этики

Институт компьютерной этики (Computer Ethics Institute) является некоммерческой организацией, которая помогает продвигать передовые технологии этичными способами. Институт компьютерной этики разработал Десять заповедей компьютерной этики:

1. Не используй компьютер с целью причинения вреда другим людям
2. Не вмешивайся в работу компьютеров других людей
3. Не пытайся найти что-нибудь в файлах на компьютерах других людей
4. Не используй компьютер для воровства
5. Не используй компьютер для лжесвидетельства
6. Не копируй и не используй коммерческое программное обеспечение до его оплаты
7. Не используй ресурсы чужого компьютера без разрешения или компенсации
8. Не присваивай себе результаты интеллектуальной деятельности других людей
9. Подумай о социальных последствиях использования программы, которую ты написал, или системы, которую ты спроектировал
10. Используй компьютер таким образом, чтобы соблюсти интересы сограждан

9.2. Совет по архитектуре Интернета

Совет по архитектуре Интернета (Internet Architecture Board, IAB) является координационным комитетом по проектированию, инжинирингу и управлению Интернетом. В его обязанности входит архитектурный надзор за деятельностью Специальной комиссии интернет-разработок (Internet Engineering Task Force, IETF), надзор за созданием новых стандартов Интернета, внесение изменений в RFC (Request for Comments).

IAB выпускает документы по вопросам этики использования сети Интернет. Он рассматривает Интернет в качестве ресурса, для которого очень важен вопрос доступности и который используется на пользу для широкого круга людей. В основном работа IAB связана с недопущением безответственных действий в сети Интернет, которые могут поставить под угрозу его существование или отрицательно повлиять на других. Он рассматривает Интернет как великий дар, и работает, чтобы защитить его для всех, кто зависит от него. IAB видит использование сети Интернет, как привилегию, которая должна рассматриваться именно с такой точки зрения и использоваться с уважением.

Следующие действия IAB рассматривает как неэтичные и неприемлемые:

- Умышленно пытаться получить несанкционированный доступ к ресурсам сети Интернет
- Препятствовать возможности использования сети Интернет
- Тратить ресурсы (человеческие, компьютерные и другие) на бесцельные действия
- Нарушать целостность компьютерной информации
- Разглашать чужие персональные данные
- Небрежно проводить эксперименты в масштабах всей сети Интернет

IAB обещает взаимодействовать с государственными учреждениями для принятия любых мер, необходимых для защиты Интернета. В частности для этих целей могут использоваться новые технологии, методы и процедуры, призванные сделать Интернет более устойчивым к нарушениям. Существует баланс между повышением защиты и сокращением функциональности. Одной из основных целей Интернета является обеспечение свободных потоков информации, которые не должны ограничиваться, в связи с этим IAB должен быть логичным и гибким в своих подходах, а также в тех ограничениях, которые он пытается реализовать. Интернет является общим инструментом, поэтому все должны работать вместе, чтобы защитить его.

ПРИМЕЧАНИЕ. RFC 1087 «Этика и Интернет» концептуально описывает поведение, которое IAB считает неэтичным и неприемлемым.

9.3. Программа корпоративной этики

Многие законодательные акты требуют, чтобы компании разрабатывали этические правила и соответствующие программы корпоративной этики. Это вызвано множеством «скользких» ситуаций, произошедших в прошлом, о которых знало (и косвенно поощряло) руководство компаний, даже если они не признали этого. Программа корпоративной этики задает «тон сверху», руководители должны не только обеспечить соблюдение определенных в ней этических норм подчиненными сотрудниками, но и сами обязаны соблюдать их. Основной целью является недопущение явного или неявного использования лозунга «успех любой ценой» в культуре компании. Существуют ситуации, в которых возникают предпосылки для проявления неэтичного поведения. Если зарплата генерального директора компании зависит от стоимости акций компании, он может найти способы для искусственного завышения цены на акции, от чего могут пострадать инвесторы и акционеры компании. Если премии менеджеров зависят от объемов продаж и при этом они сами рассчитывают эти объемы,

вполне вероятно, что их расчеты быстро перестанут отражать реальность. Если сотрудник может получить премию за экономию бюджета, он может выбрать самых дешевых (но ненадежных) производителей и/или модели, которые будут только причинять головную боль компании и ее клиентам. Хотя вопросы этики многим кажутся весьма абстрактными, заставляющими нас культурно выражаться в общественных местах и т.п., в действительности эти вопросы очень важны, они должны быть реализованы в корпоративной среде компании, в ее бизнес-процессах и стилях управления.

Руководящие принципы FSGO (Federal Sentencing Guidelines for Organizations) - это этические нормы, соблюдение которых ведет к сокращению числа правонарушений и нарушений обязательств. Они были обновлены и дополнены в 2004 году, в них были включены требования по активному участию высшего руководства и членов Совета директоров компании в реализации программы корпоративной этики. Это обеспечивает выполнение принципа должной осмотрительности (due diligence) и позволяет выявлять преступные действия, а также предотвращать их. Ряд положений SOX используют аналогичные подходы, но только в отношении бухгалтерского учета и достоверности корпоративной отчетности.

Ссылки по теме:

- Internet Architecture Board
- Computer Security Institute
- Corp-Ethics
- Society of Corporate Ethics

10. Резюме

Вопросы законодательства, этики и проведения расследований являются очень важной частью компьютерной и информационной безопасности. К сожалению, компании редко о них задумываются, пока они не сталкиваются с компьютерными преступлениями. Эти вопросы необходимо рассматривать, если мы всерьез хотим бороться с компьютерной преступностью и добиваться адекватного наказания компьютерных преступников.

В настоящее время законодательство в этой области находится на начальном этапе развития, суды предпринимают первые попытки рассмотрения компьютерных преступлений. Пока еще очень мало прецедентов в этой области, что затрудняет интерпретацию действий преступника и выбор адекватного наказания. Однако во многих странах законодательство в этой области быстро развивается, появляются новые законы, позволяющие правильно интерпретировать преступную деятельность, помогающие работе правоохранительных органов, а также защищающие интересы граждан. Еще не так давно хакерские атаки проводились просто для развлечения, но ужесточение законодательства в этой области и появление организованной компьютерной преступности, нацеленной на извлечение реальной прибыли, достаточно быстро свело подобные «развлечения» на нет.

Специалисты по безопасности должны хорошо знать и понимать законодательство и иные требования в области компьютерной безопасности, уметь правильно реализовать эти требования в компании, в которой они работают. Они должны уметь надлежащим образом проинформировать руководство компании и клиентов об их обязанностях, хорошо понимать границы своей деятельности.

Тест

Вопросы экзамена CISSP являются концептуальными, поэтому они сформулированы соответствующим образом. Задачей кандидата является выбор наилучшего из всех представленных вариантов ответа. Среди вариантов ответа может не быть идеального ответа на поставленный вопрос - кандидат должен выбрать лучший ответ из имеющихся вариантов.

1. Что из перечисленного ниже Совет по архитектуре Интернета (IAB) считает неэтичным?

- ☐ A. Создание компьютерных вирусов
- ☐ B. Ввод информации на веб-страницу
- ☐ C. Выполнение тестирования на проникновение на узлы в Интернете
- ☐ D. Нарушение интернет-коммуникаций

2. Что из перечисленного ниже является наукой о компьютерах и компьютерных технологиях, и какое отношение она имеет к преступлениям?

- ☐ A. Компьютерная криминалистика (forensics)
- ☐ B. Анализ уязвимостей компьютера
- ☐ C. Обработка инцидентов
- ☐ D. Критерии компьютерной информации

3. Что должен сделать эксперт по компьютерной криминалистике сразу после изъятия компьютера?

- ☐ A. Нужно поставить на компьютере специальные отметки и поместить в контейнер, а затем пометить сам контейнер
- ☐ B. Посыпать порошком для снятия отпечатков пальцев
- ☐ C. Скопировать образы всех дисков
- ☐ D. Убрать улику в сейф

4. Почему так сложно расследовать компьютерные преступления и искать компьютерных преступников?

- ☐ A. Законодательство по защите неприкосновенности частной жизни защищает людей от проведения подобных расследований в отношении них
- ☐ B. Для поиска компьютерных преступников требуется специальное оборудование и инструменты
- ☐ C. Преступники могут скрывать свою личность и перемещаться между различными сетями
- ☐ D. Полномочия полиции не распространяются на Интернет

5. Как называется защита доказательств и учет всех лиц, кто получал доступ к ним в процессе проведения расследования?

- ☐ A. Правило наилучшего доказательства (best evidence)
- ☐ B. Показания с чужих слов (hearsay)
- ☐ C. Обеспечение безопасности доказательств
- ☐ D. Система охраны вещественных доказательств (chain of custody)

6. Какие сложности возникают в процессе выявления и сбора компьютерных доказательств, которые предполагается использовать их в суде?

- ☐ A. Большинство компьютерных доказательств являются нематериальными
- ☐ B. Большинство компьютерных доказательств искажены
- ☐ C. Большинство компьютерных доказательств зашифрованы
- ☐ D. Большинство компьютерных доказательств являются материальными

7. Система охраны вещественных доказательств (chain of custody) содержит сведения о том, кто получал доступ к доказательству и _____.

- ☐ A. Кто обеспечивал их защиту и кто их похитил
- ☐ B. Кто работал с ними и повредил их
- ☐ C. Кто обеспечил их защиту и кто проверил это
- ☐ D. Кто проверил их и кто сделал их копию

8. Что должен сделать специалист, проводящий расследование, перед тем, как отключить атакованную систему?

- ☐ A. Извлечь жесткий диск и сделать его резервную копию
- ☐ B. Сделать копию содержимого оперативной памяти на диск
- ☐ C. Отключить ее от сети
- ☐ D. Сохранить данные из очереди печати и временных файлов

9. Почему созданные на компьютере документы обычно не считаются надежными (reliable) доказательствами?

- ☐ A. Это первичные (наилучшие) доказательства
- ☐ B. Очень сложно выявить произошедшие в них изменения
- ☐ C. Это подтверждающие (corroborative) доказательства
- ☐ D. Они не применимы в уголовном праве, их можно использовать только в гражданском праве

10. Что из перечисленного ниже является необходимой характеристикой доказательства, позволяющей обеспечить возможность его использования в суде?

- ☐ A. Оно должно быть реальным (real)
- ☐ B. Оно должно заслуживать внимание
- ☐ C. Оно должно быть надежным (reliable)
- ☐ D. Оно должно быть важным

11. Если компания умышленно оставляет уязвимость в одной из своих систем, надеясь зафиксировать попытки воспользоваться этой уязвимостью, как это называется?

- ☐ A. Реагирование на инцидент
- ☐ B. Провокация (entrapment)
- ☐ C. Нарушение закона
- ☐ D. Заманивание (enticement)

12. Если сотрудник компании подозревается в участии в компьютерном преступлении, какое подразделение компании должно быть привлечено к расследованию?

- ☐ A. Кадровая служба
- ☐ B. Юридическая служба
- ☐ C. Служба внутреннего аудита
- ☐ D. Подразделение по расчету заработной платы

13. К какой категории доказательств относятся диски и иные носители информации, на которые записаны копии оригинальных доказательств?

- ☐ A. Первичное (наилучшее) доказательство
- ☐ B. Надежное (reliable) и достаточное (sufficient) доказательство
- ☐ C. Показания с чужих слов (hearsay)
- ☐ D. Неопровержимое (conclusive) доказательство

14. Если компания не информировала своих сотрудников, что она может осуществлять мониторинг их действий, и не имеет политики в отношении проведения такого мониторинга, что может делать компания?

- ☐ A. Не осуществлять мониторинг действий сотрудников
- ☐ B. Осуществлять мониторинг только в нерабочее время
- ☐ C. Получить ордер на проведение мониторинга действия сотрудника
- ☐ D. Осуществлять мониторинг, поскольку это позволяет делать законодательство

15. Что является одной из причин сложности судебного преследования компьютерных преступников?

- ☐ A. Нет надежного способа перехвата электронных данных
- ☐ B. Компьютерные доказательства не являются наилучшими доказательствами
- ☐ C. Компьютерные преступления не всегда соответствуют категориям традиционной преступности
- ☐ D. Сложно организовать законный перехват информации (wiretapping)

16. Какое из приведенных ниже правил содержится в большинстве законов по защите неприкосновенности частной жизни?

- ☐ A. Люди имеют право на удаление любой информации о них, если они не хотят, чтобы эту информацию о них кто-то знал
- ☐ B. Организации не обязаны контролировать точность имеющихся у них данных
- ☐ C. Организации обязаны предоставлять государственным учреждениям доступ к имеющимся у них данным
- ☐ D. Организации не могут использовать собранные данные для целей, отличных от целей, для которых они были собраны

17. Какое из перечисленных ниже утверждений не является правильным в отношении разгребания мусора (dumpster diving)?

- ☐ A. Это законно
- ☐ B. Это незаконно
- ☐ C. Это является брешью в обеспечении физической безопасности
- ☐ D. Это сбор данных из мест, из которых люди не ожидают утечки информации

18. В каких случаях блокнот следователя может быть приемлемым доказательством для суда?

- ☐ A. Он не может быть приемлемым, следователь может использовать его только чтобы освежить память при даче показаний
- ☐ B. Когда нет свидетелей
- ☐ C. Когда он запрошен судом для изучения вопросов, которые задавал следователь
- ☐ D. Когда нет других материальных доказательств

19. Какие шаги предпринимают законодатели, чтобы усовершенствовать законодательство для более успешной борьбы с компьютерной преступностью?

- ☐ A. Дорабатывают законы по защите неприкосновенности частной жизни
- ☐ B. Расширяют понятие собственности, включая в него компьютерные данные
- ☐ C. Требуют обязательного страхования компаний от ущерба в результате компьютерных преступлений
- ☐ D. Пересматривают законодательство, касающееся трансграничных вопросов

20. В каких случаях руководству может быть предъявлено обвинение в халатности?

- ☐ A. Если оно следует трансграничному законодательству (transborder laws)
- ☐ B. Если оно не сообщает о произошедших компьютерных преступлениях и не пытается осуществлять судебное преследование злоумышленников
- ☐ C. Если оно информирует пользователей, что в отношении них могут осуществляться процедуры мониторинга
- ☐ D. Если оно не использует на практике принцип должной заботы (due care) для защиты ресурсов

Домен 09. Безопасность приложений.

Первоочередной целью разработки приложений и компьютерных систем является, как правило, реализация функциональности, а не обеспечение безопасности. Чтобы получить лучшее от обоих направлений, безопасность должна проектироваться и разрабатываться одновременно с функциональностью. Безопасность должна быть интегрирована в ядро продукта, она должна обеспечиваться на всех уровнях. В противном случае, когда безопасность реализуют для уже разработанного продукта, защитные меры снижают функциональность, а безопасность обеспечивается не в полном объеме, оставляя существенные уязвимости.

1. Важность программного обеспечения

Средства безопасности прикладных систем могут реализовываться различными способами и с различными целями. Они могут осуществлять контроль ввода, обработки, межпроцессного взаимодействия, доступа, результатов, а также интерфейсов между системой и другими программами. При разработке средств безопасности приложений нужно помнить про потенциальные риски, использовать различные модели угроз и результаты анализа рисков. Целью является предотвращение нарушений безопасности, снижение количества уязвимостей и возможностей для повреждения данных. Средства безопасности могут быть превентивными, детективными или исправительными. Они могут реализовываться в виде административных и физических мер, однако чаще всего они являются техническими.

Используемые средства безопасности приложений зависят от самого приложения, его целей, целей обеспечения безопасности, политики безопасности приложения, типов обрабатываемых данных, порядка их обработки, а также от окружения, в котором будет работать приложение. Для коммерческого приложения с закрытым кодом, которое будет работать только в закрытых доверенных средах, может потребоваться меньше средств безопасности, чем для приложения, которое будет передавать финансовые транзакции между различными компаниями через сеть Интернет. Основным моментом здесь является понимание потребностей приложения в безопасности, реализация правильных механизмов и средств безопасности, тщательное тестирование этих механизмов, а также их интеграция в приложение, использование структурированной методологии разработки, применение безопасных и надежных методов распространения. Выглядит просто, не так ли? Увы, это не просто. Разработка безопасных приложений или операционных систем является крайне сложной задачей. Действительно безопасных приложений очень немного.

2. Где нужно размещать безопасность?

Сегодня проблемы безопасности чаще всего рашаются с помощью таких защитных средств, как межсетевые экраны, системы выявления вторжений (IDS), контентная фильтрация, антивирусное программное обеспечение, сканеры уязвимостей и многого другого. Мы опираемся на все это множество защитных средств в основном потому, что используемое нами программное обеспечение содержит множество уязвимостей. Это приводит к тому, что внешний периметр безопасности является целостным и укрепленным, однако внутренняя среда и программное обеспечение содержат большое количество уязвимостей, которые несложно использовать при получении доступа во внутреннюю сеть.

В действительности, первопричиной большинства уязвимостей являются недостатки самого программного обеспечения. Ниже приведены несколько основных причин, поясняющих, почему сейчас чаще используются средства защиты периметра, а не обеспечение безопасности при разработке программного обеспечения:

- В прошлом, при разработке программного обеспечения не уделялось внимания вопросам безопасности, поскольку не было такой потребности. Из-за этого и сейчас многие программисты не задумываются о вопросах безопасности и не используют

методы безопасного программирования

- Большинство специалистов по безопасности не являются разработчиками программного обеспечения
- Многие разработчики программного обеспечения, не уделяют достаточного внимания вопросам безопасности
- Производители программного обеспечения стараются как можно быстрее вывести свои продукты на рынок, говоря в первую очередь об их функциональности, но не о безопасности
- Компьютерное сообщество привыкло получать программное обеспечение с ошибками, а затем применять патчи
- Покупатели программного обеспечения не могут контролировать недостатки в нем, поэтому они вынуждены обеспечивать защиту периметра

Концентрация внимания на функциональности и принятие быстрых решений оказывают негативное влияние на современном этапе компьютерной эволюции. Двадцать лет назад, когда использовались мейнфреймы, много безопасности не требовалось, т.к. лишь немногие люди знали, как они работают, а пользователи использовали для доступа к ним простые терминалы, посредством которых нельзя было ввести вредоносный код в мейнфрейм, среда была закрытой. Большинство протоколов и платформ, которые мы используем сейчас, были разработаны в те времена, когда угрозы и атаки не были распространены, и строгие меры безопасности не были нужны. Однако с тех пор эволюция компьютеров и программного обеспечения продвинулась очень далеко. Высокий спрос на компьютерные технологии и различное программное обеспечение повысил спрос на программистов, проектировщиков систем, администраторов и инженеров. Спрос на таких специалистов стал очень большим, что привело в эту отрасль целую волну людей, не имевших достаточного опыта. Недостаток опыта, быстрое изменение технологий, а также рыночные гонки, добавили проблем с обеспечением безопасности, необходимость в которой не всегда понимали.

Многие винят крупных производителей за поставки программного обеспечения, полного недостатков и ошибок, однако нужно понимать, что ими движет потребительский спрос. Всего десять лет назад (а зачастую и сегодня), мы требуем от разработчиков все больше и больше функциональности. А с точки зрения функциональности, разработчики проделали прекрасную работу. Только в последние лет семь, клиенты начали требовать от разработчиков в том числе и безопасность. Однако программисты не были обучены безопасному программированию, операционные системы и приложения не были с самого начала построены на основе безопасных архитектур, а имеющиеся процедуры разработки программного обеспечения не были ориентированы на безопасность, разработчики интегрировали средства безопасности в свои приложения на поздних этапах разработки, что существенно увеличивало стоимость и приводило ко множеству компромиссов. Конечно, производители программного обеспечения должны стремиться лучше делать свою работу, предоставляя нам безопасные продукты, однако мы должны понимать, что безопасность является относительно новым требованием, что вызывает у разработчиков значительные сложности.

В этом Домене мы сделаем попытку показать, как учесть вопросы безопасности в самом программном обеспечении и процессе его разработки. Это требует перехода от *реактивного* к *проактивному* подходу при решении проблем безопасности, чтобы избежать этих проблем или, по крайней мере, минимизировать их количество. На Рисунке 9-1 показан применяемый в настоящее время способ решения проблем безопасности.

Традиционный процесс обеспечения безопасности



Рисунок 9-1. Традиционный процесс выпуска программного обеспечения на рынок и обеспечения его безопасности

3. Различные среды имеют различные потребности в обеспечении безопасности

В наше время перед администраторами сетей и администраторами безопасности стоят очень сложные задачи: они должны интегрировать различные приложения и компьютерные системы, успевая за потребностями своей компании, расширяющей свою функциональность и внедряющей новейшие компоненты, которые руководство требует как можно быстрее закупать и начинать использовать. Это обусловлено необходимостью для компаний идти в ногу со временем, обеспечивая свое присутствие в сети Интернет с помощью веб-сайта с возможностью приема заказов через Интернет, проведение платежей по банковским картам, создания экстрасетей с партнерами и т.п. Это быстро может привести к проблемам с протоколами, устройствами, интерфейсами, проблемам совместимости, ошибкам при маршрутизации и коммутации, проблемам с управляемостью и многому другому.

Причем подразумевается, что для всего этого будет обеспечена безопасность. Это требует глубокого понимания имеющейся среды – что она из себя представляет и как она работает. Без этого невозможно внедрять в нее новые технологии осмысленным и управляемым образом.

Времена, когда компании разрабатывали простые веб-страницы и размещали их в Интернете для иллюстрации своей продукции и услуг, давно прошли. Сегодня компании разрабатывают сложные и функциональные веб-приложения, имеющие трехзвенную архитектуру и работающие с использованием промежуточного программного обеспечения. Сложность сетей и приложений постоянно растет, отслеживание ошибок и нарушений безопасности в

них становится очень сложной задачей.

Модель клиент/сервер. Архитектура клиент/сервер позволяет создавать прикладные системы, разделенные между несколькими платформами, использующими различные операционные системы и аппаратные средства. Клиентская часть запрашивает определенные сервисы, а серверная часть выполняет эти запросы. Сервер выполняет обработку данных и возвращает клиенту результаты обработки. Клиентская часть реализует интерфейсные элементы приложения и взаимодействует с пользователем, а серверная часть выполняет всю фоновую обработку, которая, как правило, является наиболее трудоемкой.

4. Среда и приложения

Программные средства безопасности могут быть реализованы в операционной системе, приложении или в системе управления базами данных (СУБД). Как правило, средства безопасности реализуются на всех указанных уровнях и используются совместно, дополняя друг друга. Каждый уровень имеет свои сильные и слабые стороны, но если все они хорошо изучены, правильно настроены и работают согласованно, можно избежать многих сценариев и разновидностей нарушения безопасности. Однако во многих случаях полагаются только на средства безопасности, реализованные в операционной системе. Такой подход имеет существенный минус, поскольку операционная система может эффективно контролировать, управлять и ограничивать доступ субъекта только к объектам в рамках самой операционной системы, но она далеко не всегда может делать это в приложениях. Если в программном коде приложения существуют недостатки в обеспечении безопасности, на уровне операционной системы крайне сложно будет реализовать эффективную защиту от компрометации приложения посредством этой уязвимости. Операционная система – это среда для работы приложений, не следует ожидать от нее учета всех нюансов работы различных приложений и реализованных в них механизмов.

С другой стороны, средства безопасности, реализованные в приложениях и СУБД, являются очень специфическими и могут обеспечить защиту только в рамках самих этих приложений и СУБД. Приложение может обеспечить защиту данных, разрешив выполнять ввод информации только определенным образом, ограничивая доступ пользователей к данным, хранящимся в критичных областях базы данных и т.д. Но оно не может запретить пользователю записывать фиктивные данные в таблицу ARP (Address Resolution Protocol), т.к. за работу этой таблицы отвечает операционная система и ее сетевой стек. У средств безопасности операционной системы и приложения есть свое место и свои ограничения. Основная задача заключается в том, чтобы понять, где заканчивается область действия одних средств безопасности, чтобы настроить и ввести в действие другие средства безопасности.

Сейчас безопасность обеспечивается в основном за счет специализированных программных и аппаратных продуктов безопасности, а также устройств защиты периметра сети, но не за счет средств безопасности, встроенных в приложения. Указанные продукты безопасности могут охватывать широкий спектр приложений, они могут иметь централизованную консоль управления. Однако такой подход не всегда обеспечивает необходимый уровень детализации, он не учитывает возможности нарушения безопасности, вызванные недостатками в используемых разработчиком процедурах разработки программного обеспечения. Межсетевые экраны и механизмы контроля доступа могут обеспечить определенный уровень защиты, не позволяя атакующим произвести атаку переполнения буфера, но реальная защита должна обеспечиваться на уровне основного источника проблем – недостатков программного кода самого приложения. Для этого разработчиками должны быть внедрены безопасные процедуры разработки программного обеспечения.

5. Безопасность и функциональность

Программирование – сложная профессия. Программист должен учитывать множество возможных источников проблем, которые могут оказать негативное влияние на безопасность. Такими источниками проблем может быть сам код приложения,

взаимодействие процедур, использование глобальных и локальных переменных, входящие данные, полученные от других программ, исходящие данные, отправляемые в другие приложения. Нужно попытаться предсказать возможные ошибки при вводе данных пользователями, ошибки в расчетах, установить соответствующие механизмы контроля и ограничения. Во многих случаях, попытки предусмотреть все «что-если» и проявление осторожности при программировании, могут привести к снижению общей функциональности приложения. А ограничение функциональности может, в свою очередь, ограничить сферу применения приложения и привести к снижению доли рынка и прибыли производителя. Чтобы избежать этого, всегда следует соблюдать определенный баланс между функциональностью и безопасностью. Однако следует учитывать, что для разработчиков программного обеспечения (и большинства их клиентов) пока наиболее важной является функциональность.

Программистам и архитекторам программного обеспечения необходимо найти золотую середину между необходимой функциональностью программы, требованиями к безопасности, а также механизмами, которые должны быть реализованы для обеспечения этой безопасности. Это еще больше усложняет и без того непростую задачу.

В большинстве приложений осуществляется обмен данными между различными частями программы, обмен данными с другими программами, с операционной системой, принимаются введенные пользователями данные. Каждый из маршрутов передачи (ввода) данных должен быть учтен, нужно проанализировать и протестировать каждый возможный сценарий взаимодействия, каждый вариант ввода данных. Только такой подход сможет обеспечить реально высокое качество приложения. Важно, чтобы была обеспечена возможность тестирования каждого модуля в отдельности, а также различных модулей при их взаимодействии между собой. Столь глубокий анализ и тестирование позволят сделать продукт более безопасным, заранее выявив недостатки, которые могут быть использованы злоумышленниками в дальнейшем.

6. Типы, форматы и размер данных

Все мы слышали об уязвимостях, приводящих к возможности проведения атаки переполнения буфера. Это далеко не новая проблема. Однако она и сейчас не потеряла своей актуальности и многие атаки по-прежнему основаны на ней.

Мы рассматривали переполнение буфера в Домене 03 и говорили, что такая атака может быть осуществлена в случае, если программный код не проверяет фактическую длину принимаемых входных данных. Специально подготовленные атакующим данные, в действительности содержащие команды, могут переполнить выделенный для них буфер, что позволит этим командам выполниться в привилегированном режиме и даст атакующему возможность получить контроль над системой. Если программист пишет программу, которая ожидает входящие данные, объемом не более 5KB, он должен правильно реализовать это в коде, выделив для хранения данных буфер необходимого объема и предусмотрев функцию проверки объема реально полученных данных. Даже если атакующий передаст этой программе более 5KB данных, программа должна автоматически отбросить лишние данные. Иначе атакующий может отправить 5KB данных и прибавить к ним еще 50KB кода, содержащего вредоносные команды, которые будут обработаны процессором.

Длина – это не единственное, о чем должны беспокоиться программисты, когда они разрабатывают компоненты программы, принимающие входные данные. Данные должны иметь правильный тип и находиться в нужном формате. Если программа ожидает получение символов ASCII, она не должна принимать шестнадцатеричные значения или Unicode.

Помимо этого, принимаемое значение должно быть корректным. Если программа запрашивает у пользователя ввод суммы, которую он хочет перевести со своего расчетного счета, она не должна позволять ввести в это поле текстовую строку. Принимаемые

программой данные, должны быть в правильном формате, программист должен реализовать процедуры, которые будут контролировать вводимые пользователем данные, чтобы предотвратить очевидные ошибки, а не начинать проведение расчетов с заведомо некорректными данными.

Рассмотренные примеры являются весьма упрощенными по сравнению с тем, с чем реально сталкиваются программисты. Тем не менее, они наглядно демонстрируют, почему программное обеспечение должно разрабатываться с учетом проверки правильности входящих данных, их типа, формата и длины, чтобы обеспечить безопасность и надежную работу.

7. Проблемы внедрения приложений и использования настроек "по умолчанию"

Как многие знают, большинство приложений по умолчанию устанавливаются с настройками, которые, как правило, далеко не безопасны. После установки программы нужно включить и настроить функции безопасности. Например, в отношении безопасности Windows NT было высказано не мало критики, однако эта система может быть настроена для обеспечения безопасности многими различными способами. Просто сразу после установки механизмы безопасности в ней не настроены. Это связано с тем, что настройки безопасности тесно связаны с особенностями среды, в которую интегрируется система, а также с тем, что такой подход обеспечивает более «дружеский» процесс установки системы. Представьте, что вы установили новый продукт, который при дальнейших попытках настройки для интеграции с другими приложениями постоянно выдает сообщение «Доступ запрещен».

При установке программного или аппаратного продукта безопасности, по умолчанию должны устанавливаться права «Нет доступа». Например, если администратор устанавливает новый межсетевой экран с пакетной фильтрацией, он не должен разрешать передачу никаких сетевых пакетов до тех пор, пока администратор специально не предусмотрел соответствующее разрешение в правилах межсетевого экрана. Однако это требует, чтобы администратор хорошо понимал, как нужно настраивать межсетевой экран, чтобы он смог эффективно работать в реальной среде. Существует очень тонкое равновесие между безопасностью, функциональностью и удобством эксплуатации. Если приложение чрезвычайно удобно для пользователя, оно, вероятнее всего, настолько же небезопасно.

Чтобы сделать безопасное приложение действительно удобным для пользователя, разработчику, как правило, требуется выполнить большой объем дополнительной работы: предусмотреть возможные ошибки пользователей, создать дополнительные диалоговые окна, шаблоны, мастера, написать пошаговые инструкции. В свою очередь, это может существенно «раздуть» код, что может стать причиной непредвиденных проблем. Эта «лишняя» работа и дополнительные проблемы не нужны производителям, поскольку обычно это не позволяет заработать дополнительные деньги, а имеет обратный эффект.

ПРИМЕЧАНИЕ. В последних версиях Windows многие сервисы по умолчанию отключены, пользователь должен специально включать их по мере необходимости. Это значительно ближе к реализации принципа «отсутствие доступа по умолчанию» (default with no access), но Microsoft еще есть, к чему стремиться.

Ошибки при внедрении и настройке программного обеспечения являются распространенной проблемой, которая становится причиной большинства нарушений безопасности в сетевой среде. Многие люди не осознают, что большое количество служб (многие из которых в действительности не нужны) включены по умолчанию сразу после установки системы. Эти службы могут позволить злоумышленникам получить важную информацию, которая поможет им при проведении атаки. А некоторые службы фактически открывают дверь в саму систему. Служба NetBIOS может быть использована для получения доступа к общим ресурсам в среде Windows, служба Telnet позволяет удаленному пользователю получить доступ к командной оболочке, другие службы также могут предоставлять доступ без

ограничений. На многих системах работают сервисы FTP, SNMP, IRC (Internet Relay Chat), которые не обеспечивают никакой безопасности, тем более, что в действительности они зачастую не используются. Многие такие службы включаются по умолчанию и остаются включенными и доступными злоумышленникам, если администратор не позаботится об их отключении или ограничении доступа к ним.

Производители программного обеспечения в первую очередь думают об удобстве для пользователей и функциональность продукта, поэтому, как правило, этот продукт, установленный "по умолчанию", обеспечивает очень низкий уровень безопасности. Также нужно учитывать, что производители не могут знать, какой уровень безопасности требуется в средах их заказчиков, в которые будет установлен продукт. Именно специалист, производящий установку продукта, должен знать, как правильно настроить его для обеспечения необходимого уровня защиты.

Еще одной проблемой безопасности является множество систем, на которых не установлены патчи. После выявления проблемы безопасности, производители разрабатывают патчи и обновления, чтобы учесть и исправить эти уязвимости. Однако выпущенные ими патчи часто не устанавливаются на уязвимые системы. Причины этого могут быть разными: администраторы могут быть не в курсе выявленных уязвимостей и выпущенных патчей, они не всегда понимают важность установки патчей, либо они могут опасаться, что установка патча приведет к возникновению других проблем. Это очень распространенные причины и все они имеют одинаковый результат – небезопасные, уязвимые системы. Для большинства реально эксплуатируемых злоумышленниками уязвимостей, на тот момент уже имеются патчи, выпущенные разработчиком несколько месяцев (или даже лет!) назад.

К сожалению, иногда патчи, повышая безопасность системы, действительно могут оказывать негативное влияние на другие механизмы в системе. Поэтому патчи должны быть тщательно протестированы для выявления таких недостатков, прежде чем они будут установлены на серверах и рабочих станциях, находящихся в промышленной эксплуатации. Это позволит избежать нарушения работы систем и не оказать негативного влияния на продуктивность работы сотрудников и компании в целом.

8. Сбои и ошибки в приложениях

Многие обстоятельства непредсказуемы, поэтому очень трудно спланировать действия для реакции на них. Однако действия на случай подобных непредсказуемых ситуаций могут быть запланированы в общем виде, а не детально, для каждой ситуации в отдельности. Если в приложении возникает сбой по какой-либо причине, оно должно вернуться обратно в безопасное состояние. Для этого, например, может потребоваться перезагрузка операционной системы, после которой пользователь снова регистрируется в системе. Именно поэтому некоторые операционные системы в таких случаях отображают «синий экран» и / или автоматически перезагружаются. Когда в такой системе происходит что-то, что может перевести ее в неустойчивое или небезопасное состояние, система делает дамп содержимого оперативной памяти и выполняет полную перезагрузку.

Различные состояния системы были рассмотрены в Домене 03, в котором описывалась работа приложений в защищенном и реальном режимах. Если запущенное в защищенном режиме приложение завершается с ошибкой, его процессы должны корректно завершить работу и освободить ресурсы, чтобы не разрушить систему и не привести к нарушению безопасности, которое может быть использовано. Если привилегированный процесс после сбоя некорректно завершает свою работу, либо остается работать, злоумышленник может попытаться получить с помощью него доступ к системе от имени этого процесса, работающего в защищенном режиме. Это позволит ему получить административные права в системе и получить над ней полный контроль.

9. Управление базами данных

Базы данных давно используются для хранения информации, представляющей ценность для компаний, в том числе являющейся интеллектуальной собственностью. Базы данных обычно работают в среде, скрытой ото всех, кроме сетевых администраторов и администраторов баз данных. Чем меньше людей знают о базах данных, тем лучше. Пользователи обычно работают с базами данных не напрямую, а через клиентский интерфейс, их действия ограничены в целях обеспечения конфиденциальности, целостности и доступности данных, хранящихся в базе данных, а также структуры самой базы данных.

ПРИМЕЧАНИЕ. Система управления базами данных (СУБД) представляет собой набор программ, используемых для управления большими объемами структурированных данных и способных выполнять запросы от различных категорий пользователей. Также эти программы позволяют управлять настройками безопасности базы данных.

Риски возрастают, если компания подключает свою сеть к Интернет, разрешает удаленным пользователям доступ во внутреннюю сеть, предоставляет все больше прав доступа внешним субъектам. Эти действия несут большой риск, поскольку они могут стать косвенной причиной получения злоумышленником доступа к серверу базы данных, находящемуся во внутренней сети компании. Раньше информация клиентов компании хранилась в базах данных, к которым имели доступ только сотрудники компании, а не сами клиенты. В наше время многие компании позволяют клиентам получать доступ к хранящейся в базах данных информации, через веб-браузер. Веб-браузер устанавливает соединение с промежуточным программным обеспечением (middleware) компании, которое соединяется с сервером базы данных. Это повышает сложность системы, а доступ к базам данных организуется новым способом.

Одним из таких примеров являются системы интернет-банкинга. Многие банки хотят идти в ногу со временем и предоставлять новые услуги, которые, по их мнению, будут востребованы у клиентов. Но интернет-банкинг – это не просто еще одна банковская услуга. Внутренняя среда большинства банков является закрытой (или полузакрытой), и организация доступа в нее из сети Интернет является очень сложной задачей. Нужно предусмотреть надежную защиту периметра сети, разработать (или закупить) промежуточное программное обеспечение (шину), настроить доступ к базе данных через межсетевой экран (а лучше – через два межсетевых экрана). Доступ к данным при этом обычно организуется с помощью компонентов шины, которые по запросам клиентов обращаются к базе данных для извлечения / записи в нее нужных данных.

Такой доступ к базе данных может быть ограничен администратором с помощью средств контроля доступа и предоставляться только нескольким разрешенным ролям. При этом каждой роли будут даны определенные права и разрешения, а затем эти роли будут назначены клиентам и сотрудникам. Пользователь, которому не назначена ни одна из таких ролей, не имеет доступа к базе данных. Таким образом, если злоумышленник сможет преодолеть защиту межсетевого экрана и других механизмов защиты периметра сети, и получит возможность выполнять запросы к базе данных, то при условии, что у него не будет учетной записи, которой назначена одна из таких ролей, база данных все еще будет находиться в безопасности. Этот процесс упрощает управление доступом и гарантирует, что ни один пользователь (в т.ч. злоумышленник) не сможет получить доступ к базе данных напрямую, а только с помощью учетной записи, которой назначена соответствующая роль. Рисунок 9-2 иллюстрирует эту концепцию.

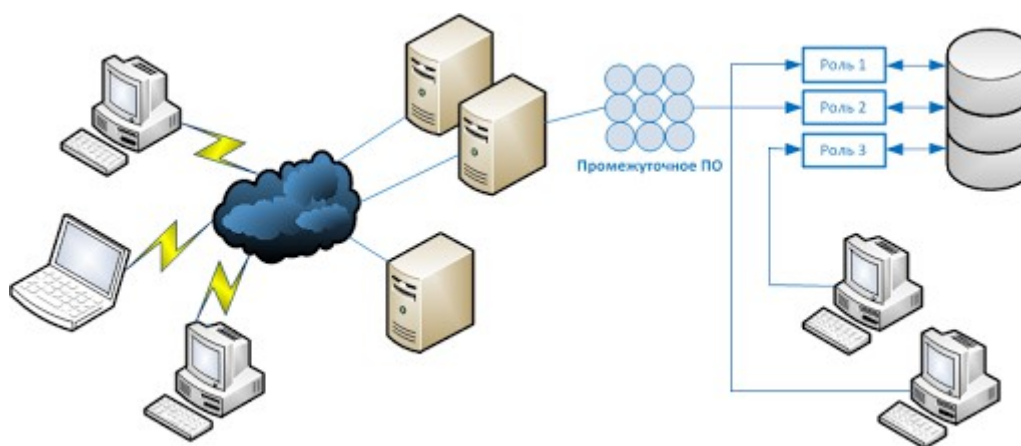


Рисунок 9-2. Один из вариантов обеспечения безопасности базы данных использует роли

9.1. Программное обеспечение для управления базами данных

База данных – это набор данных, хранящихся организованным способом, позволяющим множеству пользователей и приложений обращаться к данным, просматривать и изменять их по мере необходимости. Управление базами данных осуществляется специальным программным обеспечением, которое реализует соответствующую функциональность. Это программное обеспечение также реализует возможности для управления доступом, установки ограничений, обеспечивает целостность и избыточность данных, позволяет использовать различные процедуры для управления данными. Это программное обеспечение называется **системой управления базами данных (СУБД)**, как правило, его администрированием занимается администратор баз данных. Базы данных не только хранят данные, они также могут обрабатывать их и представлять в более удобном и логичном виде. СУБД реализует интерфейс для взаимодействия приложений и пользователей с данными, хранящимися в базе данных. СУБД помогает эффективно и рационально организовывать хранение, извлекать и записывать информацию в базу данных.

База данных предоставляет структуру для хранения собранных данных. Сама эта структура может отличаться для каждой реальной базы данных, поскольку различные компании и приложения работают с различными данными, типами данных, им необходимо выполнять различные действия с информацией. Различные приложения используют различные способы обработки данных, в различных базах данных устанавливаются различные отношения между данными, базы данных могут работать на различных платформах, к ним могут предъявляться различные эксплуатационные требования, а также требования по обеспечению безопасности. Однако любая база данных должна иметь следующие характеристики:

- Обеспечивать централизацию, позволяя не организовывать хранение данные на нескольких различных серверах по всей сети
- Позволять упростить процедуры резервного копирования
- Обеспечивать транзакционную устойчивость (transaction persistence)
- Позволять организовать работу более упорядоченно, поскольку все данные хранятся и сопровождаются в одном централизованном месте
- Обеспечивать отказоустойчивость и возможности для восстановления
- Позволять множеству пользователей совместно использовать данные
- Предоставлять механизмы безопасности, которые осуществляют контроль целостности, управление доступом, обеспечивают необходимый уровень конфиденциальности

ПРИМЕЧАНИЕ. Транзакционная устойчивость (transaction persistence) означает, что

реализованные в базе данных процедуры, выполняющие транзакции, являются надежными и проверенными. При использовании этих процедур, уровень безопасности базы данных не должен изменяться после выполнения транзакции, должна обеспечиваться целостность транзакций.

Поскольку потребности и требования к базам данных у различных компаний существенно различаются, могут использоваться различные модели данных, позволяющие увязать структуру данных с потребностями компаний и их бизнес-процессов.

9.2. Модели баз данных

Модель базы данных определяет отношения между различными элементами данных, указывает, каким образом может осуществляться доступ к данным, определяет допустимые операции, предлагаемый тип целостности, а также определяет, каким образом будут организованы данные. Модель дает формальный способ представления данных в концептуальной форме и предоставляет необходимые средства для работы с данными, хранящимися в базе данных. Базы данных могут быть реализованы на основе следующих моделей:

- Реляционная
- Иерархическая
- Сетевая
- Объектно-ориентированная
- Объектно-реляционная

Реляционная модель базы данных (relational database model) для хранения и организации информации использует атрибуты (столбцы) и записи (строки) (см. Рисунок 9-3).

Реляционная модель базы данных в настоящее время является наиболее широко используемой моделью. Реляционная база данных состоит из двумерных таблиц, каждая таблица содержит уникальные строки, столбцы и ячейки (пересечение строки и столбца). Каждая ячейка содержит только одно значение данных, представляющее собой конкретное значение атрибута соответствующей записи. Элементы данных связаны отношениями. Отношения между элементами данных предоставляют основу для организации данных. Первичный ключ (primary key) – это поле, которое содержит уникальное значение, не повторяющееся в других записях, и позволяющее связать все данные в рамках одной записи в одно уникальное значение. Например, в таблице на Рисунке 9-3, первичными ключами являются продукты G345 и G978. Когда приложение или другая запись ссылается на этот первичный ключ, в действительности она ссылается на все данные в рамках этой строки.

Продукт	Размер	Цвет	Вес
G345	Средний	Зеленый	2
G978	Средний	Синий	6

Рисунок 9-3. Реляционная база данных хранит данные в двумерных таблицах

Иерархическая модель базы данных (hierarchical data model) (см. Рисунок 9-4) объединяет связанные записи и поля в логическую древовидную структуру. Эта структура и взаимосвязи между элементами данных, отличаются от тех, которые используются в реляционной базе данных. В иерархической базе данных родительские элементы могут иметь дочерние элементы (один, несколько или ни одного). Древовидная структура имеет ветви, каждая ветвь имеет множество листьев – полей данных. В таких базах данных есть хорошо известные, заранее определенные пути доступа к данным, но они не настолько гибки при создании отношений между элементами данных, по сравнению с реляционными базами данных. Иерархические базы данных целесообразно использовать для хранения данных, имеющих отношения «один-ко-многим».

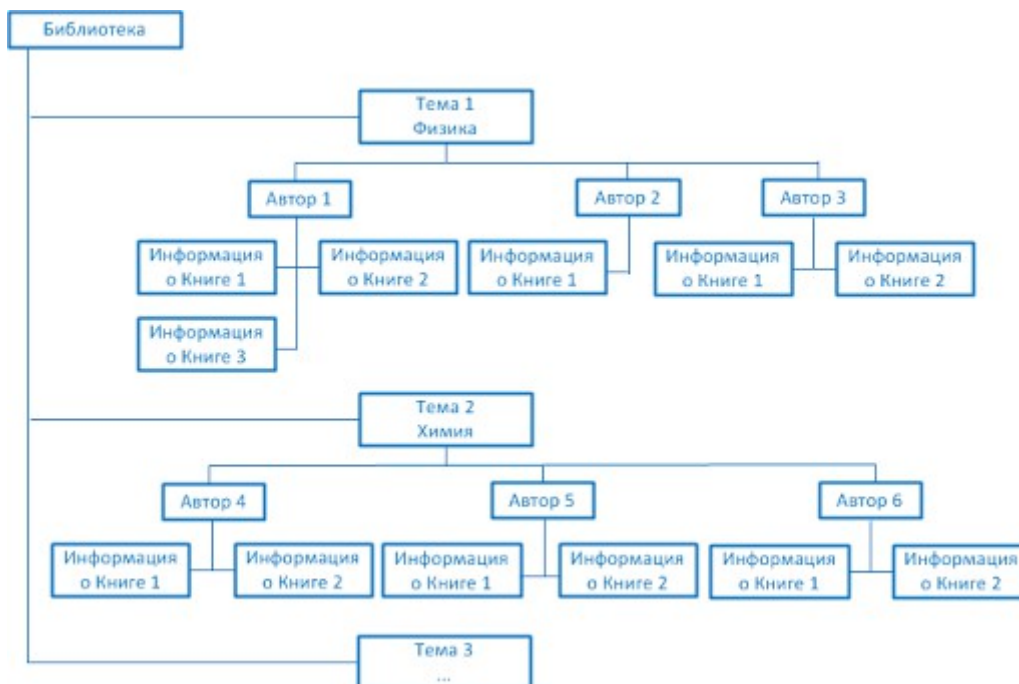


Рисунок 9-4. Иерархическая база данных хранит данные в структуре, имеющей форму деревьев, отношения данных в ней реализованы в виде отношений родительских и дочерних элементов

Иерархическая структура базы данных была одной из первых разработанных моделей, но она не получила такого распространения, как реляционные базы данных. Чтобы получить доступ к элементу данных в иерархической базе данных, необходимо знать с какой ветви начинать и по какому маршруту проходить через каждый уровень, пока не будет достигнут уровень, на котором хранятся нужные данные. В таких базах данных процедуры поиска не используют индексы, в отличие от реляционных баз данных. Кроме того, ссылки (отношения) не могут быть созданы между различными ветвями и листьями на разных уровнях.

Наиболее часто используемой реализацией иерархической модели является модель LDAP. Также, иерархическая модель используется в структуре системного реестра Windows и различных файловых системах, но в новых реализациях баз данных она обычно не используется.

Сетевая модель базы данных (network database model) построена на основе иерархической модели данных. Чтобы обойти ограничения иерархической модели, требующие для получения элемента данных знать маршрут перехода с одной ветви в другую, а затем от родительского элемента к дочернему, в сетевой модели каждому элементу данных разрешается иметь несколько родительских и дочерних записей. Это создает избыточную, похожую на сеть структуру, а не жесткую древовидную структуру. Посмотрите на Рисунок 9-5, вы увидите, что сетевая модель создает структуру похожую на полносвязную топологию сети. Это обеспечивает избыточность и дает возможность более быстрого поиска данных по сравнению с иерархической моделью.

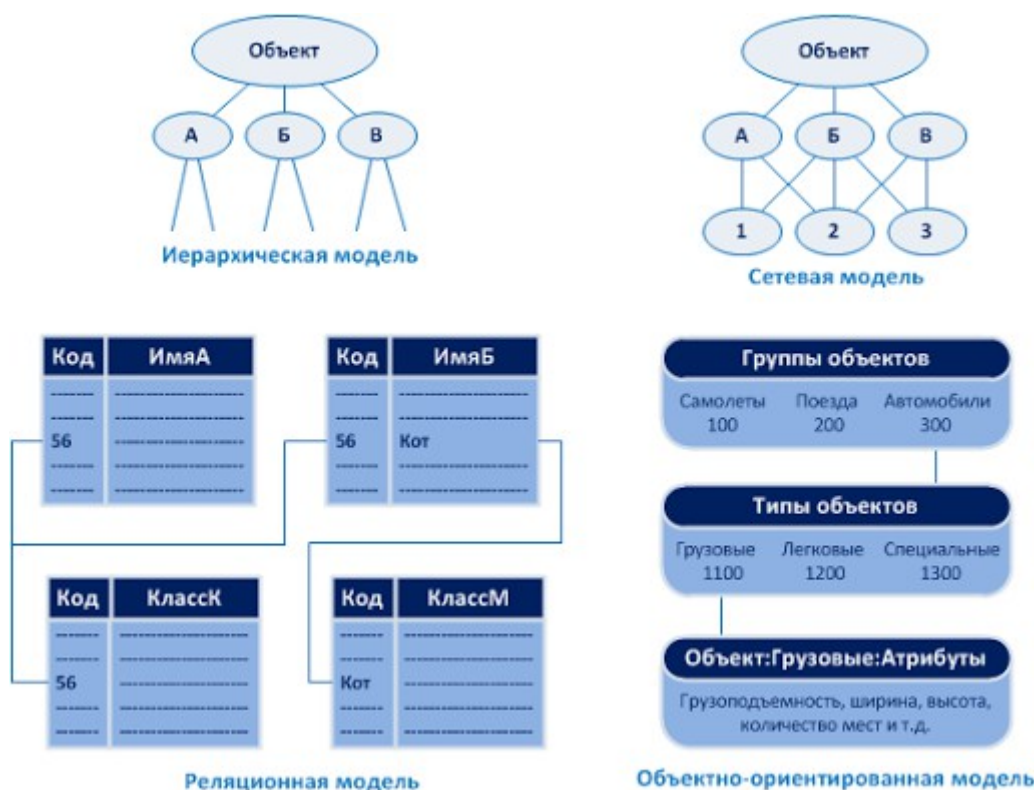


Рисунок 9-5. Различные модели баз данных

Эта модель использует конструкции из записей и множеств. Запись содержит поля, которые могут располагаться в иерархической структуре. Множества определяют отношения «один-ко-многим» между различными записями. Одна запись может быть «владельцем» любого количества множеств, при этом тот же «владелец» сам может быть членом различных множеств. Это означает, что одна запись может быть «главной» и под ней может находиться множество элементов данных, либо эта запись может находиться ниже в иерархии, под различными полями, являющимися для нее «главными». Это предоставляет значительную гибкость при разработке отношений между элементами данных.

Объектно-ориентированная база данных (object-oriented database) предназначена для работы с различными типами данными (изображения, аудио, документы, видео). Система управления объектно-ориентированными базами данных (ODBMS - object-oriented database management system) более динамична по своей природе, чем реляционная СУБД, поскольку она создает объекты при необходимости, а данные и процедуры (называемые методами) при запросе объекта предоставляются вместе с ним. При работе с реляционной базой данных, приложение должно использовать свои собственные процедуры для получения данных из базы данных и их обработки. Реляционная база данных не предоставляет процедур, как это делает объектно-ориентированная база данных. Объектно-ориентированная база данных использует классы для определения атрибутов и процедур ее объектов.

В качестве аналогии, рассмотрим две компании, в клиентских базах данных которых находятся одинаковые данные. Если вы придете в компанию А (реляционная база данных), менеджер сможет дать вам только лист бумаги, на котором будет указана информация. Вы сами должны понять, что делать с этой информацией и как правильно использовать ее для своих нужд. Если вы придете в компанию В (объектно-ориентированная база данных), менеджер даст вам коробку. В этой коробке будет листок с той же информацией, но кроме него там будет набор инструментов, позволяющих обработать информацию для удовлетворения ваших потребностей, и вам не нужно будет делать это самостоятельно. Таким образом, когда ваше приложение запрашивает данные в объектно-ориентированной базе данных, в ответ оно получает не только данные, но и код для выполнения определенных

процедур над этими данными. Мы рассмотрим объектно-ориентированное программирование далее, тогда вы лучше поймете объекты, классы и методы.

Целью создания этой модели была попытка учесть ограничения, которые накладывало использование реляционной базы данных при необходимости хранения и обработки больших объемов данных. Кроме того, объектно-ориентированные базы данных не зависят от SQL, с такими базами данных могут работать приложения, не являющиеся SQL-клиентами.

Жаргон баз данных. Ниже приведены некоторые ключевые понятия, используемые при работе с базами данных:

- **Запись (Record)** – набор связанных элементов данных
- **Файл (File)** – набор однотипных записей
- **База данных (Database)** – набор данных, связанных с перекрестными ссылками (cross-referenced)
- **СУБД (DBMS)** – система управления и работы с базой данных
- **Запись (Tuple)** – строка в двумерной базе данных
- **Атрибут (Attribute)** – столбец в двумерной базе данных
- **Первичный ключ (Primary key)** – столбец, который делает каждую строку уникальной (каждая строка таблицы должна содержать первичный ключ)
- **Представление (View)** – виртуальное представление информации, определенное администратором для ограничения просмотра субъектами определенных данных
- **Внешний ключ (Foreign key)** – атрибут одной таблицы, связанный с первичным ключом другой таблицы
- **Ячейка (Cell)** – пересечение строки и столбца
- **Схема (Schema)** – определяет структуру базы данных
- **Словарь данных (Data dictionary)** – центральное хранилище (репозиторий) элементов данных и их взаимосвязей

ПРИМЕЧАНИЕ. Язык структурированных запросов (Structured Query Language, SQL) представляет собой стандартный язык программирования, используемый для организации взаимодействия клиентов с базой данных. Большинство реализаций баз данных поддерживают SQL. SQL позволяет клиентам выполнять такие операции, как вставка, замена, поиск и добавление данных.

Объектно-ориентированные базы данных не так распространены, как реляционные базы данных, они используются в основном в таких областях, как машиностроение, биология, а также для удовлетворения некоторых потребностей финансового сектора.

Теперь давайте рассмотрим объектно-реляционные базы данных. **Объектно-реляционная база данных** (object-relational database, ORD) или объектно-реляционная система управления базами данных (object-relational database management system, ORDBMS) – это реляционная база данных с фронтальным программным обеспечением (интерфейсом), написанным на объектно-ориентированном языке программирования. Но зачем нужны такие комбинации? Реляционная база данных содержит данные в статических двумерных таблицах. При обращении к данным, они должны подвергаться какой-либо последующей обработке, иначе зачем получать эти данные? Если у нас есть интерфейс, предоставляющий процедуры (методы) обработки данных, тогда приложению, которое обращается к этой базе данных, не нужны аналогичные собственные процедуры.

Различным компаниям требуется различная бизнес-логика для работы с данными. Возможность разработать такого фронтального программного обеспечения позволяет приложениям использовать процедуры бизнес-логики и данные базы данных. Например, если у нас есть реляционная база данных, в которой хранятся данные инвентаризации

товаров на складе, нам хотелось бы иметь возможность использовать эти данные для различных бизнес-целей. Одно приложение может обращаться к базе данных, чтобы просто проверить количество имеющихся в наличии единиц товара А. Можно создать интерфейсный объект, который будет выполнять эту процедуру, обращаясь за данными в базу данных и предоставляя готовый ответ запрашивающему приложению. Также может существовать потребность проведения аналитических расчетов на основании данных инвентаризации, например, провести анализ наиболее востребованных товаров. Для этого может быть разработан другой объект, который будет собирать из базы данных нужные данные, проводить расчеты и предоставлять их результаты запрашивающему приложению. Для выполнения других расчетов и подготовки отчетов могут быть созданы другие объекты. На Рисунке 9-6 показаны различные объекты данных, выполняющие различные команды бизнес-логики.



Рисунок 9-6. Объектно-реляционная модель позволяет включать в объекты бизнес-логику и функции

9.3. Интерфейсы программирования баз данных

Данные бесполезны, если вы не можете их использовать. Приложения должны иметь возможность получать и работать с информацией, хранимой в базах данных. Они также нуждаются в некотором интерфейсе и механизме передачи информации. Ниже мы рассмотрим некоторые из таких интерфейсов:

- **Open Database Connectivity (ODBC)** – это интерфейс прикладного программирования (API – application programming interface), который позволяет приложению взаимодействовать с базой данных локально или удаленно. Приложение посылает запросы к ODBC API, ODBC определяет необходимый для конкретной базы данных драйвер, позволяющий выполнить трансляцию запросов, затем этот драйвер выполняет указанную трансляцию запросов в команды базы данных, понятные для этой базы данных.
- **Object Linking and Embedding Database (OLE DB)** разделяет данные на компоненты, которые работают как промежуточное программное обеспечение (middleware) на клиенте или сервере. Это предоставляет низкоуровневый интерфейс для связи информации, хранящейся в различных базах данных, и обеспечивает доступ к данным независимо от того, где они хранятся и в каком формате. Ниже приведены некоторые характеристики OLE DB:
 - Он заменяет ODBC, расширяет набор функций для поддержки более широкого круга нереляционных баз данных, таких как объекты баз данных и таблицы, которые не обязательно поддерживают SQL
 - Набор основанных на COM интерфейсов, которые предоставляют приложениям унифицированный доступ к данным, хранящимся в различных источниках данных (см. Рисунок 9-7)

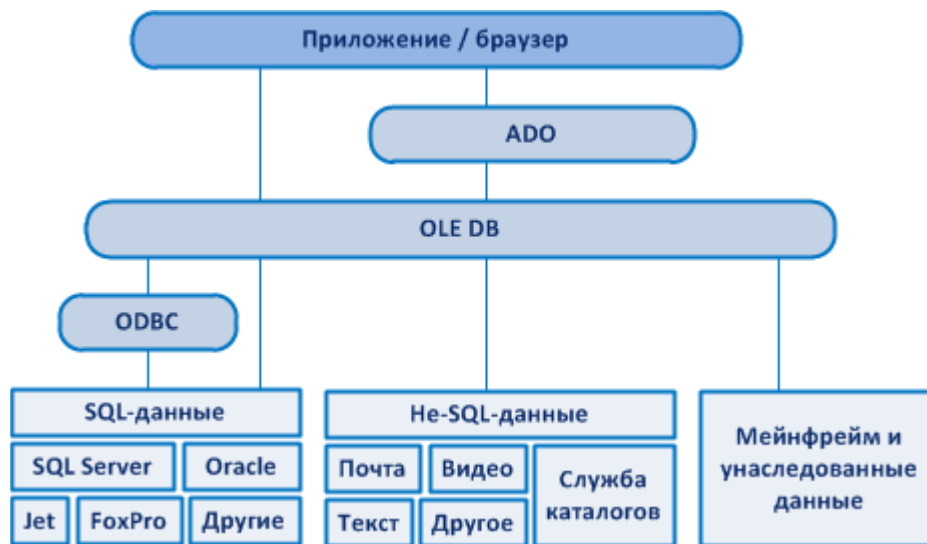


Рисунок 9-7. OLE DB предоставляет интерфейс, позволяющий приложениям взаимодействовать с различными источниками данных

- Поскольку OLE DB основан на COM, он ограничен использованием клиентских средств, разработанных для платформы Microsoft Windows
- Разработчик обращается к сервисам OLE DB через объекты данных ActiveX (ADO – ActiveX Data Objects)
- Это позволяет различным приложениям использовать различные типы и источники данных
- **ActiveX Data Objects (ADO)** – это API, позволяющий приложениям получать доступ к серверам баз данных. Он представляет собой набор интерфейсов ODBC, которые позволяют использовать функциональность источников данных посредством доступных объектов. ADO для соединения с базой данных использует интерфейс OLE DB, он может быть использован в процессе разработки на множестве различных языков сценариев. Ниже приведены некоторые характеристики ADO:
 - Это высокоуровневый программный интерфейс доступа к лежащей ниже технологии, реализующей доступ к данным (такой как OLE DB)
 - Это набор COM-объектов для доступа к источникам данных, а не просто доступа к базе данных
 - Он позволяет разработчикам писать программы доступа к данным, не зная, каким образом реализована сама база данных
 - Команды SQL не требуются для доступа к базе данных при использовании ADO
- **Java Database Connectivity (JDBC)** – это API, который позволяет Java-приложениям взаимодействовать с базами данных. Приложение может обращаться к базе данных через ODBC или напрямую. Ниже приведены некоторые характеристики JDBC:
 - Это API, который обеспечивает функциональность, аналогичную ODBC, но он специально разработан для использования приложениями баз данных на Java
 - Он позволяет использовать независимые от базы данных соединения между платформой Java и широким кругом баз данных
 - JDBC представляет собой Java API, который позволяет Java-программам выполнять SQL-выражения

9.4. Компоненты реляционной базы данных

Как и любое программное обеспечение, базы данных разрабатываются с помощью языков программирования. Большинство языков программирования баз данных включает *язык описания данных* (DDL – data definition language), который определяет схему; *язык манипулирования данными* (DML – data manipulation language), который анализирует данные и определяет, как эти данные могут обрабатываться в базе данных; *язык управления данными* (DCL – data control language), который определяет внутреннюю организацию базы данных; специальный *язык запросов* (QL – query language), который определяет запросы, позволяющие пользователям получить доступ к данным в базе данных.

Каждая модель базы данных может иметь множество других отличий, обусловленных, в том числе, разными подходами различных производителей. Однако большинство из них включает следующую базовую функциональность:

- **Язык описания данных (DDL)** определяет структуру и схему базы данных. Структуру может определять размер таблицы, размещение ключа, представления, отношения элементов данных. Схема описывает тип данных, которые будут храниться и обрабатываться, а также их свойства. DDL определяет структуру базы данных, операции доступа и процедуры целостности.
- **Язык манипулирования данными (DML)** содержит все команды, позволяющие пользователю просматривать, управлять и использовать базу данных (команды view, add, modify, sort, delete).
- **Язык запросов (QL)** дает пользователям возможность делать запросы в базу данных.
- **Генератор отчетов** готовит печатные формы с данными определенным пользователем образом.

Словарь данных

Словарь данных (data dictionary) является централизованным набором определений элементов данных, объектов схемы, а также ключей ссылок (reference keys). Объекты схемы могут содержать таблицы, представления, индексы, процедуры, функции и триггеры. Словарь данных может содержать значения по умолчанию для столбцов, информацию целостности, имена пользователей, привилегии и роли пользователей, информацию аудита. Это инструмент, используемый для централизованного управления частями базы данных посредством управления данными о данных (именуемыми *метаданными*) в базе данных. Он обеспечивает перекрестные ссылки между группами элементов данных и базами данных.

Программное обеспечение, управляющее базой данных, создает и читает словарь данных, чтобы выяснить, какие существуют объекты схемы, а также проверить, имеют ли конкретные пользователи права доступа, необходимые для их просмотра (см. Рисунок 9-8). При просмотре пользователями базы данных, они могут быть ограничены определенными представлениями. Различные параметры представлений для каждого пользователя хранятся в словаре данных. При добавлении новых таблиц, новых строк, или новой схемы, словарь данных обновляется – в него вносятся соответствующие изменения.



Рисунок 9-8. Словарь данных – это централизованное хранилище, которое содержит информацию о базе данных

Первичные и внешние ключи

Первичный ключ (primary key) – это идентификатор строки, он используется для индексации в реляционных базах данных. Каждая строка должна иметь уникальный первичный ключ, который должен представлять строку, как единое целое. Когда пользователь делает запрос на просмотр записи, база данных находит нужную запись по ее уникальному первичному ключу. Если первичный ключ не был бы уникальным, база данных не знала бы, какие записи нужно предоставить пользователю. На приведенном ниже рисунке, первичным ключом для таблицы А является кличка собаки. Каждая строка (запись) содержит характеристики каждой собаки. Поэтому, когда пользователь осуществляет поиск собаки «Бобик», ему будут предоставлена информация о породе, весе, цвете и хозяине соответствующей собаки.

Таблица А				
Первичные ключи	Собака	Порода	Вес	Владелец
	Шарик	Пудель	10 кг	Иванов
	Тузик	Такса	4 кг	Петров
	Бобик	Лайка	12 кг	Сидоров
	Дружок	Овчарка	22 кг	Александров
				Цвет
				Белый
				Черный
				Коричневый
				Черный

Первичный ключ отличается от внешнего ключа, хотя они тесно связаны между собой. Если атрибут в одной таблице, имеет значение, соответствующее первичному ключу в другой таблице, и между этими двумя таблицами установлены отношения, этот атрибут называется **внешним ключом** (foreign key). Этот внешний ключ не обязательно является первичным ключом в своей таблице. Просто он должен содержать ту же информацию, которая содержится в первичном ключе другой таблицы, и быть связанным с первичным ключом в этой другой таблице. На приведенном ниже рисунке, первичным ключом Таблицы А является «Шарик». Поскольку Таблица В имеет атрибут, содержащий те же данные, что и этот первичный ключ, и между этими двумя ключами установлена связь, он называется внешним ключом. Это еще один способ для отслеживания взаимосвязей между данными, хранящимися в базе данных.



Можно представить это в виде веб-страницы, которая содержит данные из Таблицы В. Если нам нужно больше узнать о собаке по кличке Шарик, мы дважды щелкаем по этому значению и браузер выдает характеристики этой собаки, которые хранятся в Таблице А.

Это позволяет создавать взаимосвязи между различными элементами данных в базе данных по своему усмотрению.

9.5. Целостность

Как и другие сетевые ресурсы, базы данных могут столкнуться с проблемами *конкуренции*. Проблемы конкуренции возникают, когда определенные ресурсы или данные должны быть доступны одновременно нескольким пользователям и/или приложениям. Рассмотрим следующий пример. Две группы пользователей используют один и тот же файл, содержащий таблицу товаров с ценами, чтобы знать, какой объем поставок нужен на следующей неделе, а также рассчитать ожидаемую прибыль. Если Дэн и Элизабет копируют этот файл с файлового сервера, на свои рабочие станции, у каждого из них есть копия оригинального файла. Предположим, что Дэн изменил объем складского запаса книг о компьютерах от 120 до 5, поскольку их компания продала 115 книг в течение последних трех дней. Затем он на основе текущих цен, указанных в файле, рассчитывает ожидаемую прибыль на следующую неделю. Элизабет снижает цены на ряд программных пакетов в своей копии файла и видит, что объем складских запасов книг о компьютерах еще более 100 единиц, поэтому она решает не заказывать их на следующую неделю. Дэн и Элизабет не сообщают эту информацию друг другу, а просто загружают свои копии исправленного файла на сервер для общего просмотра и использования.

Сначала Дэн копирует свои изменения на файл-сервер, а затем, через 30 секунд после Дэна, Элизабет копирует свои изменения. Как вы понимаете, теперь на файловом сервере хранится файл, в котором указаны только изменения, произведенные Элизабет, т.к. она записала свой файл поверх файла Дэна. Они не синхронизировали свои изменения и оба воспользовались неверными данными. Расчеты прибыли Дэна неверны, т.к. он не знал, что Элизабет снизила цены на следующую неделю, а у Элизабет не будет компьютерных книг, потому что она не знала, что их остаток упал до пяти единиц.

То же самое происходит и в базах данных. Если в ней не реализован соответствующий контроль, два пользователя могут одновременно использовать и изменять одни и те же данные, что может иметь пагубные последствия для динамичной среды. Чтобы исключить проблемы конкуренции, процессы могут блокировать таблицы в базе данных, вносить изменения, а затем снимать программную блокировку. При этом когда следующий процесс будет обращаться к таблице, он получит обновленную информацию. Применение блокировки гарантирует, что два процесса не получают одновременный доступ к одной и той

же таблице, а обновления будут выполняться по одному за раз. Блокировка может быть выполнена для отдельных страниц, таблиц, строк и полей, что обеспечит возможность предоставления каждому процессу и пользователю правильной и точной информации.

Программное обеспечение базы данных реализует три основных типа механизмов обеспечения целостности: семантический, ссылочный и логический. Механизм **семантической целостности** (semantic integrity) обеспечивает реализацию структурных и семантических правил. Эти правила относятся к типам данных, логическим значениям, требованиям уникальности, а также операциям, которые могут оказать негативное воздействие на структуру базы данных. Данные в базе данных должны изменяться таким образом, чтобы не нарушалась установленная между ними смысловая (семантическая) связь. В базе данных обеспечивается **ссылочная целостность** (referential integrity), если все внешние ключи ссылаются на существующие первичные ключи. Должен быть реализован механизм, который обеспечивает отсутствие внешних ключей, содержащих ссылку на первичный ключ несуществующей записи или на пустое значение. **Логическая целостность** (entity integrity) гарантирует, что записи уникально идентифицируются по значениям первичного ключа. В рассмотренном ранее примере первичными ключами являются клички собак. Для обеспечения логической целостности, в базе данных не должно существовать двух собак с одинаковыми кличками. Каждая запись должна содержать один первичный ключ, т.к. если запись не имеет первичного ключа, на нее не может ссылаться база данных.

База данных не должна содержать несогласованных значений внешних ключей, т.е. не должна иметь внешних ключей, ссылающихся на несуществующие первичные ключи. Возвращаясь к тому же примеру с собаками, если внешний ключ в Таблице В – «Шарик», то Таблица А должна содержать запись для собаки по кличке «Шарик». Если эти значения не совпадают, то нарушается их связь и база данных не может правильно ссылаться на информацию.

Существуют и другие настраиваемые операции, обеспечивающие защиту целостности данных в базе данных. Такими операциями являются функции отката, фиксации, точек сохранения, а также контрольных точек.

Откат (rollback) – это операция, которая прерывает текущую транзакцию и отменяет все произведенные в рамках этой транзакции изменения в базе данных. Эти изменения могут касаться самих данных или схемы. При выполнении операции отката, отменяются все изменения и база данных возвращается в свое предыдущее состояние (точку сохранения). Откат может потребоваться в случае, если в базе данных произошел неожиданный сбой или внешний субъект нарушил последовательность обработки. Вместо того чтобы передавать и сохранять частичную или поврежденную информацию, база данных просто возвращается в исходное состояние, а в журнал регистрации событий записывается сообщение об ошибке и выполненных действиях, чтобы они могли быть проанализированы позднее.

Операция **фиксации** (commit) завершает транзакцию и применяет все изменения, сделанные пользователем, т.е. эта операция реально записывает все изменения в базу данных. Эти изменения могут относиться к данным или к схеме. После фиксации этих изменений, обновленные данные будут доступны для всех других приложений и пользователей. Если пользователь попытается выполнить операцию фиксации изменений, но эта операция не сможет завершиться правильно, выполняется откат. Это гарантирует отсутствие в базе данных частичных изменений и поврежденных данных.

Точки сохранения (savepoints) используются для того, чтобы обеспечить восстановление целостности базы данных в случае сбоев и ошибок. При возникновении сбоя или ошибки, база данных пытается вернуться к точке (состоянию), в которой она находилась непосредственно перед возникновением этого сбоя или ошибки. Чтобы понять основной принцип, рассмотрим следующий пример. Дэйв ввел текст «В 2010 году компанией получена

прибыль в размере 1 млн.руб. Планы <точка сохранения> по прибыли выполнены на 115%». Сразу после этого произошел сбой электропитания, который привел к перезагрузке системы. Когда Дэйв снова запустил клиентское приложение базы данных, он увидел следующий текст: «В 2010 году компанией получена прибыль в размере 1 млн.руб. Планы », но дальше текст был потерян. Таким образом, точка сохранения обеспечила сохранность некоторой части его работы. Базы данных и другие приложения используют эту технологию, чтобы попытаться восстановить работу пользователей и состояние базы данных после сбоя, однако иногда происходят крупные сбои, которые невозможно исправить с помощью этой технологии.

Реализовать точки сохранения в базе данных или другом приложении легко, однако необходимо обеспечить баланс между слишком большим и слишком малым количеством точек сохранения. Использование слишком большого их количества может ухудшить производительность, тогда как недостаточное количество повышает риск потери данных и тем самым снижает продуктивность работы пользователей, т.к. потерянные данные им придется вводить заново. Точки сохранения могут создаваться через определенные промежутки времени, определенными действиями пользователя, либо при достижении определенного числа транзакций или изменений, внесенных в базу данных. Например, база данных может быть настроена на создание точки сохранения каждые 15 минут, после ввода каждых 20 операций, а также каждый раз, когда пользователь доходит до последней записи.

Точки сохранения позволяют восстановить данные, давая пользователю возможность вернуться назад во времени до момента, когда система вышла из строя или произошла ошибка. Это может уменьшить количество проблем и помогает нам работать более эффективно.

ПРИМЕЧАНИЕ. Контрольные точки (checkpoint) очень похожи на точки сохранения. Создание контрольной точки инициируется, когда программное обеспечение базы данных заполняет определенный объем памяти. При этом все данные из сегмента памяти записываются во временный файл. При возникновении сбоя, программное обеспечение пытается использовать эту информацию, чтобы восстановить рабочую среду пользователя в состоянии, предшествующем сбою.

Механизм **двухэтапной фиксации** (two-phase commit) – это еще одна защитная мера, которая применяется для обеспечения целостности данных в базе данных. Базы данных обычно работают в транзакционном режиме, что означает взаимодействие пользователя и базы данных в режиме реального времени. Противоположностью является режим пакетной обработки, при котором запросы на изменение базы данных ставятся в очередь и активируются все сразу, но не в тот же момент времени, когда пользователь делает каждый запрос. При выполнении транзакций, часто возникает потребность в обновлении более чем одной базы данных в рамках транзакции. Программное обеспечение должно убедиться, что в каждой базе данных выполнены необходимые изменения, либо не произошло никаких изменений ни в одной из баз данных. После подтверждения пользователем необходимости изменения базы данных, базы данных сначала выполняют временное сохранение этих изменений. Затем монитор транзакций отправляет команду «предфиксации» (pre-commit) каждой базе данных. Если все базы данных ответили подтверждением, монитор посылает каждой базе данных команду «фиксации» (commit). Это обеспечивает корректное и своевременное сохранение всех необходимых данных.

Ссылки по теме:

- What is a database?
- Database
- Databases 1 & 2

9.6. Вопросы безопасности баз данных

Двумя основными вопросами безопасности баз данных, которые рассмотрены в этом разделе, являются агрегирование и предположения. **Агрегирование** (aggregation) может выполняться, когда пользователь не имеет допуска или разрешения на доступ к определенной информации, но у него есть разрешение на доступ к частям этой информации. Ознакомившись со всеми частями, к которым у него есть доступ, он может догадаться об остальной информации и получить, таким образом, сведения ограниченного доступа. Также он может получить информацию из различных источников и объединить ее, чтобы узнать что-то, к чему у него нет допуска.

ПРИМЕЧАНИЕ. Агрегирование – это действие по объединению информации из различных источников. Полученное в результате сочетание данных формирует новую информацию, к которой у субъекта нет прав доступа. Совокупная информация имеет большую критичность, чем отдельные ее части.

Для предотвращения агрегирования, нужно предотвратить возможность доступа субъекта (и любого приложения или процесса, действующего от имени субъекта) ко всему набору, состоящему из независимых компонентов. Для этого объекты могут быть помещены в контейнеры, которым присвоены более высокие уровни классификации, что позволит исключить доступ к ним субъектов с менее высоким уровнем допуска. Также можно отслеживать запросы субъекта и применять системы контекстно-зависимого управления доступом. Для этого нужно сохранять историю доступа субъекта и на основании нее динамически ограничивать попытки доступа, если обнаружены признаки проведения атаки агрегирования.

Другая проблема безопасности заключается в **предположениях** (inference), являющихся результатом агрегирования. Субъект догадывается (предполагает, делает выводы) о полной истории на основе отдельных ее частей, которые он узнал в процессе агрегирования. Это может проявиться, когда данные на более низком уровне безопасности, косвенно отражают данные более высокого уровня безопасности.

ПРИМЕЧАНИЕ. Предположения позволяют получить информацию, не доступную в явном виде.

Рассмотрим следующий пример. Доступ военнослужащего к сведениям о планах передвижения войск, базирующихся в определенной стране, был ограничен. Но он имел доступ к документам с требованиями по поставкам продовольствия и распределению палаток. На основании информации, к которой у него был доступ, он мог догадаться о перемещении войск в конкретное место, зная, что туда направлено продовольствие и палатки. В рассматриваемом примере, документы, касающиеся продовольствия и палаток были отнесены к категории «конфиденциально», а сведения о перемещении войск имели гриф «совершенно секретно». Таким образом, этот военнослужащий мог получить доступ к сверхсекретной информации, которую он не должен знать.

Предотвратить подобный косвенный доступ субъекта (либо любого приложения или процесса, действующего от имени субъекта) к информации, которая позволяет сделать подобные умозаключения, является крайне сложной задачей. Обычно эта проблема учитывается на этапе разработки базы данных с помощью реализации контентно- и контекстно-зависимых правил управления доступом. **Контентно-зависимое управление доступом** основано на критичности данных. Чем более критичны данные, тем меньше группа лиц, которые могут получить доступ к ним..

Система **контекстно-зависимого управления доступом** «понимает», какие действия должны быть разрешены, основываясь на состоянии и последовательности запросов. Для этого она отслеживает все предыдущие попытки доступа пользователя, и принимает решение, какая последовательность шагов доступа может быть разрешена. Чтобы лучше понять процесс принятия решения о доступе системой контекстно-зависимого управления доступом, а также различие между системами контекстно- и контентно-зависимого

управления доступом, предположим, что Хулио запрашивает доступ к файлу А. Если используется система контентно-зависимого управления доступом, она проверяет список контроля доступа этого файла, видит, что у Хулио есть доступ к этому файлу на чтение, и на основании этого предоставляет ему доступ. При использовании контекстно-зависимого управления доступом сначала также проверяется список контроля доступа файла А, но затем система анализирует дополнительную информацию: какие другие попытки доступа делал Хулио перед этим, удовлетворяет ли последовательность его запросов требованиям к безопасной последовательности запросов, попадает ли время запроса в разрешенный интервал (с 8 утра до 5 вечера) и т.п. Если ответы на все эти вопросы положительны, Хулио получает доступ к файлу.

Как вы видите, контентно-зависимое управление доступом существенно проще контекстно-зависимого, поскольку системой выполняется меньшее количество действий.

Обычно для предотвращения атак на основе предположений (inference attack) применяется скрывание ячеек, разделение базы данных, добавление шума и пертурбация. **Скрывание ячеек** (cell suppression) – это технология сокрытия конкретной ячейки, содержащей информацию, которая может быть использована для проведения атаки на основе предположений.

Разделение (partitioning) базы данных на отдельные части значительно затрудняет для неуполномоченных лиц поиск связанных элементов данных, при объединении которых может быть угадана и раскрыта новая информация. **Шум** (noise) и **пертурбация** (perturbation) – это способ вставки в базу данных фиктивной информации, чтобы ввести злоумышленника в заблуждение или запутать его, что не позволит провести успешную атаку на основе предположений.

Если для защиты от атак на основе предположений используется контекстно-зависимое управление доступом, программное обеспечение базы данных должно отслеживать, что пользователь запрашивает. К примеру, когда Хулио делает запрос на просмотр поля 1, затем поля 5, а затем поля 20 – система разрешает ему этот доступ. Но когда после этого он запрашивает поле 15, система блокирует эту попытку доступа. Программное обеспечение, реализующее контекстно-зависимое управление доступом, должно быть предварительно запрограммировано (обычно на основе настройки правил), в какой последовательности и какой объем данных разрешается просмотреть Хулио. Если ему будет разрешен просмотр дополнительной информации, он сможет получить достаточно данных, чтобы догадаться о том, что он не должен знать.

Зачастую в процесс планирования и разработки базы данных безопасность не интегрируется. О безопасности вспоминают в самом конце, и для реализации защиты данных разрабатывают доверенный фронтальный интерфейс, позволяющий использовать базу данных. Такой подход ограничивает степень детализации настроек безопасности и функционал защитных механизмов.

При реализации защитных мер часто возникает сложность обеспечения эффективного баланса между безопасностью и функциональностью. В большинстве случаев, чем лучше вы что-то защищаете, тем меньше функциональных возможностей у вас остается. Хотя это позволяет достичь желаемого уровня безопасности, важно не мешать эффективной работе пользователей.

Представления базы данных

База данных может позволить одной группе или конкретному пользователю видеть определенную информацию, тогда как другой группе доступ к этой информации будет полностью ограничен. Это реализуется с помощью **представлений базы данных** (database view), показанных на Рисунке 9-9. Если администратор базы данных хочет разрешить руководителям среднего звена видеть показатели по прибылям и убыткам в разрезе их подразделений, но не показывать им данные по прибыли и убыткам всей компании, он

может создать представление. Высшему руководству будет дан доступ ко всему представлению, содержащему информацию по прибыли и убыткам всех подразделений и компании в целом, тогда как каждый руководитель подразделения сможет видеть только информацию, относящуюся к его подразделению.

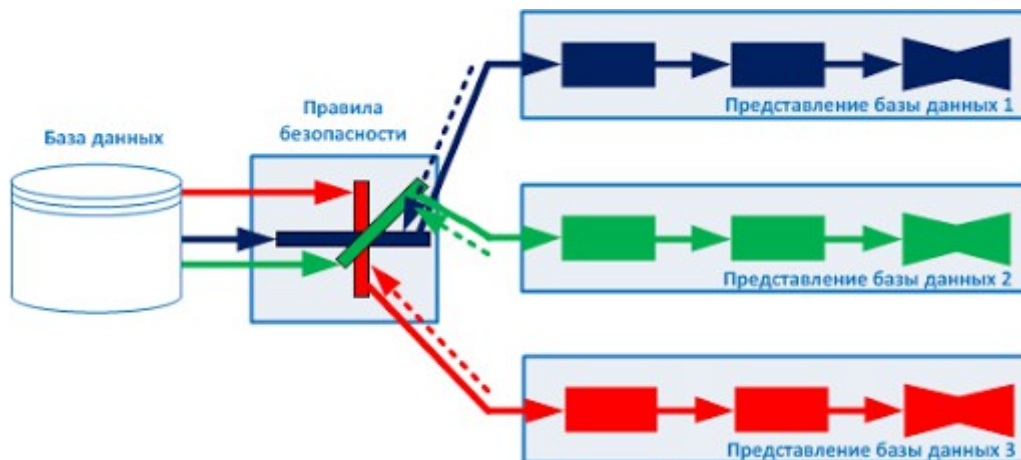


Рисунок 9-9. Представления баз данных являются логическим видом управления доступом

Подобно операционным системам, базы данных также могут использовать дискреционное (DAC) и мандатное (MAC) управление доступом, подробно рассмотренные в Домене 02. Доступ к представлениям может предоставляться на основе членства в группе, прав пользователя или меток безопасности. Если используется дискреционное управление доступом, группы и пользователи получают доступ к представлениям по результатам их идентификации, аутентификации и авторизации. Если внедрена система мандатного управления доступом, группам и пользователям доступ предоставляется на основе их допуска и уровня классификации данных.

Многоэкземплядность

Иногда компании не хотят, чтобы пользователи, находящиеся на низком уровне безопасности, имели доступ и изменяли данные на более высоком уровне. Это может реализовываться различными способами. Одним из подходов является отказ в доступе, если пользователь на более низком уровне пытается получить доступ к данным на более высоком уровне. Однако такой подход может давать этому пользователю косвенную информацию о том, что на том уровне, к которому он попытался получить доступ, есть что-то важное.

Другим подходом для реализации такого ограничения является **многоэкземплядность** (polyinstantiation), которая позволяет хранить в таблице несколько экземпляров записи, имеющих одинаковый первичный ключ, при этом каждому экземпляру будет присвоен свой уровень безопасности. После вставки информации в базу данных, доступ субъектов с низкого уровня должен быть ограничен. Однако вместо того, чтобы просто ограничить доступ, создается другой набор данных, целью которого является обман субъектов с нижнего уровня, которые в случае подобного запроса получают специально подготовленную недостоверную информацию. Например, если военно-морская база осуществляет поставки оружия из штата Делавер на Украину на судне Оклахома, сведения об этом могут быть классифицированы как «совершенно секретные». Только субъекты, имеющие допуск к «совершенно секретной» информации и выше должны иметь возможность ознакомиться с этой информацией. Для организации ограничения, создаются поддельные записи базы данных, в которых указано, что судно Оклахома осуществляет перевозку из Делавера в Африку продуктов питания, и этой информации присваивается класс «неклассифицировано», как показано в Таблице 9-1. Поскольку судно Оклахома стоит у берега и на него грузят какие-то контейнеры, всем понятно что оно будет осуществлять некие перевозки, однако люди с более низким уровнем допуска будут думать, что оно поплывет с продуктами в

Африку, а не с оружием на Украину. Это также исключит какие-либо домыслы людей с низким уровнем допуска в отношении миссии этого судна. Все будут знать, что судно Оклахома занято, и при планировании своих перевозок будут рассматривать другие суда.

Уровень	Судно	Груз	Отправитель	Получатель
Совершенно секретно	Оклахома	Оружие	Делавер	Украина
Неклассифицировано	Оклахома	Продукты	Делавер	Африка

Таблица 9-1. Пример многоэкземплярности, дающий полную, но недостоверную информацию субъектам с низким уровнем допуска

В рассмотренном примере многоэкземплярность была использована для создания двух версий одного и того же объекта, с целью скрытия от субъектов на низком уровне безопасности истинной информации, а также предотвращения их попыток использования и изменения этих данных тем или иным способом. Это способ предоставления альтернативной истории субъектам, не имеющим необходимого уровня допуска, чтобы знать правду. Но это лишь один из примеров применения многоэкземплярности. Напрямую это не связано с безопасностью, однако чаще всего это используется именно в целях обеспечения безопасности критичной информации. В любом случае, если создается копия объекта, которая заполняется измененными данными, т.е. в базе данных хранятся два экземпляра одного и того же объекта, имеющие различные атрибуты, это означает применение многоэкземплярности.

Обработка транзакций в режиме реального времени

Обработка транзакций в режиме реального времени (OLTP – online transaction processing) обычно используется при кластеризации баз данных для обеспечения отказоустойчивости и высокой производительности. OLTP предоставляет механизмы, которые отслеживают возникновение проблем и надлежащим образом решают их. Например, если в программном процессе происходит сбой, и он прекращает функционирование, механизмы мониторинга OLTP могут обнаружить это и попытаться перезапустить этот процесс. Если перезапустить процесс не удастся, производится откат транзакции, для предотвращения повреждения данных или выполнения только части транзакции. Любая выявленная ошибочная или некорректная транзакция должна быть записана в журнал транзакций. В этом журнале также сохраняются сведения об успешно выполненных транзакциях. Информация записывается в журнал до и после выполнения транзакции, создавая таким образом отчет о событиях.

Основной целью использование OLTP является обеспечение того, что транзакция либо выполняется правильно, либо не выполняется совсем. Транзакция обычно является неделимым набором связанных операций. Если не выполнена хотя бы одна операция из этого набора, должен быть произведен откат всей транзакции, что гарантирует сохранение целостности и правильности информации в базе данных.

Набор систем, участвующих в выполнении транзакции, управляется и контролируется программным обеспечением OLTP, которое отслеживает, чтобы все проходило гладко и корректно.

OLTP может, при необходимости, выполнять балансировку нагрузки и распределять входящие запросы по системам, входящим в кластер. Если число запросов к базе данных возрастет и приведет к снижению производительности одной из систем, OLTP может перенаправить некоторые из этих запросов на другие системы. Это гарантирует обработку всех запросов, при этом пользователям не нужно будет долго ждать завершения транзакции.

При использовании нескольких экземпляров базы данных, важно обеспечить наличие в них одинаковой информации. Рассмотрим следующий пример: Кэти идет в банк и снимает со своего счета 65 000 руб. из имеющихся 100 000 руб. База данных А получает запрос на изменение данных и сохраняет новый остаток по ее счету, составляющий 35 000 руб., но база данных Б не обновляется. В ней по-прежнему остается информация об остатке на счете Кэти

в размере 100 000 руб. На следующий день Кэти обращается в банк с просьбой предоставить ей информацию об остатке на ее счете, и этот запрос направляется в базу Б, содержащую некорректную информацию, поскольку транзакция не была перенесена в эту базу данных. Чтобы избежать подобной ситуации, OLTP гарантирует, что транзакция не будет завершена до тех пор, *пока все базы данных не получат и не сохранят все необходимые изменения*.

OLTP записывает транзакции по мере их осуществления (т.е. в режиме реального времени), а в распределенной среде транзакции обычно затрагивают несколько баз данных. Это ведет к сложностям, которые, в свою очередь, могут стать причиной нарушения целостности данных, поэтому программное обеспечение базы данных должно соответствовать требованиям ACID:

- **Атомарность** (Atomicity). Разделяет транзакции на единицы выполнения и обеспечивает, что никакая транзакция не будет зафиксирована в базе данных частично. Либо изменения фиксируются в полном объеме, либо производится откат базы данных к предыдущему состоянию.
- **Согласованность** (Consistency). Транзакция должна следовать политике целостности, разработанной в отношении этой конкретной базы данных, и обеспечивать, что все данные в различных базах данных согласованы друг с другом.
- **Изоляция** (Isolation). Транзакции выполняются изолированно до момента их завершения, без взаимодействия с другими транзакциями. Результаты выполняемых транзакцией изменений недоступны, пока она не будет полностью завершена.
- **Надежность** (Durability). Только после того, как точность сохранения транзакции проверена во всех системах, она фиксируется, после чего откат базы данных к предыдущему состоянию становится невозможен.

9.7. Хранилища и интеллектуальный анализ данных

Хранилища данных (data warehousing) объединяют данные из нескольких баз данных или источников данных в большие базы данных, что позволяет осуществлять исчерпывающий поиск данных и их всесторонний анализ. Данные извлекаются из различных баз данных и передаются в централизованное хранилище. При этом производится нормализация данных, т.е. из них удаляется избыточная информация и они преобразуются в необходимый для хранилища формат. Это позволяет пользователям работать только с одной системой, а не отправлять запросы в различные базы данных.

Базы данных, являющиеся источниками данных для хранилища, используются для ведения обычной операционной деятельности. А хранилище данных используется для анализа данных, выполняющегося с целью подготовки прогнозов, необходимых для принятия бизнес-решений, оценки эффективности маркетинга, рыночных тенденций и даже для выявления мошеннической деятельности.

Хранилище данных не является простой зеркальной копией данных, собранных из различных баз данных и сохраненных в одном месте. Хранилище – это база данных, информация в которой прошла предварительную обработку, в ходе которой связанные данные обобщаются и коррелируются. В результате данные представляются пользователю в более удобном, понятном и сокращенном виде.

Централизованное хранение данных обеспечивает более легкий доступ к ним и упрощает управление ими, однако нужно учитывать, что это требует повышенных мер безопасности. Если злоумышленник получил доступ к хранилищу данных, он может получить все данные компании сразу.

Интеллектуальный анализ данных (data mining) – это процесс выявления в данных, содержащихся в хранилище данных, новой, полезной, неизвестной ранее информации.

Инструменты интеллектуального анализа данных применяются для поиска связанных данных, проведения корреляции данных с целью создания метаданных. Метаданные могут позволить увидеть ранее неизвестные связи и зависимости между отдельными наборами информации. Они могут помочь при выявлении необычных последовательностей действий, новых шаблонов поведения. В качестве простого примера эффективного применения интеллектуального анализа данных можно рассмотреть пример процесса выявления мошенничества при страховании. Предположим, что в хранилище данных содержится информация о страховых случаях миллионов клиентов различных страховых компаний. Для выявления закономерностей в этой информации применяются средства интеллектуального анализа данных. При этом может быть выявлено, что через несколько месяцев после каждого переезда клиента Джона Смита, он обращался в страховую компанию для получения компенсации в связи со страховым случаем. Так, он переехал в 1967 году и спустя два месяца в его новом доме произошел подозрительный пожар, затем он переехал в 1973 году и через три месяца у него украли мотоцикл, он снова переехал в 1984 году и в через два месяца у него обворовали квартиру. Такие взаимосвязи очень сложно выявить вручную, особенно если клиент пользовался услугами различных страховых компаний, которые просто обновляли свои данные, но не организовывали их централизованное хранение и глубокий анализ.

При интеллектуальном анализе данных выполняется упрощение сложных данных с использованием математических функций, основанных на нечеткой логике (fuzzy logic) и теории множеств (set theory), а также экспертных систем. Именно такие методы позволяют найти незаметные ранее закономерности. Часто происходит так, что ценность полученных таким образом метаданных, значительно превышает ценность самих данных, из которых они были получены, поэтому они должны быть очень хорошо защищены.

Целью создания хранилищ данных и проведения интеллектуального анализа данных является извлечение информации и получение знаний о деятельности компании, происходящих в ней тенденциях, как это показано на Рисунке 9-10. Эти знания могут помочь руководству компании выявить недостатки и найти пути для оптимизации деятельности. Например, если наша компания работает в розничной торговле, нам хотелось бы, чтобы клиенты тратили в наших магазинах как можно больше денег. Чтобы улучшить бизнес нашей компании, нам нужно лучше понимать привычки, поведение и психологию клиентов при совершении покупок. Если конфеты и другие мелкие товары поместить на полку рядом с кассой, объем покупки этих товаров возрастает на 65% по сравнению объемом покупки тех же товаров, если они находятся в другом месте магазина. Если один из наших магазинов находится в местности, где проживают более обеспеченные люди, и мы видим, что в этом магазине есть постоянный (или даже растущий) спрос на дорогие вина, мы можем сделать вывод, что это самое подходящее место для продажи дорогих сортов сыра и продуктов для гурманов. Продавать эти продукты в других магазинах, где обслуживаются в основном менее обеспеченные клиенты, большого смысла не имеет.

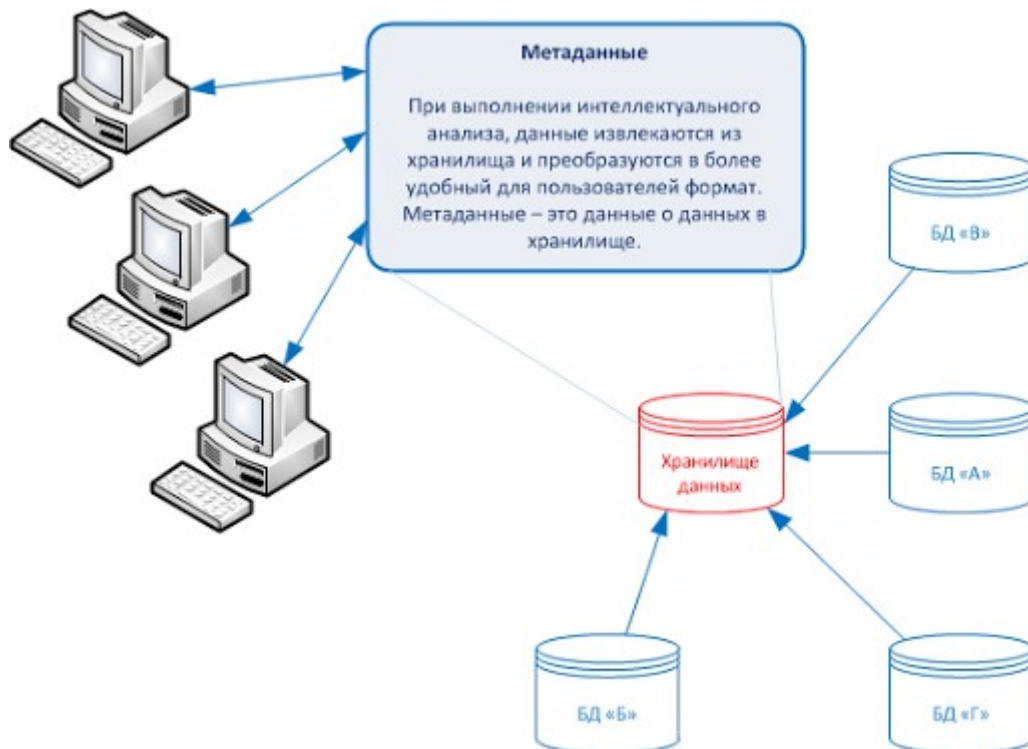


Рисунок 9-10. Инструменты интеллектуального анализа данных используются для выявления связей между данными в хранилище.

ПРИМЕЧАНИЕ. Интеллектуальный анализ данных – это процесс анализа содержимого хранилища данных с помощью инструментов, которые пытаются выявить тенденции, провести корреляцию, найти взаимосвязи и аномалии, не зная смысла данных. Метаданные – это результат обработки данных из хранилища данных с помощью специальных инструментов. На вход хранилища данных поступают данные, а на выходе из него формируются метаданные.

Компания должна выполнять эту работу, если она хочет иметь подобные данные для принятия более эффективных решений и получения конкурентных преимуществ. Это не просто агрегирование информации, это позволит руководству лучше понять различные аспекты деятельности компании, выбрать пути для повышения рентабельности бизнеса, повышения продуктивности работы персонала.



Интеллектуальный анализ данных также называют *обнаружением знаний в базе данных*

(KDD – knowledge discovery in database), он представляет собой комбинацию методов, позволяющих выявить реальные и полезные закономерности. Различные типы данных могут иметь различные взаимосвязи, поэтому метод анализа выбирается в зависимости от типа данных и от искомых закономерностей. Ниже приведены три подхода, используемые в системах KDD для выявления этих закономерностей:

- **Классификационный** (Classification). Выполняется группировка данных, имеющих те или иные сходства.
- **Вероятностный** (Probabilistic). Выявляет взаимозависимости данных и определяет вероятности их взаимосвязи.
- **Статистический** (Statistical). Выявляет взаимоотношения между элементами данных и пытается найти закономерности.

В Таблице 9-2 приведены различные типы систем, используемые в зависимости от требований к получаемому результату.

	Система, основанная на данных	Система, основанная на правилах	Система, основанная на знаниях
Может работать с	Данными	Данными и правилами	Данными, правилами и знаниями
Результатом может быть	Информация	Информация, решения, решения в режиме реального времени	Информацию, решения, ответы, экспертные советы, рекомендации
Обычно использует	Жестко запрограммированные правила Процедурные языки	Основана на правилах Декларативные языки	Нечеткая логика, вероятностные суждения, другие техники на основе искусственного интеллекта и экспертных систем
Идеально для	ИТ- и системных правил	Простых бизнес-правил	Сложных бизнес-правил
Лучше всего подходит для этих типов приложений	Традиционные информационные системы	Принятие решений, обеспечение соответствия требованиям	Получение советов и рекомендаций, выбор продуктов, поиск и устранение проблем
Интеллектуальные возможности	Отсутствуют	Неглубокая логика Широкая область	Глубокая логика Узкая область

Таблица 9-2. Различные типы систем, используемых в зависимости от потребностей

10. Разработка систем

Для реализации наиболее эффективной системы безопасности, она должна планироваться и управляется на протяжении всего жизненного цикла программного обеспечения, а не реализовываться с помощью сторонних средств, интегрированных во фронтальную часть системы после ее разработки. На протяжении своего жизненного цикла, продукт сталкивается со множеством различных рисков безопасности и инцидентов, поэтому вопросы безопасности должны учитываться с самого начала, с этапа планирования продукта, и на всех последующих этапах – проектирования, кодирования, внедрения и эксплуатации. Если безопасность была добавлена на завершающем этапе разработки продукта, а не учитывалась на каждом этапе его жизненного цикла, стоимость и время, необходимые для обеспечения безопасности такого продукта, резко возрастают. Безопасность не следует рассматривать как короткий спринт, это длинный марафон со множеством препятствий.

Многие разработчики, программисты и архитекторы знают, что добавление безопасности на поздних этапах реализации системы значительно дороже и сложнее, чем ее интеграция в систему, начиная с этапов планирования и проектирования. Различные компоненты безопасности при их реализации на поздних этапах реализации системы, могут оказать негативное воздействие на многие аспекты ее функционирования, ограничивая ряд уже разработанных функций и заставляя систему работать нестандартными и не

предусмотренными способами. Такой подход обходится дороже, поскольку разработчикам нужно вернуться к проектированию, внести изменения в код программы, пересмотреть отдельные аспекты архитектуры системы.

10.1. Управление разработкой

Многие разработчики знают, что хорошее управление проектом обеспечивает движение проекта в правильном направлении, наличие необходимых ресурсов и информации, а также планов на случай возникновения проблем и непредвиденных обстоятельств (по принципу «надейся на лучшее, но готовься к худшему»). Управление проектами является важной частью разработки программных продуктов, а управление безопасностью является важной частью управления проектами.

План обеспечения безопасности должен быть составлен в начале проекта разработки и интегрирован в функциональный план. Это гарантирует, что безопасность не будет забыта. Первоначальный план носит общий характер, охватывает весь проект и ссылается на документы, содержащие более подробную информацию. Он может ссылаться на компьютерные стандарты (RFC, стандарты IEEE и лучшие практики), документы, разработанные в рамках предыдущих проектов, политики безопасности, планы реагирования на инциденты, национальные или международные руководящие документы (Оранжевая книга, Красная книга, Общие Критерии и т.п.). Это поможет обеспечить эффективность плана.

У плана обеспечения безопасности должен быть собственный жизненный цикл. Он должен дополняться, изменяться и детализироваться по мере выполнения проекта. Важно поддерживать его в актуальном состоянии, чтобы на него можно было ссылаться в будущих проектах. Эффективно контролировать работы и решения при реализации крупного и сложного проекта – непростая задача.

Планирование безопасности и деятельность в рамках управления проектом должны контролироваться, чтобы сохранять уверенность в правильности принятых решений в отношении безопасности. Если требуется обеспечить гарантии необходимого уровня защиты, нужно подтвердить, что безопасность учитывалась на каждом этапе жизненного цикла проекта, при выполнении определенных процедур, в процессе разработки, при принятии решений и выполнении иной деятельности в рамках проекта. Документация по проекту должна в точности отражать все аспекты процесса разработки продукта, а также все аспекты его функционирования после внедрения в реальную среду.

10.2. Этапы жизненного цикла

Для разработки программного обеспечения может использоваться несколько различных типов моделей, которые используют различные жизненные циклы. В этом разделе описаны основные компоненты, общие для всех этих моделей. В основном, каждая модель выполняет одно и то же, главное различие между ними заключается в том, как именно разбиты на части разработка и эксплуатация системы.

Проект может начинаться просто с хорошей идеи, что потребует импровизации от программистов и инженеров, либо проект может быть изначально тщательно продуман и структурирован, чтобы следовать определенным жизненным циклам, а программисты и инженеры должны следовать этому плану. Первый вариант вначале может показаться более интересным, поскольку команда может пропустить скучные требования, забыть про документацию и произвести продукт в более короткие сроки и в рамках бюджета. Однако команда, которая потратит время на проработку всех сценариев каждого из этапов жизненного цикла, в итоге получит больше удовольствия, т.к. ее продукт будет более качественным, клиенты будут больше доверять ему, а команда заработает больше денег в долгосрочной перспективе, ей не нужно будет хаотично разрабатывать и поддерживать патчи, закрывающие дыры в системе безопасности, пропущенные изначально.

Различные модели, так или иначе, содержат следующие этапы:

- Инициирование проекта
- Функциональное проектирование и планирование
- Техническое задание на разработку системы
- Разработка программного обеспечения
- Установка / внедрение
- Эксплуатация / сопровождение
- Удаление

В этот перечень не включена безопасность в виде отдельного пункта. Это связано с тем, что безопасность должна быть частью каждого из перечисленных этапов. Решение вопросов безопасности уже после реализации продукта, стоит гораздо больше, чем решение этих вопросов в процессе разработки продукта. Основной движущей силой продукта является функциональность, и разработчикам нужно многое обдумать в этом направлении, однако настоящий Домен посвящен вопросам *безопасности*, которые должны быть учтены на каждом из этапов жизненного цикла продукта.

Инициирование проекта

На этом этапе все пытаются понять, зачем нужен проект и каковы его границы. Проект может быть направлен на реализацию конкретных потребностей конкретного клиента, либо на удовлетворение возникшего на рынке спроса на новый продукт. На этом этапе команда управления проектом анализирует необходимые характеристики и функциональность нового продукта, проводит мозговые штурмы и рассматривает возможные ограничения.

Должно быть разработано концептуальное определение проекта, что обеспечит правильное понимание проекта всеми участниками, и позволит разрабатывать продукт более эффективно и результативно. На этом этапе может проводиться оценка уже имеющихся на рынке продуктов, определение требований, которым имеющиеся продукты не удовлетворяют. Также это может быть прямой запрос от клиента на разработку конкретного продукта.

В любом случае, предназначен ли разрабатываемый продукт для конкретного клиента или для всего рынка, должен быть проведен первоначальный анализ продукта, должно быть сформулировано первоначальное высокоуровневое заявление, в котором будут указаны необходимые для реализации этого проекта ресурсы, прогнозируемые сроки разработки. Также должна быть проведена оценка ожидаемой прибыли от продукта. Эта информация должна быть представлена высшему руководству, которое будет принимать решение, нужно ли переходить к следующему этапу или требуется дополнительная информация.

На этом этапе должны быть определены потребности пользователя и подтверждены основные цели безопасности продукта. Должно быть установлено, будет ли продукт обрабатывать критичные данные, и, если да, должны быть определены уровни критичности данных. Должен быть проведен первоначальный анализ рисков, в рамках которого будут оценены угрозы и уязвимости, оценка соотношения стоимости и преимуществ различных мер безопасности. Необходимо учесть вопросы, относящиеся к безопасности: вопросы целостности, конфиденциальности и доступности. Должны появиться очертания целевого уровня для каждого атрибута безопасности.

После проработки базовой структуры безопасности, которой будет следовать проект, должны быть организованы процессы управления рисками. Управление рисками должно продолжаться в течение всего жизненного цикла проекта. Информацию о рисках можно начинать собирать и анализировать уже на этапе инициирования проекта, она будет детализироваться по мере выполнения этапов функционального проектирования и

разработки технического задания.

Управление рисками

Одним из наиболее важных аспектов управления рисками, является умение задавать правильные вопросы. Мы уже рассматривали управление рисками в Домене 01, поэтому в этом Домене мы рассмотрим только те риски, которые непосредственно влияют на бизнес в целом. Управление рисками должно продолжаться на этапах разработки и внедрения программного обеспечения.

В процессе разработки программного обеспечения, обычно все фокусируются на создании богатой функциональности и скорейшем выпуске продукта на рынок. Очень часто безопасность не учитывается должным образом или она быстро отходит на второй план, когда начинают «поджимать» сроки. Для обеспечения безопасности продукта недостаточно только того, чтобы программисты знали о методах безопасного программирования, безопасность должна пронизывать весь проект. Разработчики программного обеспечения должны учесть сценарии реализации угроз безопасности и предусмотреть соответствующие решения. Однако в действительности безопасность никогда не рассматривается, как один из важных компонентов процесса разработки. Разработчики не вспоминают про безопасность, пока продукт не купят, а покупатели не столкнутся с успешными атаками, основанными на уязвимостях, вызванных организацией процесса разработки продукта. Но будет уже слишком поздно, чтобы надлежащим образом интегрировать безопасность в проект. Вместо этого разработчики подготовят и выпустят патч.

Первыми шагами в управлении рисками является выявление угроз и уязвимостей, расчет уровня риска. После оценки всех рисков, руководство должно принять решение о приемлемом уровне риска. Конечно, было бы неплохо, чтобы руководство не принимало вообще никаких рисков, чтобы продукт был разработан максимально качественно и всесторонне протестирован и защищен «от дурака», однако это слишком сильно увеличило бы сроки разработки продукта и значительно повысило бы его стоимость. Необходимо пойти на определенные компромиссы и принять соответствующие решения, чтобы обеспечить баланс между рисками и экономической целесообразностью.

Анализ рисков

Анализ рисков проводится для выявления рисков, связанных с продуктом, и возможных последствий их реализации, с которыми клиент может столкнуться при использовании этого разрабатываемого продукта. Обычно в рамках процесса анализа рисков задается множество вопросов, составляется длинный список уязвимостей и угроз, с указанием вероятности эксплуатации этих уязвимостей и последствий реализации каждой из угроз. Для различных продуктов задаются разные вопросы, они зависят от таких факторов, как цель продукта, ожидания в отношении среды, в которой он будет функционировать, задействованного персонала, а также типа бизнеса компаний, которые будут приобретать и использовать этот продукт. Ниже приводится краткий список вопросов, которые должны быть заданы в процессе анализа рисков программного обеспечения:

- Существует ли возможность переполнения буфера, как ее избежать и протестировать?
- Выполняет ли продукт надлежащую проверку формата / правильности всех данных, вводимых пользователем?
- Какие источники угроз существуют во внешней и внутренней среде? Что это за источники?
- Бизнес какого типа зависит от этого продукта, в бизнесе какого типа может возникнуть ущерб, если продукт не будет работать некоторое время?
- Существуют ли угрозы утечки информации через скрытые каналы, которые должны

быть учтены?

- Отказоустойчивость какого типа должен обеспечивать продукт, и когда реализация этого будет инициирована?
- Необходимо ли шифрование? Какого типа? Какая требуется стойкость?
- Нужны ли планы действий на случай экстренных ситуаций?
- Будет другая сторона (например, интернет-провайдер или хостинг-провайдер) сопровождать этот продукт для клиента?
- Необходим ли мобильный код? Зачем? Как он может быть реализован?
- Будет ли этот продукт работать в среде, подключенной к сети Интернет? Какие последствия это может иметь для продукта?
- Нужно ли этому продукту взаимодействовать с уязвимыми системами?
- Уязвим ли этот продукт к DoS-атакам?
- Уязвим ли этот продукт для вирусов?
- Необходимы ли механизмы предупреждения о вторжениях?
- Будут ли у сотрудников клиента или внешних лиц мотивы саботировать этот продукт? Зачем? В чем может выражаться такой саботаж?
- Будут ли у компаний-конкурентов клиента мотивы совершить мошенничество с помощью этого продукта? Зачем? Как может быть реализовано такое мошенничество?
- Какие другие системы будут затронуты, если этот продукт выйдет из строя?

Это только небольшой список, каждый из вопросов которого должен делиться на ряд более детальных вопросов, что необходимо для выявления и учета всех возможных угроз и рисков.

После выявления всех рисков, нужно количественно оценить вероятность и последствия их реализации, чтобы обеспечить выбор и применение надлежащих контрмер в процессе разработки и в самом продукте. Если продукт будет использоваться только для создания текстовых документов, для него потребуются меньший уровень защиты и меньший объем тестирования, по сравнению с продуктом, который будет работать с данными банковских карт.

Большинство шагов процесса анализа рисков, описанных в Домене 01, могут быть применены и в процессе анализа рисков при разработке продукта. После того, как членами проектной команды выявлены все угрозы, рассчитана вероятность их реализации и последствия этого, риски могут быть перечислены в порядке их критичности. Если существуют риски высокого уровня, реализация которых может разорить клиентов, такие риски должны быть указаны в самом верху списка. Риск с низким уровнем критичности следует расположить внизу списка. Таким образом мы получим список, в котором наиболее вероятные и потенциально разрушительные риски идут первыми, а менее вероятные и менее опасные риски следуют за ними.

Эти риски должны быть учтены в архитектуре продукта, также как и функциональность продукта, они требуют выполнения определенных процедур на этапах разработки и сопровождения. Например, архитектура банковского программного обеспечения может требовать установки фермы веб-серверов в демилитаризованной зоне (DMZ), а других компонентов и базы данных – за межсетевым экраном, чтобы обеспечить дополнительный уровень защиты. Архитектура этого продукта будет включать разделение компонентов системы, что потребует разработать методы взаимодействия между различными компонентами. Если продукт будет предоставлять функции безопасной электронной почты, все риски, связанные с работой такого сервиса, должны быть проанализированы и должным

образом учтены. Также должны быть продуманы и учтены процедуры внедрения. Каким образом клиент установил этот продукт? Каковы системные требования и требования к среде? Этот продукт должен работать с инфраструктурой открытых ключей (PKI)? Для многих продуктов очень важен уровень поддержки после установки продукта. Потребуется ли от производителя держать клиентов в курсе выявляемых проблем безопасности? Требуется ли ведение журналирования и аудита? Чем больше таких вопросов будет продумано в самом начале, тем меньше проблем возникнет в конце этого процесса.

Важно понимать разницу между анализом рисков проекта и анализом рисков безопасности. Часто эти понятия считают синонимами. Проектная группа может провести анализ рисков только в отношении самого проекта. Это сильно отличается от анализа рисков безопасности, при котором рассматриваются различные угрозы и недостатки. Понимать и выполнять нужно оба этих анализа, но разными способами.

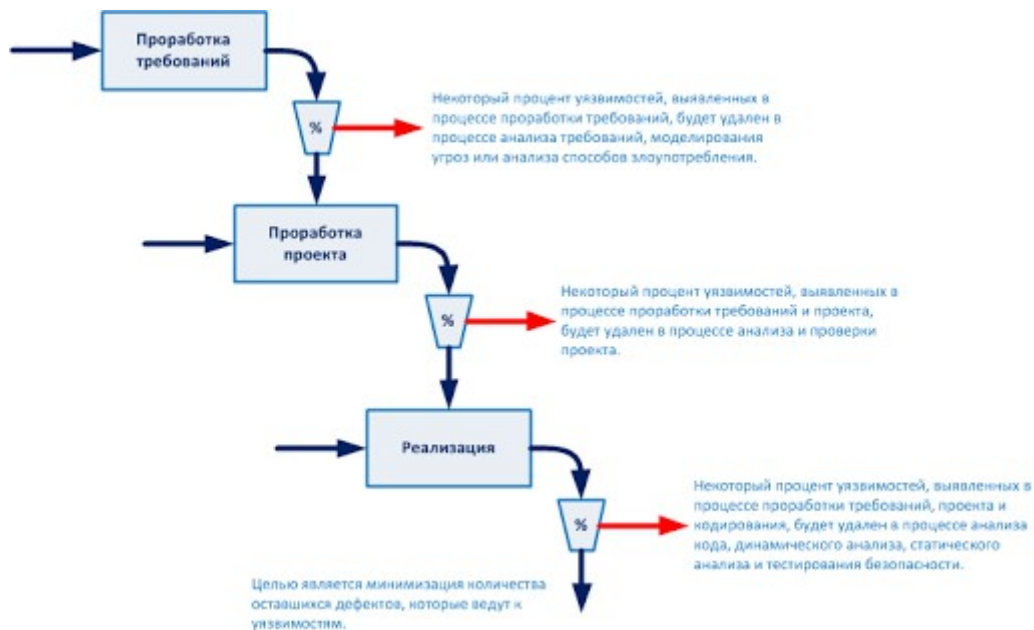
Функциональное проектирование и планирование

На этом этапе план проекта уже разработан, определена архитектура программного обеспечения и действия, необходимые для обеспечения безопасности и создания контрольных точек безопасности, гарантирующих качество реализованных защитных мер, определены процессы управления конфигурацией и изменениями. К данному моменту определены ресурсы проекта, начато формирование графиков тестирования, разработаны критерии оценки, позволяющие надлежащим образом проверить реализованные защитные меры. Формально оформлены функциональные требования, т.е. ожидания от продукта изложены официально (как правило, описаны в соответствующих документах). Разработан план тестирования, который будет актуализироваться на каждом этапе, что обеспечит надлежащую проверку всех вопросов.

Требования безопасности могут быть получены из нескольких различных источников:

- Функциональные требования к программному обеспечению
- Государственные, международные или локальные (на уровне компании) стандарты и руководящие принципы
- Экспортные ограничения
- Уровень критичности обрабатываемых данных (например, военные стратегические данные или данные обычной коммерческой компании)
- Соответствующая политика безопасности
- Анализ затрат и выгод
- Требуемый уровень защиты для достижения целевого рейтинга уровня гарантий (assurance level rating)

Первоначальная оценка риска, скорее всего, будет актуализироваться по мере реализации проекта в результате появления и изучения дополнительной, более детальной информации. В некоторых проектах, проводить анализ рисков нужно несколько раз – на разных этапах жизненного цикла. Например, при первоначальном анализе рисков, проектная команда может знать, что продукт будет выполнять идентификацию и аутентификацию доменных пользователей, что требует среднего уровня безопасности. На более позднем этапе жизненного цикла может выясниться, что этот продукт должен работать еще и с биометрическими устройствами и иметь возможность интеграции с системами, требующими высокого уровня безопасности. При этом проектная команда должна будет провести новый анализ рисков в полном объеме, т.к. появились новые важные аспекты.



На этом этапе учитывается функциональность, требуемая от продукта и указанная в проектной документации. Если продукт разрабатывается для конкретного заказчика, проектная документация используется в качестве инструмента, помогающего убедить заказчика, что команда разработки понимает его требования к продукту. Проектная документация, как правило, разрабатывается аналитиками, под руководством разработчиков и архитекторов, а затем предоставляется заказчику. Изучая ее, заказчик решает, нужно ли добавить какие-либо функции или что-то убрать, после чего в документацию вносятся соответствующие изменения. После согласования проектной документации с заказчиком, команда разработчиков вместе с заказчиком может конкретизировать все ожидания заказчика по отношению к продукту.

Вопросы в отношении безопасности должны быть заданы при обсуждении высокоуровневых вопросов. Примерами таких вопросов могут быть: Нужна ли аутентификация и авторизация? Нужно ли шифрование? Должен ли продукт взаимодействовать с другими системами? Будет ли осуществляться доступ к продукту напрямую из сети Интернет?

Многие компании пропускают этап функционального проектирования и сразу перепрыгивают на разработку технических требований для продукта. Либо они разрабатывают проектную документацию, но не показывают ее клиенту. Это может привести к значительным задержкам и вызвать необходимость пересмотра проекта, поскольку перед тем, как начать прорабатывать детали, требуется продумать в общих чертах продукт в целом. Если клиент не участвует в этом процессе, вполне вероятно, может получиться так, что он думает, что разработчики создают продукт, который выполняет X, тогда как команда разработчиков считает, что клиент хочет Y. Много времени можно потратить впустую, разрабатывая продукт, не являющийся тем, что на самом деле хочет клиент. Поэтому необходимо четко определить направление и цели перед тем, как приступить к кодированию. Обычно это является важной функцией команды управления проектом.

Техническое задание на разработку системы

Требования к программному обеспечению могут исходить из трех моделей:

- **Информационная модель.** Указывает, какой тип информации будет обрабатываться, и как она будет обрабатываться.
- **Функциональная модель.** Определяет задачи и функции, которые должно выполнять приложение.
- **Поведенческая модель.** Описывает состояния, в которых приложение будет

находиться в процессе и после определенных переходов.

Например, информационная модель антивирусного программного обеспечения определяет, какую информацию должна обрабатывать антивирусная программа: вирусные сигнатуры, модифицированные системные файлы, контрольные суммы критичных файлов и действия вирусов. Функциональная модель антивирусного программного обеспечения определяет, что программа должна быть способна сканировать жесткий диск, проверять сообщения электронной почты на наличие известных вирусов, контролировать критичные системные файлы, обновлять вирусные сигнатуры и собственный программный код. Поведенческая модель определяет, что при старте системы, антивирусная программа должна провести быстрое сканирование критичных областей. При этом переход компьютера в рабочее состояние будет событием, которое изменяет состояние антивирусной программы. Выявление вируса также будет событием, которое изменит состояние антивирусной программы, и она приступит к уничтожению вируса. Каждое из таких состояний должно быть учтено, чтобы гарантировать, что продукт не перейдет в небезопасное состояние и будет функционировать предсказуемым образом.

Данные информационной, функциональной и поведенческой моделей используются в качестве требований при проектировании программного обеспечения. В результате проектирования мы получаем такие данные, как архитектурный и процедурный проекты, показанные на Рисунке 9-11.



Рисунок 9-11. Информация всех трех моделей учитывается при проектировании

Архитекторы и разработчики берут сведения об организации данных и данные информационной модели, а затем преобразовывают их в структуры данных, которые необходимы для разработки программного обеспечения. Архитектурный проект определяет отношения между крупными структурами и компонентами программного продукта. Процедурный проект преобразует структурные компоненты в описательные процедуры.

Итак, к настоящему моменту выбраны механизмы контроля доступа, определены права доступа и разрешения, выбраны методы и алгоритмы шифрования, вопросы с обработкой критичных данных улажены, идентифицированы необходимые объекты и компоненты, оценены межпроцессные коммуникации, определены механизмы обеспечения целостности, проанализированы все остальные требования безопасности, определены соответствующие решения.

Для следующих этапов должна быть проведена декомпозиция работ (WBS – work breakdown structure), включая этапы разработки и внедрения, учитывая временной график и детализацию работ по тестированию, разработке, подготовке, интеграционному тестированию и поставке продукта.

Техническое задание – это инструмент, применяемый для описания требований пользователей и внутреннего поведения системы. Эти два элемента связываются друг с другом, чтобы показать, как внутреннее поведение реально выполняет требования пользователей.

Этот этап нужен для того, чтобы увидеть больше деталей о продукте и среде, в которую он будет внедрен. Требуемая функциональность была определена на предыдущем этапе. Данный этап учитывает механизмы, необходимые для реализации этой функциональности, и определяет, как будет разрабатываться код продукта, как он будет тестироваться и внедряться.

Необходимо учитывать модульность и возможность повторного использования продукта, или компонентов продукта. Код, который реализует критичные с точки зрения безопасности функции, должен быть максимально простым (чтобы в нем было проще обнаружить ошибки) и достаточно небольшим (чтобы можно было провести его всестороннее тестирование в различных ситуациях). Компоненты могут вызываться и использоваться различными частями продукта или другими приложениями. Возможность повторного использования компонентов продукта помогает оптимизировать продукт и обеспечить более эффективную и структурированную среду разработки.

Могут возникнуть проблемы переносимости продукта между различными платформами. Эти проблемы должны рассмотрены и решены на ранних этапах разработки продукта. Если продукт должен работать на Windows или Unix-системах, требования к разработке его кода будут существенно отличаться от требований к разработке кода приложения, которое будет устанавливаться только на мейнфреймах. Также должна быть рассмотрена среда, в которую будет устанавливаться этот продукт. Будет ли этот продукт использоваться отдельными пользователями, либо все пользователи сети получают тот или иной вариант доступа к этому продукту? Будет ли продукт однопользовательским или многопользовательским, имеет большое значение при разработке необходимых спецификаций.

Тестируемость (контролируемость) продукта и его компонентов должна быть продумана на этом раннем этапе, а не на более поздних этапах. Программисты могут внедрить в код *перехватчики* (hooks, хуки), которые покажут тестировщикам состояние продукта на разных этапах обработки данных. Одно только то, что работа продукта выглядит правильной, и он дает на выходе правильные результаты, не означает отсутствия внутренних ошибок. Именно поэтому для тестирования должен использоваться модульный подход, при тестировании нужно следовать за потоком данных, проверяя каждый шаг их обработки.

На этом этапе должны быть внимательно рассмотрены все вопросы, заданные при инициировании проекта, нужно убедиться, что по каждому учтенному вопросу разработаны соответствующие спецификации. Например, если в продукте будет выполняться аутентификация, на данном этапе будут изложены все детали осуществления этого процесса. Если при использовании продукта возникает большой риск мошенничества, на данном этапе должны быть определены все необходимые контрмеры, должно быть показано, каким образом они должны быть интегрированы в продукт. Если существует риск утечки информации через скрытые каналы, необходимо рассмотреть этот вопрос и разработать соответствующий псевдокод, который покажет, каким образом будет исключен или минимизирован этот риск.

Если продукт разрабатывается для конкретного клиента, спецификации продукта должны быть доведены до сведения этого клиента, чтобы еще раз убедиться, что все понимают продукт одинаково и все движутся в верном направлении. Этот этап предназначен для того, чтобы решить все вопросы, сложности, устранить любую путаницу до начала реальной разработки кода продукта.

Решения, принятые на данном этапе проектирования, являются ключевыми шагами для этапа

разработки. Проектирование – это единственный способ перевода требований клиента в программные компоненты, поэтому проектирование программного обеспечения является основой и значительно влияет на качество получаемого в результате продукта и его поддержки. Если продукт изначально не был хорошо спроектирован, последующие этапы станут гораздо более сложными.

Разработка программного обеспечения

Это этап, на котором к работе приступают программисты и разработчики. Обычно они участвуют в работе и до этого момента, указывая направления и давая советы, но на данном этапе на них падает вся основная работа.

На этом этапе программисты должны разрабатывать код, используя методы безопасного программирования, не допускающие возможности компрометации программного обеспечения. Также, программисты должны добавлять в программу код для проверки длины входящих данных, чтобы предотвратить переполнение буфера; проверять код программы на отсутствие скрытых каналов; проверять правильность использования типов данных; убедиться, что пользователи не смогут обойти контрольные точки; проверять синтаксис команд, а также рассчитывать значения контрольных сумм. Следует пройти по различным сценариям атак, чтобы увидеть, каким образом код может быть атакован или несанкционированно изменен. Проведение отладки и анализа кода должно выполняться одинаковыми по уровню профессионализма разработчиками, и все должно быть четко задокументировано.

Большинству программистов не нравится ничего документировать, и они ищут способ избежать этой задачи. Шесть – двенадцать месяцев спустя, никто уже не будет помнить конкретные вопросы, которые были рассмотрены, как они решались, как решались возникшие проблемы, либо программист, который знал все подробности, перейдет на работу к конкуренту или выиграет в лотерею и уедет жить на остров. Это еще один повод переработать и потратить дополнительные человеко-часы. Документация является чрезвычайно важной по различным причинам, она может сэкономить компании немалые деньги в долгосрочной перспективе.

Верификация и проверка

Верификация (Verification) определяет, насколько точно продукт отражает и соответствует техническому заданию. Поскольку в результате разработки может быть получен продукт, который не соответствует первоначальному техническому заданию, необходим этот шаг, гарантирующий отсутствие расхождений.

Проверка (Validation) определяет, обеспечивает ли продукт необходимые решения для соответствующей проблемы реального мира. В крупных проектах, легко потерять из виду основную цель. Проведение проверки гарантирует, что основная цель данного проекта достигнута.

Формальное и неформальное тестирование должно начинаться как можно раньше.

Тестирование модулей может начаться еще в самом начале разработки. Как только программист разработает компонент или модуль кода, проводится его проверка с несколькими различными значениями входных данных во множестве различных ситуаций.

Обычно тестирование модулей продолжается в течение всего процесса разработки. По окончании разработки официальное тестирование продукта должна провести абсолютно другая группа людей, в которую не входит никто из тех, кто разрабатывал продукт или ранее участвовал в тестировании его модулей. Это пример **разделения обязанностей**. Не должно быть так, что программист разрабатывает, тестирует, а затем и готовит окончательный релиз программного обеспечения. Чем больше глаз увидит код, чем больше рук поработают с программным обеспечением, тем выше вероятность того, что ошибки будут найдены до выпуска продукта.

Разумеется, любые программные перехватчики и закладки, вставленные программистами для тестирования или внесения изменений, должны быть удалены из приложения до того, как

оно начнет использоваться в реальной работе, поскольку они могут легко могут стать той дверью, через которую в продукт проникнут злоумышленники.

Не существует универсального рецепта для тестирования безопасности, поскольку программные продукты очень сильно различаются по своей функциональности и задачам безопасности. Важно связать риски безопасности с кодом и тестовыми задачами. Выполняя этот процесс линейно, можно выявлять уязвимости, готовить специальный тестовый сценарий, проводить тестирование, а затем анализировать код на предмет того, насколько хорошо он готов противодействовать реализации этой уязвимости. На этом этапе, должно быть проведено тестирование в близкой реальной среде, которая должна отражать производственную среду, что позволит убедиться в том, что код работает не только в лаборатории.

На данном этапе обычно проводятся атаки, пытающиеся скомпрометировать безопасность продукта, тесты на проникновение. Они направлены на выявление любых пропущенных уязвимостей. Оценивается функциональность, производительность приложения, а также его сопротивляемость попыткам взлома. Все необходимая функциональность продукта должна быть внесена в чек-лист для того, чтобы гарантировать, что в процессе тестирования учтена каждая функция.

Тесты безопасности должны быть проведены в отношении уязвимостей, выявленных ранее в процессе реализации проекта. Должны быть проведены попытки переполнения буфера, проведения атак, взлома программного обеспечения. Должна быть проверена реакция всех интерфейсов на ввод неожиданных данных, воздействие на систему DoS-атак, необычной работы пользователей. Если в программе происходит сбой, она должна надлежащим образом обработать эту ситуацию и вернуться обратно в безопасное состояние. Продукт должен быть проверен в различных средах с различными приложениями, конфигурациями и аппаратными платформами. Продукт может прекрасно работать при установке на чистую Windows 2000 на отдельном компьютере, но он может выдавать неожиданные ошибки при установке на ноутбук, который удаленно подключен к сети и, на котором установлен клиент SMS.

Разделение обязанностей. Различные типы сред (среда разработки, тестирования и эксплуатации) должны быть надлежащим образом разделены, их функциональность и выполняемые в них операции не должны дублироваться. Разработчики не должны иметь доступа к коду, работающему в промышленной среде. Код должен быть протестирован, сохранен в библиотеке, а уже затем установлен в промышленной среде. В процессе проведения тестирования модулей и официальном тестировании, все выявленные проблемы должны направляться команде разработчиков в виде отчетов по проблемам. Разработчики исправляют проблемы, после чего производится повторное тестирование. Этот процесс продолжается, пока все не убедятся, что продукт готов к промышленной эксплуатации. Если продукт разрабатывался для конкретного заказчика, он будет проводить ряд собственных тестов, прежде чем официально примет продукт. После официальной приемки продукта, он выпускается на рынок или передается заказчику.

ПРИМЕЧАНИЕ. Иногда разработчики вносят в продукт строки кода, которые позволяют им, нажав несколько клавиш, получить доступ в приложение в обход любых мер безопасности и средств контроля доступа. Это требуется разработчикам, чтобы они могли быстро получить доступ в приложение или к его коду на этапе разработки. Это называется «черным ходом» или «закладкой для поддержки». Они должны быть полностью удалены перед тем, как продукт начнет реально использоваться.

Установка и внедрение

Этап внедрения фокусируется на функционировании и эксплуатации разработанного программного обеспечения. На этом этапе клиент приобретает разработанный продукт и устанавливает его в своей среде. Затем продукт должен быть настроен для обеспечения требуемого уровня защиты. После этого должна быть протестирована функциональность и производительность приложения, его средства безопасности должны быть проанализированы и сопоставлены с требованиями, предъявляемыми компанией к безопасности.

Настройки должны быть документированы производителем, документация должна поставляться вместе с продуктом для использования клиентами. Должны быть разработаны руководства для пользователей, а также руководства по эксплуатации и сопровождению продукта. Изучив их, пользователи будут знать, как правильно работать с системой, а технический персонал будет знать, как правильно настроить продукт. Функционирование средств безопасности системы должно контролироваться, чтобы быть уверенным в работе системы в соответствии с тем, что обещано в соглашении об уровне обслуживания.

В период между внедрением и началом эксплуатации системы должна быть проведена ее аккредитация. Этот процесс следует за процессом сертификации, в рамках которого формально или неформально тестируются все функции безопасности, чтобы определить, удовлетворяют ли они требованиям безопасности, предъявляемым компанией. Сертификация представляет собой процесс анализа и оценки средств и функций безопасности. Как правило, эта задача ставится перед независимым внешним экспертом (вопросы сертификации и аккредитации были подробно рассмотрены в Домене 03).

Аккредитация – это официальное принятие системы руководством компании и явное принятие риска. При аккредитации рассматривается вся система, а не только отдельное приложение или усовершенствованная функция, поскольку безопасность – это сервис, который реализуется на разных уровнях системы и может проявляться по-разному. В процессе аккредитации представители руководства и технический персонал должны работать совместно, чтобы убедиться в качестве и уровне защиты, обеспечиваемом приобретенной и внедренной технологией. Технический персонал хорошо понимает вопросы функционирования системы и технические вопросы, а руководящий персонал понимает миссию компании, разбирается в ее финансовых вопросах и вопросах ответственности. Вместе они могут охватить большую область в процессе сертификации и аккредитации.

Если руководство убедилось, что новая система обеспечивает безопасность, понимает и принимает на себя остаточный риск, оно должно выпустить официальное заявление об аккредитации.

Для новой системы должен быть настроен и включен аудит, должен проводиться мониторинг происходящих в ней событий, должны быть разработаны планы и процедуры восстановления в случае чрезвычайных ситуаций. Указанные планы и процедуры должны быть протестированы, чтобы убедиться, что система в случае сбоя или чрезвычайной ситуации реагирует так, как это было запланировано.

Эксплуатация и сопровождение

Когда вы дошли до этого этапа, не думайте, что теперь работы по безопасности завершены и все под контролем. Напротив, на этапе эксплуатации безопасность является такой же или даже более важной задачей, по сравнению с предыдущими этапами.

Начальная часть этого этапа включает в себя настройку новой системы и ее правильную интеграцию с сетью и средой. Часто оказывается, что средства безопасности не включены или неправильно (для конкретной среды) настроены. Даже если они были изначально хорошо разработаны, это может оказаться совершенно не важным, если они не используются в действительности или используются ненадлежащим образом.

Операционные гарантии (operational assurance) обеспечиваются путем постоянного проведения тестирования на уязвимости, аудита и мониторинга событий. Именно благодаря деятельности по обеспечению операционных гарантий, администратор узнает о новых уязвимостях или компрометациях безопасности, и может предпринимать нужные действия.

Если в системе или среде происходят существенные изменения, может потребоваться провести новый анализ рисков, а также новую сертификацию и аккредитацию. Такими изменениями может быть добавление новых систем и / или приложений, переезд в другое здание или изменение уровня критичности обрабатываемых данных.

Удаление

Если пришло время заменять старое программное обеспечение на новое, должны быть выполнены определенные шаги, обеспечивающие безопасность такого перехода. Для этого могут потребоваться различные мероприятия, зависящие от уровня критичности данных, содержащихся в системе. Может потребоваться создать архивную копию информации или резервную копию всей системы вместе с данными, перед тем уничтожить данные выводимой из эксплуатации системы. Если удаляемые данные являются критичными, они должны быть уничтожены специальными методами, такими как многократная перезапись, размагничивание или физическое уничтожение носителя информации. Выбор способа уничтожения критичных данных зависит уровня их критичности, а также от политики компании по уничтожению информации.

Если выводимый из эксплуатации продукт является простым текстовым редактором или антивирусной программой, этот этап может быть простым. Но если этот продукт интегрирован во все компоненты инфраструктуры компании, надлежащее выведение его из эксплуатации без нанесения ущерба производительности работы и безопасности, может оказаться непосильной задачей.

Виды тестирования

Если нам нужна уверенность в качестве нашего программного обеспечения, мы должны протестировать его. Существуют различные виды тестов, через которые должно пройти программное обеспечение, направленные на поиск различных недостатков, которые могут быть в программе. Ниже приведены некоторые из наиболее часто используемых подходов к тестированию:

- **Тестирование модулей (Unit testing).** Отдельные компоненты помещаются в контролируемую среду, в которой программисты проверяют структуры данных, логику и граничные условия.
- **Интеграционное тестирование (Integration testing).** Проверка того, как компоненты работают вместе, и насколько их совместная работа соответствует описанию в проектной документации.
- **Приемочное тестирование (Acceptance testing).** Проверяет соответствие кода требованиям заказчика.
- **Регрессионное тестирование (Regression testing).** После внесения изменений в систему, проводится повторное тестирование, направленное на то, чтобы убедиться в функциональности, производительности и защите измененной системы.

В процессе тестирования программного обеспечения, мы должны подумать не только о том, что внутри и снаружи этого ящика, мы должны бросить ящик на пол, попинать его, покидать его об стены. Очень трудно представить себе все способы, которыми пользователи потенциально могут нанести вред программному продукту. Не менее трудно представить себе все способы, которые будут использовать хакеры, пытаясь взломать это программное обеспечение. Перечисленные ниже пункты, это лишь некоторые из вещей, которые должны быть сделаны в процессе тестирования программного обеспечения:

- Должны быть введены данные различных типов
- Должны быть введены данные из разных точек в пределах диапазона допустимых данных
 - Произвести проверку границ, чтобы выявить возможности для переполнения буфера
 - Проконтролировать процедуры проверки данных, чтобы убедиться, что

программное обеспечение принимает данные только тех типов, которые ему нужны (т.е. приложение не должно принимать буквы в строке ввода числовой информации и т.п.)

- Ввести данные, выходящие за пределы допустимого диапазона
- Проверить реакцию программы на различные действия пользователя
- Проверить данные до и после обработки, чтобы выявить неправильные изменения
- Проверить отсутствие возможности (уязвимости) повторного использования объекта
 - Субъект может получить несанкционированный доступ к остаточным данным в объекте или области памяти

Данные могут быть загрязнены различными способами на этапе загрузки в приложение, а также на этапе выгрузки из него (либо одновременно на обоих этих этапах). Чтобы убедиться в правильности обрабатываемых данных, необходимо реализовать следующие входные и выходные проверки:

- Проверка входных данных
 - Обнаружение и исправление ошибок
 - Значения дайджестов сообщений (хэш-функций)
 - Транзакционные и денежные счетчики
 - Меры контроля повторного представления (resubmission) и самопроверки
- Контроль на выходе
 - Обработка ошибок
 - Значения для проверки (reconciliation values)
 - Процедуры обработки
 - Журналы аудита и журналирование

Сбор мусора (garbage collection) – это автоматизированный механизм операционной системы, входящий в состав работ по управлению памятью. **Сборщик мусора** (garbage collector) находит выделенные ранее блоки памяти, которые больше не используются, и освобождает эти блоки, помечая их как свободные. Также он выполняет сбор рассеянных (scattered) блоков свободной памяти, и объединяет их в более крупные блоки. Это помогает обеспечить более стабильную среду и не тратить драгоценную память. Некоторые языки программирования, такие как Java, автоматически выполняют сбор мусора; другие, например, C, требуют, чтобы разработчик выполнял эту задачу вручную, что оставляет возможность для ошибок. Плохие парни постоянно будут пытаться взломать любое программное обеспечение, поэтому только целостное программирование и тестирование поможет защитить ваш продукт от деятельности злоумышленников.

Анализ заверченного проекта

Важно после завершения проекта собрать всю команду, чтобы обсудить проект в целом и те вещи, которые должны быть улучшены в следующий раз. Если отнестись серьезно к этому этапу и правильно его провести, компания может сэкономить деньги и время в будущих проектах, т.к. команда сможет извлечь уроки, проанализировать свои ошибки, чтобы не повторять их, упорядочить свои процессы. Все это будет способствовать тому, что следующий проект будет выполняться более плавно, с меньшим количеством ошибок и за меньшее время.

Это должно быть упорядоченным событием, кто-то должен вести встречу, кто-то должен делать пометки (вести протокол), но при этом встреча должна проходить в расслабленной атмосфере, каждый член команды должен чувствовать возможность выразить свое мнение и

идеи. Эта встреча не должна превратиться в поток взаимных обвинений или жалоб.

Некоторые компании не видят смысла в этом мероприятии и просто заканчивают один проект, чтобы начать следующий, который, скорее всего, столкнется с теми же проблемами, что и предыдущий проект. Выполнение проектов – это процесс обучения, а бизнес считает лучшим продуктом тот продукт, который сделан за минимальное время и с наименьшими затратами. Руководству нужно понимать, как эти два аспекта могут идти рука об руку, и убедиться, что завершающий анализ должен являться частью каждого проекта. Наиболее успешные компании оптимизируют процессы реализации проектов и управление проектами, они оттачивают свое мастерство, превращая работу над проектом в повторяемые процедуры, которые позволяют производить продукт ожидаемого уровня качества. Такие компании постоянно тратят время на то, чтобы посмотреть, каким образом можно усовершенствовать свои процессы.

Этапы жизненного цикла программного обеспечения. Ниже приведены общие этапы разработки программного обеспечения с указанием ключевых задач по безопасности, которые должны выполняться на каждом этапе.

- **Инициирование проекта**
 - Определение концепции проекта
 - Заявление и первоначальное изучение
 - Первичный анализ рисков
- **Функциональное проектирование и планирование**
 - Требования выявлены и определены
 - Спецификация системного окружения определена
 - Формальный проект разработан
- **Техническое задание на разработку системы**
 - Анализ функционального проекта
 - Разбивка функциональности
 - Проведение детального планирования
 - Проектирование кода
- **Разработка программного обеспечения**
 - Разработка кода программного обеспечения
- **Установка / внедрение**
 - Установка и внедрение продукта
 - Тестирование и аудит
- **Эксплуатация / сопровождение**
 - Изменения, исправления, а также незначительные модификации продукта
- **Удаление**
 - Замена продукта новым продуктом

В этом Домене описывается жизненный цикл, состоящий из семи этапов. В других моделях могут использоваться другие жизненные циклы, состоящие из иного количества этапов, но, по сути, на них будут решаться те же основные задачи.

10.3. Методы разработки программного обеспечения

За многие годы были созданы различные Методы разработки программного обеспечения (SDM – System Development Methods), направленные на удовлетворение различных требований разработчиков и поставщиков. Обычно эти методы называют руководствами по

разработке, которыми они на самом деле и являются, они помогают разработчикам на различных этапах создания программного обеспечения (анализ, проектирование, программирование, сопровождение). Существует множество таких методов, ниже представлены несколько из них:

- **Водопад (Waterfall).** Классический метод, в котором процесс разработки выглядит как поток, последовательно проходящий отдельные этапы проекта разработки. Этот метод требует полного и успешного завершения предыдущего этапа, его формального анализа и документирования перед переходом к следующему этапу проекта. Возврата к предыдущим этапам не происходит. Результат появляется только в конце разработки.
- **Спираль (Spiral).** Метод уделяет повышенное внимание анализу рисков, созданию прототипов и моделированию, выполняемым на различных этапах цикла разработки. Каждый виток спирали соответствует созданию фрагмента или версии программного обеспечения, после чего определяется его качество и планируются работы следующего витка спирали. Этот метод периодически возвращается к предыдущим этапам для уточнения целей и характеристик проекта. Метод сочетает в себе подходы методов прототипирования и водопада. Результат появляется, фактически, на каждом витке спирали, работы завершаются после того, как разработчик и заказчик придут к согласию относительно приемлемости полученного результата.
- **Структурное программирование (Structured Programming Development).** Методология программирования, которая применяет иерархическую структуру логических блоков и методы процедурного программирования. Структурирование программы с использованием подпрограмм (процедур и функций) минимизирует применение команд произвольного перехода (таких как GOTO) и делает акцент на единых точках входа и выхода. Такой иерархический подход упрощает понимание и дальнейшее внесение изменений в программу. Структурное программирование позволяет совместно использовать модули, что улучшает использование памяти.
- **Итеративная разработка (Iterative Development).** Метод, использующий циклический подход (Цикл Деминга - планирование-реализация-проверка-корректировка) к разработке программного обеспечения, при котором работы производятся параллельно с непрерывным анализом полученных результатов и корректировкой предыдущих этапов работы. В отличие от традиционных моделей, итеративная разработка фокусируется на планировании контрольных точек проекта (milestones) на основе ресурсов и времени, постоянной оценке соответствия текущего состояния проекта первоначальным целям. Итеративная разработка позволяет динамически оценивать состояние проекта в целом и вносить необходимые поправки для повышения эффективности реализации проекта.
- **Модифицированная модель прототипирования (MPM – Modified Prototype Model).** Метод, специально созданный, чтобы противостоять сложностям при разработке веб-приложений. Модифицированная модель прототипирования позволяет разработчикам быстро перевести требования клиента в отображаемый на экране прототип. Прототипы обычно используются разработчиками и заказчиками, когда они не уверены в характеристиках окончательного продукта. Использование прототипов позволяет уточнить окончательный продукт и сделать технические требования менее туманными.
- **Экспериментальная модель (Exploratory Model).** Метод, используемый в случаях, когда отсутствуют четко определенные цели проекта. Вместо концентрации на подробных задачах, экспериментальная модель использует набор спецификаций, заключающих в себе сведения о работе окончательного продукта. Тестирование

является важной частью экспериментальной разработки, поскольку оно позволяет убедиться, что текущий этап проекта соответствует вероятным сценариям реализации.

- **Совместная разработка (JAD – Joint Analysis Development).** Метод, который позволяет разработчикам напрямую взаимодействовать с пользователями в процессе разработки, а также акцентирует внимание на совместной (командной) работе разработчиков и специалистов, которые хорошо разбираются в методологии и проектировании.
- **Быстрая разработка приложений (RAD – Rapid Application Development).** Метод определения требований пользователей (с помощью прототипов) и быстрой разработки систем. Применяется циклический подход к разработке - каждая новая версия продукта основывается на оценке предыдущей версии заказчиком. Минимизация времени разработки достигается за счет переноса в новый продукт уже готовых модулей и добавления необходимой функциональности. Метод RAD может использоваться для решения срочных задач.
- **Модель повторного использования (Reuse Model).** Модель, в которой для разработки программного обеспечения используются уже существующие компоненты. Эта модель лучше всего подходит к реализации проектов, основанных на объектно-ориентированном программировании. Поскольку в этой модели программы не разрабатываются «с нуля», это значительно снижает стоимость и время разработки.
- **Чистая комната (Cleanroom).** Подход, который пытается предотвратить возможные неточности и ошибки, следуя структурированным и формализованным методам разработки и тестирования. Этот подход используется для разработки высококачественных и критичных приложений, которые будут проходить строгий процесс сертификации.
- **Компонентно-ориентированная разработка (Component-Based Development).** Модель, которая использует независимые и стандартизированные модули (компоненты), из которых собираются работоспособные программы. Каждый стандартный модуль содержит функциональный алгоритм или набор команд, а также интерфейс для взаимодействия с другими модулями. Примером таких модулей могут быть объекты, которые применяются в объектно-ориентированном программировании. Разработка на основе компонентов обеспечивает возможность повторного использования и расширения функциональности программ. Компонентно-ориентированная разработка широко используется в современном программировании, этот метод позволяет значительно снизить стоимость разработки программного обеспечения.
- **Экстремальное программирование (Extreme Programming).** Методология, которая обычно применяется в ситуациях, требующих быстрой адаптации под изменяющиеся требования заказчика. Экстремальное программирование придает особое значение обратной связи от заказчика для оценки результатов проекта. Принципы создания программного обеспечения при использовании экстремального программирования отбрасывают традиционное долгосрочное планирование, выполняемое для возможности повторного использования кода, и вместо этого фокусируется на создании простого кода, оптимизированного только для этого проекта. Экстремальное программирование исходит из предпосылки, что требования заказчика вероятнее всего существенно изменятся в рамках жизненного цикла проекта, и концентрируется на процессе разработки, чтобы подстроиться под эти изменения.

10.4. Средства автоматизированной разработки программного обеспечения

Автоматизированная разработка программного обеспечения (CASE – Computer-aided

software engineering) - это набор инструментов («CASE-средств») и методов разработки и управления программным обеспечением, используемых программистами, разработчиками, менеджерами проектов и аналитиками, помогающих ускорить разработку программ, обеспечить их высокое качество, простоту сопровождения и минимизировать число ошибок. CASE-средства автоматизируют многие задачи (в т.ч. управленческие, административные и технические), ранее выполнявшиеся вручную.

Первыми CASE-средствами были трансляторы, компиляторы, ассемблеры, линковщики и загрузчики. Однако по мере усложнения проектов и самого программирования, росла необходимость в более комплексных инструментах. Инструменты разделились на редакторы программ, отладчики, анализаторы кода, системы контроля версий. Эти средства позволили учесть возросшие требования к проектированию, разработке и тестированию программ и проектов в целом. CASE-инструменты реализуют одну или несколько задач в рамках процесса проектирования или разработки программного обеспечения.

Используя CASE-средства, многие производители получили возможность быстрее выпустить свои продукты на рынок, поскольку процесс их создания стал более «автоматизированным». CASE-средства позволяют, условно говоря, правильно и быстро выполнять процессы проектирования и разработки программного обеспечения.

Если предоставляемая CASE-средством автоматизация охватывает весь жизненный цикл продукта, такие средства называют инструментами *комплексной автоматизированной разработки программного обеспечения* (I-CASE – Integrated CASE).

10.5. Разработка прототипов

Часто бывает необходимо создать для заказчика и разработчиков модель, в которой собраны все требования к программному продукту. Эта модель, называемая *прототипом*, может показать заказчику, что в действительности находится в голове у команды разработчиков, как они понимают установленные заказчиком требования. Это позволяет заказчику согласовать направление, в котором будет работать команда, и получить представление о конечном продукте, дает ему возможность внести изменения и дополнительно разъяснить отдельные требования, оказавшиеся неопределенными или непонятными. Кроме того, использование прототипа позволяет раньше начать тестирование в рамках процесса разработки, чтобы заранее выявить и учесть ошибки и проблемы.

Некоторые проекты являются очень большими, что может потребовать разделения продукта на отдельные части и подготовки прототипа для каждой части. В любом случае, создается ли прототип для части или для всего продукта, аналитик будет использовать сокращенное представление требований при разработке прототипа. Прототипы обычно разрабатываются с использованием специальных инструментов, которые ускоряют этот процесс. Это позволяет быстро перевести проект в более наглядную форму.

При использовании прототипа, тестирование системы безопасности можно начать на более раннем этапе. На каждом этапе разработки и с каждым прототипом может проводиться тестирование на проникновение, анализ уязвимостей, проверка форматов данных.

Если создание прототипа в виде программы оказывается непрактичным, могут быть разработаны бумажные прототипы, в которых взаимодействие, запросы, виды экранов и схемы программной логики изображены на бумаге. Бумажные прототипы также могут разрабатываться для заказчиков и/или разработчиков. На отдельных листах могут быть изображены виды экрана, а также схемы действий, которые происходят «за кадром».

10.6. Методология безопасного проектирования

Методология безопасного проектирования имеет важное значение для разработки безопасной и надежной вычислительной среды. В безопасных приложениях возможность эксплуатации уязвимостей злоумышленниками сведена к минимуму, что, в свою очередь,

снижает риски, связанные с информационными активами. Нарушения информационной безопасности могут иметь катастрофические последствия для компании, в частности, финансовые потери, разглашение коммерческой тайны, ущерб репутации и потеря доверия клиентов. Методология безопасного проектирования обеспечивает раннюю реализацию политик безопасности и безопасных методик, вместо «прикручивания» безопасности к почти завершенному проекту.

Безопасное проектирование приложений основано на знании платформ, на которых будет работать разрабатываемое программное обеспечение. Важно проанализировать подверженность этих платформ известным уязвимостям и недостаткам. Этим же уязвимостям и недостаткам, часто используемым злоумышленниками для проведения атак, будут подвержены работающие на этих платформах программы.

С помощью **анализа поверхности атак** (attack surface analytics) создается модель угроз. Техника анализа поверхности атак также предоставляет структурированный процесс анализа точек входа в программу. Эта техника акцентирует внимание на документировании всех возможных точек входа, независимо от их привилегий. Полученная при этом информация может быть использована для детализации требований к разрабатываемому программному обеспечению.

Другим аспектом, который должен быть рассмотрен на этапе проектирования, являются типы данных, которые будет обрабатывать приложение. Это поможет определить необходимые процедуры обработки параметров на входе и выходе. Очистка входящих и исходящих данных снижает возможность использования злоумышленниками специально подобранных символов для компрометации системы и получения несанкционированного доступа к ее ресурсам. Для архитектора программного обеспечения обязательным требованием является хорошее понимание контрмер, необходимых для снижения таких угроз.

10.7. Методология безопасной разработки

Для создания безопасных приложений, важное значение имеет применение методов безопасной разработки. Неправильный подход к разработке легко может поставить под угрозу весь проект.

Безопасная разработка обеспечивает управление требованиями к конечному продукту. Она постоянно акцентирует внимание на анализе разработанного кода на предмет недостатков и уязвимостей, а не откладывает этот анализ до момента завершения разработки программы. В самом деле, общепризнанным фактом является то, что большинство уязвимостей дешевле исправить на раннем этапе проектирования или разработки, а не после того, как программное обеспечение уже было внедрено у заказчика.

Важной стратегией для обеспечения безопасности процесса разработки является включение в процесс программирования регулярного проведения анализа кода. Анализ кода позволяет раньше выявлять уязвимости, архитектурные ошибки, а также возможности улучшения. Он включает в себя процедуры проверки кода, как обычными программистами, так и специалистами по безопасности. Важно, чтобы все рекомендации, данные в результате таких проверок, были задокументальны, независимо от того, были ли они фактически реализованы.

Использование автоматизированных инструментов анализа кода сокращает трудозатраты на анализ больших сегментов кода, однако важно понимать, что анализаторы кода могут только отметить недостатки кода, логические ошибки не могут быть обнаружены без непосредственного участия человека.

Поскольку анализ кода – это коллективная работа, вероятно, потребуется создать несколько экземпляров программы. При выполнении нескольких циклов анализа и изменения исходного кода, управление актуальными версиями и направление всех доработок в единый

источник становится более сложной задачей и может привести к проблемам безопасности. Внедрение централизованного хранилища кода упрощает процесс анализа. Технологии контроля версий отслеживают изменения в исходном коде и автоматически направляют все результаты разработки в единый обновленный файл. Дополнительной выгодой от использования системы управления версиями является то, что такая система позволяет осуществлять *возврат к предыдущей версии* (reversion), т.е. отмену последних изменений в файле, что позволяет программистам вернуть программу в первоначальное состояние в случае, если измененный код не работает. По сравнению с выполнением резервного копирования и отслеживания изменений исходного кода вручную, использование автоматизированной системы снижает вероятность недокументированных изменений, тайных добавлений кода, незамеченных ошибок, а также снижает другие риски безопасности, являющиеся результатом сложной структуры программы.

10.8. Проверка на защищенность

Проверка на защищенность – это методика всестороннего анализа, которая изучает поведение программ в рамках искусственно созданных сценариев атаки. Целью проверки на защищенность является анализ поведения программы при выполнении процедур тестирования на проникновение. В процессе проверки на защищенность выполняется анализ различных уязвимостей, которые могут существовать в недавно разработанных приложениях. При этом проверяется, что манипуляции с приложением не позволяют получить доступ к критичным системным процессам и ресурсам.

Проверка на защищенность позволяет оценить реализацию в программе механизмов проверки входных данных. Злоумышленники могут использовать различные подходы для внедрения в программу вредоносного кода, зависящие от типа этой программы. Проверка на защищенность также позволяет оценить устойчивость программы к попыткам переполнения буфера и проанализировать поведение программы в случае критических ошибок.

Всесторонняя проверка защищенности включает в себя проведение, как ручных, так и автоматических тестов. Автоматические тесты помогают выявить широкий спектр недостатков, которые, как правило, связаны с небрежной или ошибочной разработкой кода. Для проведения автоматизированного тестирования обычно используют программы, известные как фаззеры (fuzzer), сканеры уязвимостей и сканеры кода. Фаззеры передают на вход программы случайные или специально сформированные данные для нарушения выполнения программы. Сканеры уязвимостей проверяют наличие в программе недостатков, являющихся следствием ошибок использования типов в строго типизированных языках, ошибок разработки и конфигурирования, ошибок в транзакционных последовательностях, а также в условиях, с которыми связана работа триггеров. Автоматические тесты, в первую очередь, определяют отправные точки, которые нуждаются в дальнейшем тщательном ручном анализе.

Ручное тестирование применяется для анализа аспектов программы, которые требуют человеческой интуиции, что недоступно вычислительной технике. Также, тестировщики пытаются найти недостатки проектирования. К ним относятся логические ошибки, используя которые, злоумышленники могут управлять выполнением программы, заставляя программу выполнять определенные действия в нестандартном порядке, с целью получения более высокого уровня привилегий или обхода механизмов аутентификации. Ручное тестирование включает в себя аудит кода программистами, специализирующимися на вопросах безопасности, которые пытаются нарушить алгоритмы работы программы, используя некорректные входные данные и методы обратного инжиниринга. Ручное тестирование имитирует выполнение реальных сценариев, применяемых в реальных атаках. В некоторых случаях, ручное тестирование также предполагает использование методов социальной инженерии для анализа человеческих слабостей, которые могут привести к нарушению безопасности системы.

10.9. Управление изменениями

Изменения в процессе разработки или промышленной эксплуатации могут причинить много вреда, если они сделаны неправильно. Изменения могут происходить по нескольким различным причинам. Заказчик может изменить свои требования уже в процессе разработки, и попросить добавить, удалить или изменить отдельные функции. На этапе промышленной эксплуатации, могут потребоваться изменения, вызванные другими изменениями в среде, появлением новых требований к продукту или системе, либо новыми патчами или обновлениями. Такие изменения должны выполняться контролируемым образом, чтобы обеспечить их согласованность, правильную реализацию и отсутствие негативного влияния на другие функции. Управление изменениями – это процесс управления жизненным циклом приложения и документирования необходимых действий по контролю за изменениями.

Процесс управления изменениями должен быть внедрен в начале проекта, чтобы каждый знал, как рассматриваются изменения, что ожидается от каждого из участников, когда делается запрос на изменение. Некоторые проекты обречены с самого начала, поскольку в них изначально не был внедрен надлежащий контроль изменений. Часто при разработке программного обеспечения, заказчик и производитель сначала договариваются о функционале и других требованиях к продукту, затем заказчик должен подписать договор, подтверждающий это соглашение. Если в дальнейшем заказчик захочет произвести любые изменения, он должен будет заплатить производителю за дополнительную работу. Если в начале не был подписан договор, фиксирующий соглашение между заказчиком и производителем, заказчик может постоянно просить внести изменения и дополнения, которые потребуют от команды разработчиков тратить на эти незапланированные работы дополнительное время, в результате чего производитель теряет деньги и не может разработать продукт в срок.

Есть и другие причины для внедрения процесса управления изменениями. Эти причины связаны с организацией работы, стандартами процедурами и ожидаемыми результатами. Например, команда должна знать, что делать, если поступает запрос на изменения, когда продукт уже находится на заключительном этапе разработки. Обычно, в таких случаях, руководитель группы должен сообщить менеджеру проекта, сколько дополнительного времени потребуется для завершения проекта, если включить это изменение, и какие шаги необходимо предпринять, чтобы это изменение не повлияло на другие компоненты продукта. Кроме того, при этом не должна быть нарушена безопасность, а само изменение должно быть одобрено руководством. Если управление изменениями не осуществляется, часть команды разработчиков может реализовать такое изменение, не уведомив об этом других членов команды. Это может нарушить работу отдельных компонентов программного обеспечения, разрабатываемых другими членами команды. Когда все части продукта будут объединены и выяснится, что некоторые из них оказались несовместимы между собой, это может иметь серьезные последствия для команды разработчиков, поскольку руководство не утверждало никаких изменений.

Изменения должны быть утверждены, задокументированы и протестированы. Некоторые тесты может потребоваться провести повторно, чтобы убедиться, что изменение не влияет на другие функции продукта. Внесение программистом изменений в исходный код должно выполняться на тестовой версии системы. Ни при каких обстоятельствах программист не должен вносить изменения в код, который уже находится в промышленной эксплуатации. После внесения в код изменений, они должны быть протестированы, после чего новый код должен быть отправлен в библиотеку. В промышленную эксплуатацию код может попадать только из библиотеки, а не от программиста или из тестовой среды.

Управление изменениями должно оцениваться в ходе аудитов системы. Проблема, которая стала причиной изменений на этапе тестирования, может остаться незамеченной, поэтому в ходе аудита системы следует проанализировать процедуры управления изменениями, как

они реализованы и выполняются.

Ниже приведены некоторые шаги, которые необходимо выполнять в рамках процесса управления изменениями:

1. Оформить официальный запрос на изменение
2. Проанализировать запрос
 - а) Разработать стратегию реализации
 - б) Рассчитать стоимость реализации
 - с) Проанализировать все последствия для безопасности
3. Зарегистрировать запрос на изменение
4. Передать запрос на изменение для утверждения
5. Разработать изменение
 - а) Переписать код части продукта, добавить или исключить требуемую изменением функциональность
 - б) Связать эти изменения в коде с официальным запросом на изменение
 - с) Передать программное обеспечение для тестирования и подтверждения качества
 - д) Повторять, пока не будет обеспечено надлежащее качество
 - е) Произвести изменение версии
6. Отчитаться о результатах руководству

Произведенные в системах изменения, могут потребовать повторной сертификации и аккредитации. Если изменения значительны, вероятно, потребуется переоценить (сертифицировать) функциональность и уровень защиты, а руководство должно будет одобрить всю систему в целом, включая новые изменения (произвести аккредитацию).

10.10. Модель зрелости процессов разработки программного обеспечения

Модель зрелости процессов разработки программного обеспечения (СММ – Capability Maturity Model) описывает процедуры, принципы и приемы, которые лежат в основе зрелости процесса разработки программного обеспечения. Эта модель была разработана, чтобы помочь производителям программного обеспечения улучшить свои процессы разработки, предоставив эволюционный путь развития для достижения упорядоченных и повторяемых методов, которые повышают качество программного обеспечения, уменьшают жизненный цикл разработки, позволяют лучше управлять проектами, создавать контрольные точки и своевременно достигать их, использовать проактивный подход, вместо менее эффективного реактивного подхода.

Эта модель предоставляет политики, процедуры, руководства и лучшие практики, помогающие компании создать стандартизованный подход к разработке программного обеспечения. Целью является постоянный анализ и улучшение процессов для оптимизации результатов, увеличения возможностей и получения более качественного программного обеспечения по более низкой цене.

Модель предлагает многоуровневую структуру, которая позволяет различным компаниям осуществлять непрерывное совершенствование своих процессов. Она является инструментом для компаний-разработчиков, а также для тех, кто хочет оценить процессы и качество работы таких компаний. Например, если финансовой компании нужно выбрать подрядчика для разработки специализированного приложения, она может принять решение на основе изучения рекламных проспектов компаний-разработчиков, либо она может попросить эти компании представить результаты оценки своих процессов разработки по модели СММ. Оценкой компаний-разработчиков занимаются специализированные компании, которые выполняют сертификацию их процессов разработки программного обеспечения. Многие компании-разработчики получают такую оценку, чтобы в дальнейшем использовать ее в рекламных целях для привлечения новых клиентов.

В модели СММ используются пять уровней зрелости:

- **Начальный (Initial).** Решения в рамках процесса разработки принимаются в зависимости от обстоятельств, либо этот процесс полностью хаотичен. Компания не использует эффективные процедуры управления проектами и не занимается планированием. Результаты такого процесса разработки непредсказуемы и полностью зависят от способностей отдельных сотрудников, качество не гарантируется.
- **Повторяемый (Repeatable).** Созданы и задокументированы формальные процедуры управления проектами, управления изменениями и обеспечения качества. Компания может повторять накопленный ранее успешный опыт в каждом проекте. Результаты более предсказуемы, но все еще зависят от способностей отдельных сотрудников. У компании нет определенной формальной модели процессов.
- **Определенный (Defined).** Описаны и внедрены формальные процедуры, которые объединены в общий процесс разработки программного обеспечения, повторяемый в каждом проекте. Появляется возможность точной оценки сроков и себестоимости реализации продукта. Результат предсказуем и не зависит от способностей отдельных сотрудников. Компания идет по пути постоянного совершенствования процессов.
- **Управляемый (Managed).** Описаны и внедрены формальные процессы сбора и анализа количественных показателей, в том числе, показателей качества. Определены и применяются метрики, позволяющие измерить эффективность процессов разработки. Метрики учитываются в программе совершенствования процессов. Осуществляется более точное планирование. Результат полностью предсказуем.
- **Оптимизированный (Optimizing).** Компания выделяет бюджет и планирует непрерывное совершенствование процессов. Степень улучшения процессов и их эффективность может оцениваться количественно.

10.11. Передача исходного кода программного обеспечения на хранение независимой третьей стороне

Если компания-заказчик платит компании-разработчику за разработку программного обеспечения для нее, ей следует использовать один из вариантов *передачи исходного кода этого программного обеспечения на хранение независимой третьей стороне* (software escrow) для защиты своих интересов. При этом третья сторона хранит копию исходного кода и, возможно, другие материалы, которые она должна передать заказчику только при наступлении конкретных, заранее оговоренных, обстоятельств. В основном такими обстоятельствами является уход с рынка компании-разработчика, разработавшей программное обеспечение, или невыполнение ей своих обязательств по какой-либо иной причине. Передача исходного кода третьей стороне, защищает компанию-заказчика, поскольку она оплачивает разработку программного обеспечения, и если вдруг разработчик перестает выполнять свои обязательства по сопровождению и доработке продукта, заказчик теряет возможность эффективного использования продукта и не имеет доступа к исходному коду, чтобы взять функции сопровождения и доработки на себя или передать их другой компании.

Возникает логичный вопрос – почему сам разработчик не передает исходный код заказчику, ведь он заплатил за его разработку? Обычно этот процесс так не работает. Исходный код разработанного программного продукта является интеллектуальной собственностью компании-разработчика. Разработчик нанимает и платит заработную плату программистам, имеющим необходимые навыки, за разработку этого продукта. Если разработчик просто передаст исходные коды заказчику вместе с самим продуктом, это будет являться фактом передачи своей интеллектуальной собственности, своих секретов. Вместо этого клиенту передается скомпилированный код, но не исходный код. Скомпилированный код – это результат обработки исходного кода компилятором, который переводит его в форму, необходимую для выполнения компьютером, но нечитаемую для человека. Большая часть

прибыли от продажи программного обеспечения основана на лицензировании, при этом лицензия ограничивает возможные действия покупателя программного обеспечения с его скомпилированным кодом.

11. Методология разработки программного обеспечения

Приложения пишутся на языке программирования, команды которого указывают операционной системе и процессору, какие действия должны произойти, чтобы выполнить требования пользователя приложения. Сменилось несколько поколений языков программирования, каждое следующее поколение было основано на предыдущем, расширяя его функциональные возможности и предоставляя программистам все более мощные инструменты.

Для разработки приложений могут использоваться различные типы языков программирования: машинный язык, язык ассемблера, либо языки высокого уровня. Машинный язык – это тот язык, который понимает процессор, и с которым он может работать напрямую. Ассемблер и языки высокого уровня непосредственно процессору не понятны, поэтому они должны быть транслированы в машинный язык. Этот процесс, как правило, выполняется компилятором, задачей которого является перевод понятного человеку языка программирования в понятный компьютеру машинный язык (или объектный код).

Исходный код и Машинный код. При обработке компилятором исходного кода программы, на выходе получается объектный код, предназначенный для конкретной платформы и процессора. Этот объектный код является исполняемой формой приложения, которую пользователь покупает у поставщика. Запущенный объектный код представляется в виде машинного кода, понятного процессору.

Поколения языков программирования. Языки программирования развивались на протяжении длительного времени, предоставляя программистам и системам все более широкую функциональность. Поколения языков программирования перечислены ниже:

- Первое поколение: Машинный язык
- Второе поколение: Язык ассемблера
- Третье поколение: Язык высокого уровня
- Четвертое поколение: Язык очень высокого уровня
- Пятое поколение: Естественный язык

ПРИМЕЧАНИЕ. Если вам потребуется больше узнать о различных поколениях языков программирования, ознакомьтесь со статьей «Языки программирования» по ссылке: www.logicalsecurity.com/resources/resources_articles.html.

Обычно покупатель приобретает программное обеспечение в виде объектного кода. Это программное обеспечение уже скомпилировано и готово к установке и работе на системе покупателя. Для преобразования программы в объектный код, понятный процессорам определенного типа, производитель программного обеспечения использует компилятор. Программа, которая работает на компьютере с процессором Alpha, может не работать на компьютере с процессором Pentium, поскольку различные типы процессоров требуют различные варианты машинного языка для исполнения программ.

Если программа продается в виде исходного кода, покупатель, чтобы запустить ее на своем компьютере, должен будет самостоятельно обработать ее с помощью соответствующего компилятора. Коммерческие программы редко распространяются в виде оригинального исходного кода, поскольку это может позволить конкурирующим производителям извлечь из него оригинальные идеи и методы, примененные разработчиками. Исходный код считается интеллектуальной собственностью производителя, разработавшего программное обеспечение, поэтому он должен быть хорошо защищен.

Для перевода программы, написанной на языке высокого уровня (исходного кода), в объектный или машинный код, используются различные программы. Такими программами

являются интерпретаторы, компиляторы и ассемблеры. **Интерпретаторы** последовательно переводят команды программы в машинный код непосредственно в процессе выполнения программы, а **компиляторы** заранее транслируют в машинный код большие фрагменты исходного кода. **Ассемблеры** переводят язык ассемблера в машинный язык. Большинство приложений распространяются в скомпилированном виде, в то время как программы, написанные на скриптовых языках – интерпретируются.

11.1. Концепции объектно-ориентированного программирования

Раньше для разработки программного обеспечения использовались классические методы: ввод-обработка-вывод. Использовалась модель информационных потоков, содержащих иерархические структуры информации. Данные поступали на вход программы, программа обрабатывала данные от начала до конца, выполняла логические процедуры и возвращала результат.

Объектно-ориентированное программирование (ООП) позволяет выполнять те же функции, но по-другому, с использованием более эффективных методов. Вам нужно понимать основные концепции ООП.

ООП использует классы и объекты. Объект реального мира, например, «стол», является членом (или экземпляром) более широкого класса объектов, например, «мебель». Класс «мебель» имеет набор связанных с ним атрибутов (например, цвет, размер, вес, стиль, стоимость), которые при создании наследует объект. Эти атрибуты будут использоваться при *создании экземпляра* объекта класса «мебель» – например, «стол» или «стул». Объект «стол», являющийся членом класса «мебель», наследует все атрибуты, определенные для данного класса (рис. 9-12).

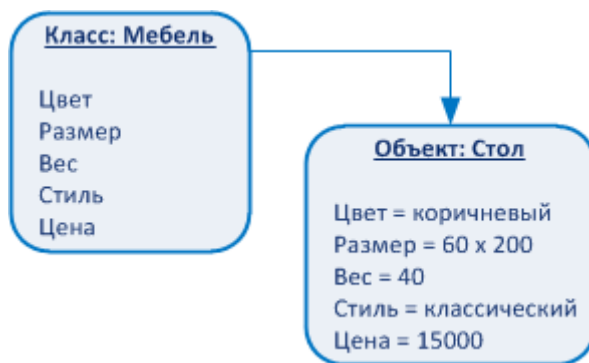


Рисунок 9-12. В объектно-ориентированном программировании при наследовании каждый объект относится к определенному классу и обладает всеми атрибутами этого класса

Программист разрабатывает класс и указывает в нем все его характеристики и атрибуты. Самым изящным в этом подходе является то, что программисту не нужно разрабатывать каждый объект. В качестве аналогии, рассмотрим кофеварку. Один клиент с помощью интерфейса кофеварки выбирает приготовление кофе латте. Другому клиенту нужен капучино, который варится совсем по-другому. Ваша кофеварка может удовлетворить потребности обоих, и приготовить для каждого клиента тот вариант кофе, который ему нравится. При этом само устройство для обоих случаев используется одно и то же, но его интерфейс позволяет ему принимать различные запросы и создавать на основании них различные результаты.

Но каким образом класс создает объекты на основе запросов? Часть программного обеспечения, написанного на объектно-ориентированном языке, принимает запросы, отправляемые, как правило, другим объектом. Запрашивающий объект хочет, чтобы новый объект выполнял определенные функции. Скажем, объект «А» хочет, чтобы объект «В» выполнил вычитание чисел, отправленных от «А» к «В». При поступлении запроса, создается объект (экземпляр), имеющий весь необходимый программный код. Объект «В»

выполняет задачу вычитания и отправляет результат обратно объекту «А». Не имеет значения, на каком языке программирования написаны эти два объекта, важно, что они знают, как общаться друг с другом. Один объект может взаимодействовать с другим объектом с помощью интерфейса прикладного программирования (API) этого объекта. API является механизмом, позволяющим объектам общаться друг с другом. В качестве аналогии можно рассмотреть иностранные языки. Если два человека хотят общаться, они должны говорить на одном языке, например, на русском. Если один из них знает русский плохо (например, знает всего несколько фраз), набор тем для общения будет сильно ограничен. Но если эти люди говорят на разных языках, пообщаться им не удастся.

ПРИМЕЧАНИЕ. Объект – это предварительно подготовленный код, включенный в модуль.

Так что же такого замечательного в ООП, чем методики ООП отличаются от методик "не-ООП"? "Не-ООП" приложения – монолитны. Приложение является просто большой кучей кода. Если требуется что-то изменить в этой куче, нужно будет пройти через все функции программной логики, чтобы понять, к чему может привести это изменение. Если программа содержит сотни или тысячи строк кода, это нельзя назвать простой и приятной задачей. Если же вы решили написать свою программу на объектно-ориентированном языке, у вас не будет одного монолитного приложения, ваше приложение будет состоять из небольших компонентов (объектов). Если нужно будет внести некоторые изменения или обновления в отдельные функции такого приложения, достаточно будет просто изменить код в рамках класса, отвечающего за создание объектов, которые выполняют соответствующую функцию, и не беспокоиться об остальных функциях программы. Ниже перечислены основные преимущества ООП:

- Модульность
 - Автономные объекты, взаимодействующие посредством обмена сообщениями
- Отложенная реализация
 - Внутренние компоненты объекта могут быть переопределены без изменения других частей системы
- Возможность повторного использования
 - Детализация (усовершенствование) классов посредством наследования
 - Использование тех же самых объектов другими программами
- Естественность
 - Объектно-ориентированный анализ, проектирование и моделирование непосредственно связаны с потребностями и решениями бизнеса

Большинство приложений имеют ряд одинаковых функций. Вместо того чтобы разрабатывать один и тот же код, для использования в десяти различных приложениях, использование ООП позволяет один раз создать объект с соответствующей функциональностью и затем использовать его в других приложениях. Это уменьшает время разработки и экономит деньги.

Теперь, когда мы разобрались с основными принципами, давайте рассмотрим используемую терминологию. **Метод** – это функция или процедура, которую может выполнять объект. Объект может быть создан, чтобы, к примеру, принимать команды от пользователя и формировать на основе них запросы для отправки определенному серверному приложению. Другой объект может выполнять метод, который извлекает данные из базы данных и вводит их в веб-формы.

Объекты инкапсулируют значения атрибутов, т.е. эта информация упакована под одним именем и может быть использована как единое целое другими объектами. Объекты должны

иметь возможность взаимодействовать друг с другом, и это реализуется с помощью **сообщений**, отправляемых интерфейсу прикладного программирования (API) принимающего объекта. Если объект «А» должен сообщить объекту «В», что остаток на расчетном счете клиента нужно уменьшить на 100 рублей, он посылает объекту «В» соответствующее сообщение. Это сообщение состоит из ссылки на объект-получатель, ссылки на метод, который должен быть выполнен, а также соответствующих аргументов.

У объекта могут быть общедоступные (shared) и скрытые (private) части. Общедоступные части объекта – это интерфейс (API), который позволяет ему взаимодействовать с другими компонентами. Сообщения поступают через интерфейс, указывая объекту, какую нужно выполнить операцию (метод). Реально работу (запрошенные операции) выполняют скрытые части объекта. Другим компонентам не нужно знать, как устроен каждый объект изнутри, им достаточно знать, какие операции он выполняет. Таким образом осуществляется **скрытие информации**. Детальная информация о выполнении операций скрыта от всех элементов программы, находящихся вне объекта. Объекты взаимодействуют через четко определенные интерфейсы, поэтому им не нужно знать, каким образом работают другие объекты.

ПРИМЕЧАНИЕ. Скрытие данных обеспечивает инкапсуляцию, которая защищает приватные данные объекта от доступа извне. Нет необходимости в предоставлении полного доступа одних объектов к внутренним данным или процессам другого объекта.

Использование методов объектно-ориентированного программирования позволяет аналитикам и разработчикам взглянуть на более высокий уровень операций и процедур, не просматривая каждый отдельный объект и его код. Такая модульность обеспечивает более наглядную и понятную модель.

Абстракция (abstraction) – это возможность опустить ненужные и неважные детали, которые затем будут реализованы в дочерних классах. Это дает возможность выделения концептуальных аспектов системы. Например, если архитектору программного обеспечения нужно понять, как организованы потоки данных в программе, ему нужно будет разобраться с глобальными, высокоуровневыми частями программы и отследить, какие шаги проходят данные от момента их поступления в программу до момента выхода из нее в виде результатов. Ему было бы трудно понять концепции программы, если бы ему потребовалось анализировать программу на уровне мелких деталей. С помощью абстракции, ненужные детали исключаются, что позволяет работать только с важными частями программы.

Обмен сообщениями может осуществляться несколькими способами. Два объекта могут иметь единственное соединение (один к одному), множественные соединения (один ко многим), а также обязательные и опциональные соединения. Важно оформить схему таких связей, чтобы найти неприемлемые маршруты, по которым может передаваться информация. Это поможет обеспечить невозможность передачи критичных данных объектам на более низком уровне безопасности.

У каждого объекта должна быть спецификация, которой он должен придерживаться. Такой порядок обеспечивает более чистое программирование, уменьшает количество программных ошибок и недостатков. Приведенный ниже список является примером того, что должно быть разработано для каждого объекта:

- Название объекта
- Описания атрибутов
- Имена атрибутов
- Содержание атрибутов
- Типы данных атрибутов
- Внешний ввод данных в объект

- Вывод данных из объекта вовне
- Описания операций
- Названия операций
- Описания функциональных интерфейсов
- Описания работы операций
- Вопросы производительности
- Ограничения и лимиты
- Связь между экземплярами
- Связь посредством сообщений

Разработчик создает класс, который описывает эту спецификацию. При создании экземпляров объектов, они наследуют эти атрибуты.

Каждый объект можно использовать повторно, как уже было сказано. Это позволяет более эффективно использовать ресурсы и время программиста. Различные приложения могут использовать одни и те же объекты, что снижает объем лишней работы, а также повышает функциональность приложения. Объекты могут быть легко добавлены и интегрированы в структуру приложения.

Объекты могут быть включены в библиотеку, что обеспечивает более экономичный способ их использования несколькими приложениями (см. Рисунок 9-13). Библиотека предоставляет индексы и указатели на объекты, находящиеся в той же или другой системе.

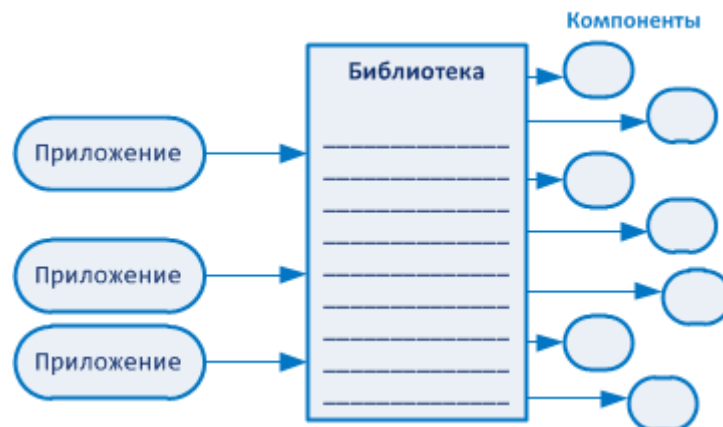
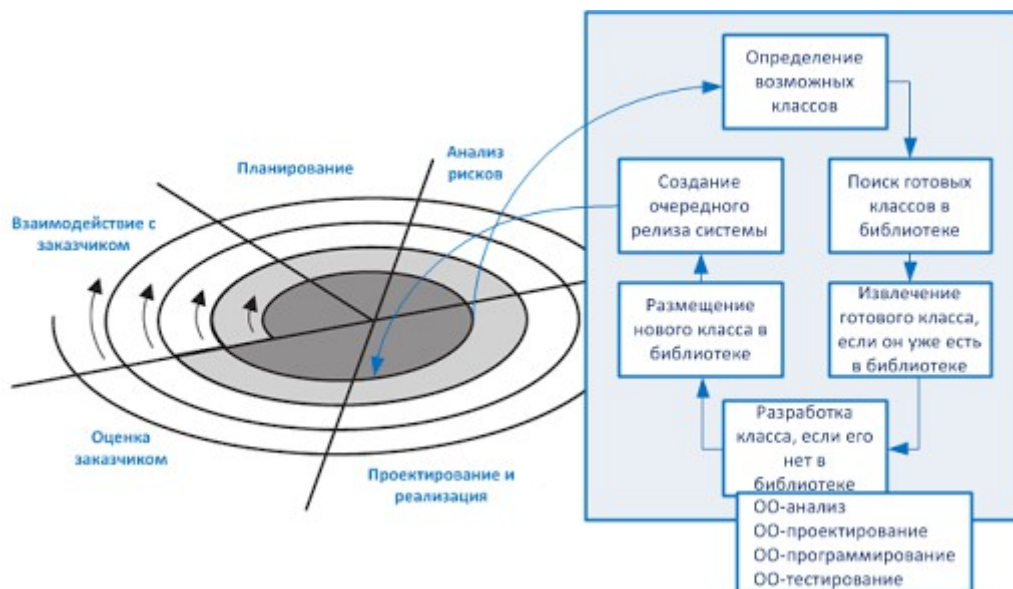


Рисунок 9-13. Приложения находят нужные им объекты в библиотеки с помощью индексов

Если приложение разработано на основе модульного подхода (например, с использованием методов ООП), появляется возможность повторного использования его компонентов, уменьшается сложность, могут применяться методы параллельной разработки. Эти возможности позволяют совершать меньше ошибок, упрощают процесс внесения изменений, повышают эффективность использования ресурсов, позволяют соблюдать сроки разработки кода лучше, чем при использовании классических подходов. Кроме того, ООП обеспечивает функциональную независимость – каждый модуль реализует только отдельные требуемые функции и имеет интерфейс, понятный другим частям приложения.

Объект инкапсулирован, т.е. структура данных (функциональность операций) и применимые способы доступа к ней объединены в одно целое. Другие объекты, субъекты и приложения могут использовать этот объект и его функции, обращаясь к нему через контролируемые и стандартизированные интерфейсы и отправляя ему сообщения.



Полиморфизм

Полиморфизм – это слово из греческого языка, которое означает «наличие нескольких форм». Понимание этой концепции обычно вызывает сложности, поэтому давайте рассмотрим ее на примере. Если я разрабатываю программу на объектно-ориентированном языке, я могу создать переменную, которая может использоваться для хранения данных различных типов. Приложение будет определять, какой тип данных использовать, непосредственно во время выполнения программы (run time). К примеру, если имя моей переменной USERID и я разрабатываю объект, в котором предусмотрено, что эта переменная может хранить либо целые числа, либо строки, это обеспечивает дополнительную гибкость. С помощью такой переменной, идентификатор пользователя может быть принят в виде числа (номер записи) или имени (строки символов). При этом приложение «А», использующее этот объект, может работать с идентификаторами в виде целых чисел, а приложение «В», использующее тот же объект, может работать с идентификаторами, представленными в виде строки символов.

Полиморфизм - это возможность объектов с одинаковой спецификацией иметь различную реализацию и выдавать различные результаты при получении одинаковых данных. Реализация методов объекта может быть изменена, например, в процессе наследования.

Объектно-ориентированный анализ (OOA – object-oriented analysis) – это процесс определения классов объектов, которые будут подходить для решения. Проводится анализ проблемы для определения классов объектов, которые нужно будет использовать в приложении.

Объектно-ориентированное проектирование (OOD – object-oriented design) создает представление проблемы реального мира и связывает ее с программным решением, используя ООП. Результатом объектно-ориентированного проектирования является проект, который разделяет на модули данные и процедуры. Проект связывает объекты данных и операции обработки.

11.2. Моделирование данных

В предыдущих разделах был в упрощенном виде рассмотрен **подход структурированного анализа**. Подход структурированного анализа рассматривает все объекты и субъекты приложения и устанавливает взаимосвязи, коммуникационные маршруты и наследуемые свойства. Это отличается от **моделирования данных**, при котором рассматриваются данные, независимо от способа их обработки и компонентов, обрабатывающих данные. Модели данных отслеживают прохождение данных от начала до конца и проверяют корректность данных на выходе. Объектно-ориентированный анализ является примером подхода структурированного анализа. Если аналитик проводит объектно-ориентированный анализ приложения, он должен убедиться, что все отношения правильно установлены, цепочки

наследования предсказуемы и удобны, что экземпляры объектов практичны и предоставляют необходимую функциональность, что атрибуты каждого класса охватывают все необходимые значения, используемые приложением. Если другой аналитик проводит анализ модели данных того же приложения, он будет отслеживать данные и возвращаемые после их обработки значения. Приложение может иметь прекрасную ООА-структуру, но если оно получает задание посчитать $1 + 1$ и возвращает результат 3, видимо что-то в нем все-таки неправильно. Это именно то, что рассматривает моделирование данных.

Другой пример моделирования данных связан с базами данных. Моделирование данных может быть использовано, чтобы понять суть данных и отношений, которые управляют ими. Элемент данных в одном хранилище данных, может быть указателем на другое хранилище данных. Такие указатели реально должны указывать на нужное место. Моделирование данных проверяет это, а структурированный анализ ООА – нет.

11.3. Архитектура программного обеспечения

Архитектура программного обеспечения относится к компонентам, из которых состоит программное решение, позволяющее решать реальные задачи. Архитекторы программного обеспечения рассматривают приложение на более высоком уровне, чем программисты, которые сосредотачивают свое внимание на структурах данных, правилах разработки программ, переменных и связях между объектами. С точки зрения архитектуры видно, как именно приложение в действительности учитывает и выполняет требования, определенные и согласованные на этапе проектирования.

При создании архитектуры программного обеспечения, требования к нему разделяются на отдельные части, которые могут быть решены отдельными программными решениями. Процесс такого разделения является промежуточным этапом между анализом требований к программному обеспечению и разработкой конкретных компонентов, из которых будет состоять приложение.

Если от приложения требуется, чтобы оно выполняло сканирование жестких дисков и электронной почты на наличие вирусов, архитектура программного обеспечения разобьет это требование на отдельные блоки (модули), которые должны будут реализовать функции этого приложения. Среди таких блоков будут следующие функциональные модули:

- Хранилище вирусных сигнатур
- Агент, который выполняет сравнение содержимого файлов с сигнатурами вирусов
- Процедуры анализа содержимого сообщений электронной почты
- Процедуры извлечения файлов из архивов
- Процедуры на случай обнаружения вируса
- Процедуры на случай выявления зашифрованного вложения электронной почты

Такой способ разработки программного продукта обеспечивает лучшую управляемость и модульный подход к задачам и решениям. Если вы дадите группе программистов задание разработать антивирусную программу, они не будут знать, что каждому из них нужно делать конкретно. Однако если одному из них вы дадите указание написать фрагмент программы, который будет хранить и обновлять файлы сигнатур, другому – разработать компонент для сравнения файлов с сигнатурами, а третьему – задание по созданию модуля для работы с заархивированными файлами, все программисты получат конкретные задачи, четкие цели и разойдутся по своим рабочим местам для выполнения осмысленной работы.

Архитекторы программного обеспечения должны обеспечить видение задач проекта и его целей.

11.4. Структуры данных

Структура данных – это представление логических связей между элементами данных. Она указывает степень взаимосвязи элементов, методы доступа, а также организацию элементов данных.

Структура может быть простой, как, например, скалярное значение, которое представляет единственный элемент, вызываемый по идентификатору и хранящийся в одной ячейке памяти. Скалярные элементы могут быть сгруппированы в массивы, доступ к которым выполняется по индексам. Другими структурами данных могут являться иерархические структуры, представленные в виде списков со множественными связями, и содержащие скалярные элементы, векторы, и, возможно, массивы. Иерархическая структура обеспечивает категоризацию и взаимозависимость между элементами. На Рисунке 9-14 показаны простые и более сложные структуры данных.

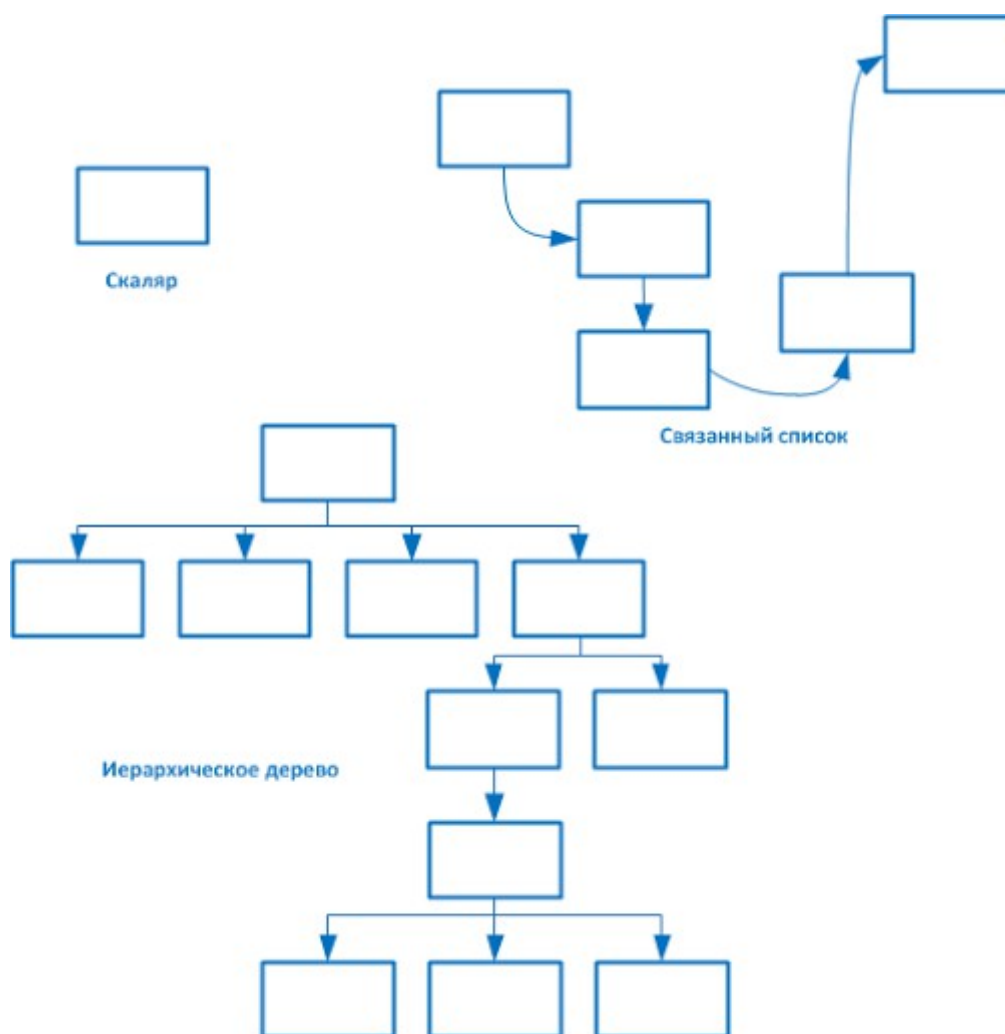


Рисунок 9-14. Структуры данных по своей природе могут быть очень простыми и очень сложными

11.5. Связность и связанность

Связность (cohesion) отражает, сколько различных типов задач может выполнять модуль. Если модуль выполняет только одну задачу (вычитание значений) или несколько похожих задач (вычитание, сложение и умножение) он считается имеющим высокую связность, что весьма хорошо. Чем выше связность модуля, тем легче его обновлять или вносить в него изменения, не затрагивая при этом другие модули, взаимодействующие с ним. Также это упрощает повторное использование модуля и его сопровождение, поскольку он более прост по сравнению с модулем, имеющим низкую связность. Модуль с низкой связностью

выполняет несколько различных задач, что увеличивает сложность модуля и делает непростой задачей его дальнейшее обслуживание и повторное использование.

Связанность (coupling) – это показатель, который отображает, как много взаимодействий требуется одному модулю для выполнения своих задач. Если модуль имеет низкую связанность, это означает, что он не нуждается во взаимодействии с множеством других модулей, для выполнения своей работы. Высокая связанность означает, что модуль зависит от множества других модулей, необходимых ему для выполнения своих задач. Низкая связанность более предпочтительна, поскольку такие модули легче понять, проще повторно использовать, а также проще изменять без воздействия на окружающие модули. Низкая связанность говорит о том, что программисту удалось создать хорошо структурированный модуль. Например, если одному сотруднику для выполнения одной своей задачи требуется взаимодействовать с пятью другими людьми, возникает слишком много сложностей, что отнимает слишком много времени, и дает больше возможностей для ошибок.

Примером низкой связанности может быть передача одним модулем значения переменной в другой модуль. Если модуль «А» передает одно значение модулю «В», другое – модулю «С», и еще одно – модулю «D», это будет являться примером высокой связанности. При этом модуль «А» не может выполнять свои задачи пока не получит результаты от модулей «В», «С» и «D».

ПРИМЕЧАНИЕ. Модули следует разрабатывать самодостаточными и выполняющими единственную логическую функцию, что будет являться высокой связанностью. Модули не должны сильно зависеть друг от друга, что будет являться низкой связанностью.

12. Распределенные вычисления

Многие современные приложения используют клиент-серверную модель, при которой меньшая часть приложения (клиент) может работать на множестве различных систем, а большая часть (сервер) – запускается на сервере. Серверная часть имеет значительно более широкую функциональность и работает на более мощной платформе, по сравнению с клиентской частью. Клиенты направляют запросы серверной части, а сервер возвращает им уже готовые результаты. Выглядит достаточно просто. Но каким образом организуется взаимодействие клиентских частей с серверной?

Тремя основными вариантами архитектуры взаимодействия клиентских и серверной частей являются: CORBA, EJB и Microsoft COM. Они будут рассмотрены далее в этом разделе.

Распределенная модель вычислений требует регистрации клиентских и серверных компонентов, которым нужно найти друг друга в сети по именам или идентификаторам, выяснить, какую функциональность выполняют различные компоненты. Первой задачей клиентской части обычно является поиск нужных компонентов и определение их функций. Это необходимо для организации управляемого и контролируемого взаимодействия между компонентами, обеспечения возможности передачи запросов и результатов правильным компонентам.

Жизнь была бы намного проще, если бы у нас была только одна архитектура межкомпонентного взаимодействия, которую использовали бы все разработчики. Но в действительности существует множество различных архитектур, предназначенных для работы в различных средах программирования. Тем не менее, все они так или иначе выполняют главную функцию – обеспечивают возможность взаимодействия компонентов клиентской и серверной сторон друг с другом.

12.1. CORBA и ORB

Чтобы компоненты могли взаимодействовать, должны использоваться стандартизованные интерфейсы и коммуникационные механизмы. Это единственный способ обеспечить возможность взаимодействия.

CORBA (Common Object Request Broker Architecture - Общая архитектура брокера объектных запросов) – это открытый объектно-ориентированный стандарт архитектуры, разработанный Object Management Group (OMG). CORBA позволяет организовать взаимодействие между собой огромного количества программного обеспечения, платформ и оборудования в современных средах. CORBA позволяет приложениям взаимодействовать друг с другом, независимо от того, где они находятся и кто их разработал.

Этот стандарт определяет API, коммуникационный протокол и методы взаимодействия клиентов и сервера, позволяющие работать совместно различным приложениям, написанным на разных языках программирования и работающих на различных платформах. Звучит прекрасно.

Разработанная OMG модель CORBA позволяет использовать в среде различные сервисы. Модель определяет семантику объектов, что обеспечивает стандартизацию видимых извне характеристик, которые одинаково рассматриваются всеми остальными объектами среды. Такая стандартизация позволяет различным разработчикам создавать сотни и тысячи компонентов, которые могут взаимодействовать с другими компонентами в среде, не зная при этом, как эти компоненты в действительности работают. Разработчики знают, как взаимодействовать с этими компонентами, поскольку они используют стандартные интерфейсы и следуют правилам модели CORBA.

В этой модели, клиенты запрашивают сервисы у объектов. Клиент передает объекту сообщение, которое содержит имя объекта, запрошенную операцию и другие необходимые для выполнения этой операции параметры.

Модель CORBA предоставляет стандарты для построения законченной распределенной среды. Она состоит из двух основных частей: компонентов, ориентированных на систему (**брокеры объектных запросов** (ORB – Object Request Broker) и сервисов объектов), и компонентов, ориентированных на приложения (объекты приложений и общие возможности). ORB управляет всеми коммуникациями между компонентами и позволяет им взаимодействовать в гетерогенной и распределенной среде, как показано на Рисунке 9-15. ORB работает независимо от платформы, на которой находятся объекты, что обеспечивает прекрасную совместимость.

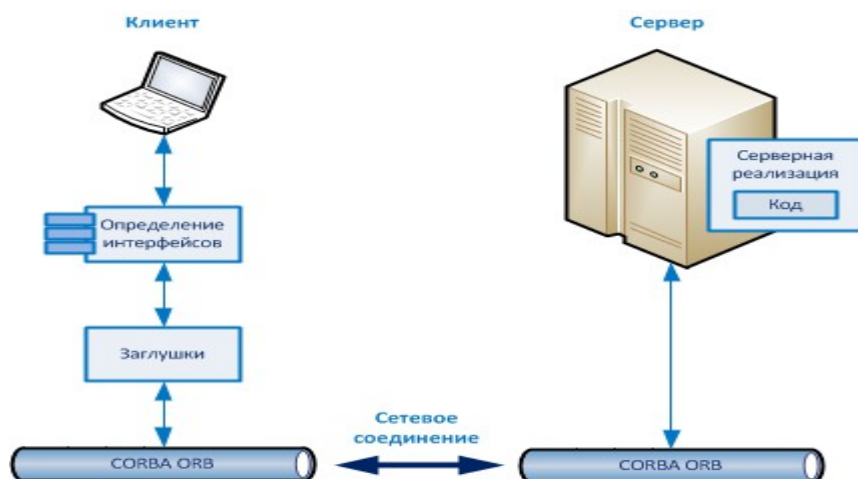


Рисунок 9-15. ORB позволяет взаимодействовать и совместно работать различным компонентам через сетевое соединение

ORB – это промежуточное программное обеспечение (middleware), которое устанавливает отношения клиент-сервер между объектами. Если клиенту нужно получить доступ к объекту на сервере для выполнения этим объектом операции или метода, ORB перехватывает его запрос и выполняет поиск нужного объекта. Если объект будет найден, ORB вызывает нужный метод (или операцию), передает ему параметры, а затем возвращает результаты клиенту. Клиентскому программному обеспечению не нужно знать, где находится объект и

как найти его. Это задача ORB. При этом объекты могут быть написаны на разных языках и работать на разных операционных системах и платформах, но клиенту не нужно беспокоиться об этом.

При взаимодействии объектов друг с другом, они используют *каналы* (pipe), которые являются сервисами межкомпонентной связи. Доступны различные типы каналов, например, удаленный вызов процедур (RPC – Remote Procedure Call) и ORB. ORB обеспечивают связь между распределенными объектами. Если объекту на рабочей станции нужен объект, находящийся на сервере обработки данных, он может сделать запрос через ORB. При этом ORB выполнит поиск необходимого объекта и обеспечит коммуникационный маршрут между двумя этими объектами на время, пока не завершится процесс их взаимодействия. Это клиент-серверные коммуникационные каналы, используемые во многих сетевых средах.

ORB является механизмом, который позволяет объектам взаимодействовать локально или удаленно. Он позволяет одним объектам делать запросы к другим объектам и получать ответы. Для клиента все это происходит прозрачно, ему предоставляется канал для связи со всеми нужными ему объектами. Использование CORBA позволяет приложению предоставлять доступ к своим объектам различным типам ORB. Это обеспечивает переносимость приложений и решает многие из проблем совместимости, с которыми могут столкнуться разработчики и поставщики, продукция которых работает в различных средах.

12.2. COM и DCOM

COM (Component Object Model - Объектная модель компонентов) представляет собой модель, которая обеспечивает межпроцессное взаимодействие в рамках одного приложения или между приложениями на одной компьютерной системе. Эта модель была создана Microsoft, она описывает стандартизированные API, схемы именования компонентов и коммуникационные стандарты. Если разработчику нужно, чтобы его приложение могло взаимодействовать с операционной системой Windows и различными работающими в ней приложениями, ему следует использовать стандарт COM.

Figure 11-21 CORBA provides standard interface definitions, which offer greater interoperability in heterogeneous environments.

DCOM (Distributed COM - Распределенная COM) поддерживает такую же модель взаимодействия компонентов, добавляя к ней возможность распределенного взаимодействия процессов (IPC – Interprocess Communications). COM позволяет приложениям использовать компоненты на той же системе, тогда как DCOM позволяет приложениям использовать объекты, находящиеся в различных местах в сети. Таким образом, DCOM позволяет выполнять клиент-серверное взаимодействие, в поддерживающей COM операционной системе или приложении.

Без DCOM программистам пришлось бы писать гораздо более сложный код, чтобы найти в сети нужные объекты, настроить сетевые сокет и реализовать необходимые для взаимодействия сервисы. DCOM заботится обо всех этих вопросах (и ряде других), позволяя программисту сосредоточиться на своих задачах в части разработки необходимой функциональности своего приложения. У DCOM есть библиотека, которая выполняет управление сеансами взаимодействия, синхронизацией, буферизацией, выявлением и обработкой ошибок, переводом форматов данных.

DCOM работает в качестве промежуточного программного обеспечения, реализуя распределенную обработку и предоставляя разработчикам сервисы, обеспечивающие возможность взаимодействия процессов в масштабе сети (см. Рисунок 9-16).

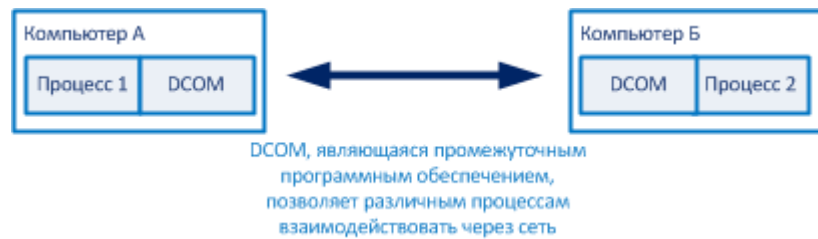


Рисунок 9-16. Хотя DCOM предоставляет коммуникационный механизм для распределенной среды, он продолжает работать на основе архитектуры COM

Другими видами промежуточного программного обеспечения, предоставляющего аналогичные функциональные возможности, являются: ORB, MOM (message-oriented middleware - промежуточное программное обеспечение, ориентированное на обмен сообщениями), RPC, ODBC и т.п. DCOM предоставляет ORB-подобные сервисы, сервисы передачи данных, распределенные службы обмена сообщениями и распределенные службы транзакций, являющиеся надстройками над его механизмом RPC. DCOM объединяет все эти функции в одну технологию, которая использует тот же интерфейс, что и COM.

SOAP

Что делать, если вам нужно, чтобы программы, работающие на различных операционных системах, взаимодействовали посредством коммуникационных механизмов, основанных на веб-технологиях? Вам следует использовать SOAP (Simple Object Access Protocol - Простой протокол доступа к объектам). SOAP представляет собой протокол, основанный на XML, который кодирует сообщения в форму, понятную веб-сервисам. Если вы работаете на компьютере с операционной системой Windows 2000 и вам нужно получить доступ к серверу, работающему под управлением Windows 2008, который предоставляет некий веб-сервис, программы на вашей системе и сервере могут использовать для взаимодействия протокол SOAP, не углубляясь в вопросы совместимости. Такой вариант взаимодействия чаще всего осуществляется по протоколу HTTP, поскольку работает на подавляющем большинстве современных компьютеров.

Когда компьютер с Windows 2000 делает запрос к серверу с Windows 2008 для получения некоего сервиса, SOAP кодирует запрос таким образом, чтобы другие программы могли понять этот запрос и предоставить запрошенный сервис.

SOAP определяет XML-схему или структуру, описывающие порядок осуществления коммуникаций. XML-схема SOAP определяет порядок взаимодействия объектов напрямую. Одним из преимуществ SOAP является то, что программы могут взаимодействовать через межсетевые экраны, т.к. их правилами взаимодействие по протоколу HTTP обычно разрешено. Это помогает обеспечить работоспособность модели клиент-сервер в том случае, если клиенты и сервер находятся по разные стороны межсетевого экрана.

12.3. EJB

EJB (Enterprise JavaBeans) – это спецификация для разработки распределенных приложений, написанных на Java. EJB предоставляет интерфейсы и методы, позволяющие различным приложениям взаимодействовать в сетевой среде. При использовании протокола IIOP (Internet Inter-ORB Protocol), клиентской частью не обязательно должна быть программа, написанная на Java, это может быть любой корректно работающий клиент CORBA.

ПРИМЕЧАНИЕ. Компонент Java называется Java Bean.

J2EE (Java Platform Enterprise Edition) имеет несколько API, EJB является лишь одним из них. EJB используется для инкапсуляции бизнес-логики приложения в его серверной части (в модели клиент-сервер). Аналогично моделям COM и CORBA, которые были созданы для обеспечения модульного подхода к программированию с целью обеспечения совместимости, EJB определяет модель клиент-сервер, являющуюся объектно-ориентированной и независимой от платформы. Основной целью является получение стандартизированного метода реализации кода серверного программного обеспечения, реализующего бизнес-логику корпоративных приложений.

12.4. OLE

OLE (Object linking and embedding) - это технология связывания и внедрения объектов, основанная на COM, которая предоставляет возможность совместного использования объектов на локальном компьютере. OLE позволяет встраивать в документы объекты, такие как графические изображения, таблицы и т.п.

ПРИМЕЧАНИЕ. Возможность вызова одной программой другой программы называется связыванием (linking). Возможность поместить часть данных внутрь другой программы или документа называется встраиванием (embedding).

OLE позволяет также связывать различные объекты и документы. Например, когда вы редактируете документ в Microsoft Word и вставляете в него ссылку (URL), эта ссылка автоматически выделяется синим цветом и подчеркивается, что говорит пользователю, который будет читать ваш документ, что он может щелкнуть по этой ссылке, чтобы открыть соответствующий веб-сайт. Это является примером связывания. Если вы добавляете таблицу Excel в документ Word, экземпляр этой таблицы встраивается в ваш документ. Чтобы отредактировать эту таблицу непосредственно из Word, нужно дважды щелкнуть на таблице и операционная система откроет нужную среду (Microsoft Excel), позволяющую выполнить необходимые изменения.

Для работы в сети Интернет была разработана технология ActiveX. Компоненты ActiveX являются портативными, в остальном они похожи на другие компоненты. Компоненты ActiveX могут работать на любой платформе, поддерживающей DCOM.

12.5. Распределенная вычислительная среда

Распределенная вычислительная среда (DCE – Distributed Computing Environment) – это стандарт, разработанный Open Software Foundation (OSF), называемой также Open Group. В основном это промежуточное программное обеспечение, доступное многим поставщикам для использования в своих продуктах. Оно обладает возможностями для поддержки множества различных видов приложений в рамках всей корпоративной среды. DCE предоставляет службу RPC, сервис безопасности, службу каталогов, службу времени, а также поддерживает распределенные файловые системы.

DCE – это набор управляющих сервисов, коммуникационные возможности которых основаны на RPC. Это программное обеспечение работает на сетевом уровне и предоставляет услуги приложениям, находящимся выше. DCOM и DCE предоставляют практически аналогичные функции. Однако DCOM был разработан Microsoft и более закрыт по своей природе.

Служба времени DCE обеспечивает синхронизацию системных часов узлов в сети, позволяет приложениям определять последовательности и планировать события, учитывая синхронизацию времени. Эта синхронизация времени предназначена для приложений, пользователи не могут напрямую получить доступ к этой функциональности. Служба каталогов дает возможность пользователям, серверам и ресурсам взаимодействовать из любых мест в сети. При передаче имени службе каталогов, она возвращает сетевой адрес ресурса вместе с другой необходимой информацией. DCOM использует **глобально уникальные идентификаторы** (GUID – globally unique identifier), тогда как DCE использует **универсальные уникальные идентификаторы** (UUID – universal unique identifier). Оба варианта применяются для однозначной идентификации пользователей, ресурсов и компонентов среды. DCE проиллюстрирован на Рисунке 9-17.

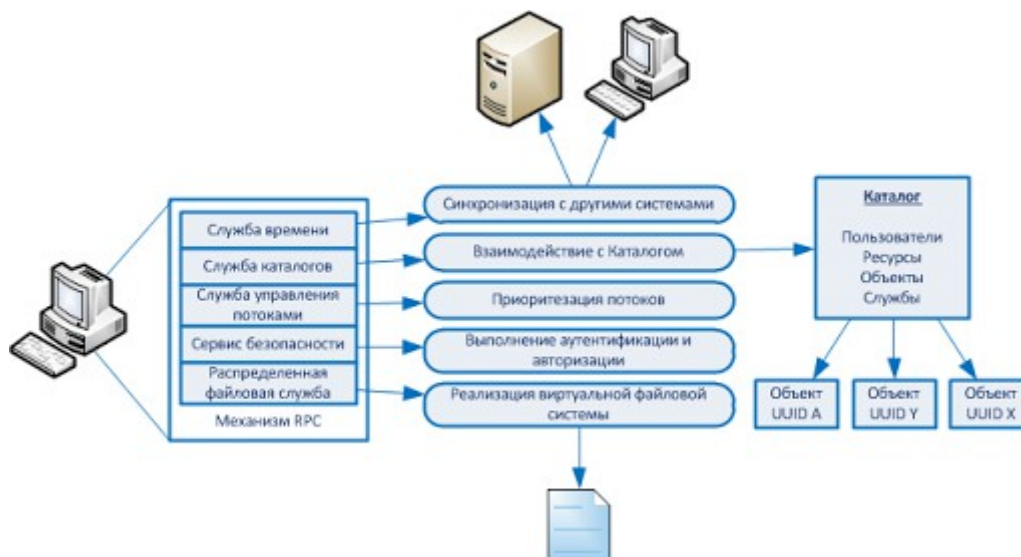


Рисунок 9-17. DCE предоставляет множество сервисов, объединенных в единую технологию

Функция RPC берет у отправляющей программы аргументы и команды, подготавливает их к передаче по сети. RPC определяет сетевой транспортный протокол, который используется для нахождения в службе каталогов узла-получателя и определения его адреса. Служба управления потоками в режиме реального времени реализует расстановку приоритетов в многопоточной среде. Сервис безопасности обеспечивает функции аутентификации и авторизации. Распределенная файловая служба (DFS - Distributed File Service) предоставляет единую интегрированную файловую систему, которую все пользователи DCE могут использовать для обмена файлами. Это важно, поскольку на различных компьютерах могут быть установлены разные операционных системы, которые могут не понимать другие файловые системы. При использовании DCE, локальная файловая система DCE работает совместно со встроенной файловой системой.

13. Экспертные системы

Экспертные системы, называемые также **системами, основанными на знаниях**, используют механизмы искусственного интеллекта (ИИ) для решения проблем.

Программное обеспечение ИИ использует нечисловые алгоритмы для решения сложных проблем, выявления скрытых закономерностей, доказательства теорем, игр, сбора данных, а также помощи в прогнозировании и диагностике целого ряда проблем. Вид вычислений, выполняемых программным обеспечением ИИ, принципиально не может быть реализован посредством традиционной программной логики.

Экспертные системы подражают человеческой логике для решения проблем, которые обычно требуют человеческого мышления и интуиции. Эти системы представляют экспертные знания в виде данных или правил, которые по мере необходимости используются для решения поставленных задач. Экспертные системы, собирают данные человеческих «ноу-хау» и хранят их в специализированной базе данных, называемой базой знаний. Фрагменты собранных данных используются для поиска причин проблемы.

Обычная программа работает со входными данными и параметрами только таким способом, который был заложен в нее при разработке. Хотя обычные программы могут без труда рассчитать выплаты по ипотечному кредиту на 20 лет вперед при известной процентной ставке, они не могут спрогнозировать расположение звезд через 100 миллионов лет, поскольку у них нет значений всех переменных, которые должны применяться для такого расчета. Хотя обе программы (обычная программа и экспертная система) имеют конечный объем доступной им информации, экспертная система будет пытаться «думать» как человек, анализировать различные сценарии и сможет в результате дать ответ даже при отсутствии

некоторых необходимых данных. Обычное программирование управляет данными с помощью процедур, в то время как люди пытаются решить сложные проблемы, используя абстрактные и символические подходы.

Чтобы получить информацию, например, из книги, человек должен прочитать ее, интерпретировать ее смысл, а затем пытаться использовать полученные знания в реальном мире. Именно таким образом пытается работать экспертная система.

Специалисты в области ИИ разрабатывают методы, которые обеспечивают моделирование информации на более высоком уровне абстракции. Эти методы являются частью используемых языков и инструментов, которые позволяют разрабатывать программы, работающие подобно человеческой логике. Такие программы, способные провести человеческую экспертную оценку в конкретной области, и называются *экспертными системами*.

Экспертная система представляет собой компьютерную программу, содержащую базу знаний и набор алгоритмов и правил, используемых для вывода новых фактов на основе имеющихся данных и входящих запросов.

Программирование на основе правил (Rule-based programming) или **логическое программирование** является распространенным способом разработки экспертных систем. Правила основаны на логических последовательностях «если-то» и определяют набор действий, которые должны быть выполнены в данной ситуации. Это один из способов, которые экспертная система использует для нахождения зависимостей, что называется **сравнением с образцом** (pattern matching). Механизм, называемый **механизмом логических выводов** (inference engine), автоматически сопоставляет факты на соответствие образцу и определяет, какие правила применимы в этом случае. Соответствующие этим правилам действия, запускаются механизмом логических выводов в соответствии с его настройками.

Предположим, что доктор Лоренц озадачен симптомами пациента и не может сопоставить их ни с одной болезнью и определить подходящее лечение. Он использует экспертную систему, чтобы она помогла ему в определении диагноза. Доктор Лоренц вводит в систему исходные данные в виде последовательности ответов на ее вопросы. Экспертная система использует полученные данные, чтобы шаг за шагом пройти через все собранные факты, в поисках образцов, связанных с известными болезнями. Хотя доктор Лоренц очень умен и считается одним из лучших врачей в своей области, он не всегда может вспомнить детали всех возможных болезней. Для экспертной системы это не представляет проблемы, т.к. она отрабатывает базу данных, набитую медицинской информацией, которой можно заполнить несколько библиотек.

Просматривая медицинскую информацию, экспертная система может увидеть, что у пациента была тяжелая травма шесть месяцев назад, три месяца назад он обращался к врачу, жалуясь на звон в ушах и затуманенное зрение, кроме того, он уже несколько лет страдает от диабета. Система обратит внимание на жалобы пациента на хроническую усталость и периодические боли. На каждом следующем шаге, экспертная система «роет» все глубже, в поисках подробной информации, а затем использует всю собранную информацию и сопоставляет ее с имеющейся базой знаний. В конце концов, экспертная система сообщает диагноз доктору Лоренцу, говоря ему, что пациент страдает от редкого заболевания, встречающегося только в Бразилии, которое вызвано определенной плесенью, растущей на бананах. Поскольку пациент страдает от диабета, его чувствительность к этому загрязнителю намного выше. Также система сообщает необходимое лечение. Тогда доктор Лоренц возвращается в комнату, где ждет его пациент, и объясняет ему проблему, защищая свою репутацию самого умного доктора.

Такая система не только использует базу данных фактов, но и собирает все богатство знаний от экспертов в конкретной области. Эти знания фиксируются с помощью интерактивных инструментов, разработанных специально для фиксации человеческих знаний. Затем

собранный база знаний используется экспертными системами, которые помогают человеку принимать решения, давая советы, помогая принимать решения быстро и правильно, а также освобождая экспертов от обычных повторяющихся задач. В частности, это может позволить компании сохранить и использовать опыт своих специалистов, даже если они уже ушли из компании.

Обычно экспертная система состоит из двух частей: механизма логических выводов и базы знаний. Механизм логических выводов обрабатывает полученную от пользователя информацию, внешние файлы, планы и другую доступную информацию. База знаний содержит данные, относящиеся к конкретной проблеме или области. Экспертные системы используют механизм логических выводов, чтобы принять решение, каким образом выполнять программу или какие правила должны быть установлены и соблюдены. Механизм логических выводов экспертной системы предоставляет системе необходимые правила, позволяя брать исходные факты и комбинировать их для создания новых фактов.

Система использует ИИ-языки программирования, позволяющие ей принимать решения в реальном мире. Система создается разработчиком экспертной системы (программистом), специалистом по базам знаний (аналитиком), с учетом мнения экспертов. Она строится на фактах, практических правилах и советах экспертов. Полученная от экспертов информация в ходе разработки системы, хранится в базе знаний и используется в дальнейшем в сценариях вопросов-ответов при работе с пользователями. Система выполняет для пользователя функции консультанта и может рекомендовать несколько альтернативных решений, одновременно анализируя конкурирующие гипотезы.

Часто экспертные системы используются для автоматизации процесса анализа журналов регистрации событий в системах выявления вторжений (IDS).

14. Искусственные нейронные сети

Искусственная нейронная сеть (ANN – artificial neural network) – это математическая или вычислительная модель, основанная на нейронной структуре мозга. Компьютеры выполняют такие виды деятельности, как расчеты с большими числами, хранение объемных документов, выполняют сложные математические функции, но они не могут распознавать образы или учиться на собственном опыте, как может это делать мозг. ANN состоит из множества модулей, симулирующих нейроны, каждый из которых обладает небольшим объемом памяти. Эти модули обрабатывают данные, которые вводятся через их многочисленные соединения. Используя правила обучения, такие системы способны учиться на примерах и могут обобщать.

Мозг хранит информацию в форме образов. Люди могут распознавать лица других людей, глядя на них с разных углов. Каждая черта лица человека состоит из сложного набора образов. Даже если видна только половина лица или лицо находится в тени и плохо освещено, человеческий мозг может дополнить недостающие фрагменты, позволяя человеку узнать своего друга и знакомого. Компьютеру, на котором работает обычное программное обеспечение, применяющее стандартную логику, необходимо иметь образец каждого фрагмента и лицо полностью, повернутое под определенным углом – тогда он сможет сравнить его с образцами и распознать.

Для того, чтобы помнить, думать, использовать предыдущий опыт, использовать логику и распознавать образы, мозг использует нейроны. Способности мозга обеспечиваются большим количеством нейронов и многочисленными связями между ними. Сила мозга является результатом генетического программирования и обучения.

Искусственные нейронные сети пытаются повторить основные функции нейронов и схемы их работы, чтобы решать проблемы по-новому. Они состоят из множества простых вычислительных нейронных модулей, связанных друг с другом. Входящие данные предоставляются одному, нескольким или всем модулям, которые, в свою очередь,

выполняют определенные функции над этими данными.

Мозг использует кластеры нейронов, осуществляющих обработку информации интерактивными и динамическими способами. Нейроны не имеют биологических ограничений по взаимосвязям между собой, поэтому нейрон может иметь тысячи соединений. Кроме того, нейроны в мозге работают в трехмерном мире, в то время как электронные блоки в ANN имеют физические ограничения на возможное количество соединений и, поэтому, они работают в двумерном мире.

Подобно мозгу, реальная мощь ANN основана на ее способности к обучению. В мозгу, маршруты соединений с нейронами, в которых хранится часто используемая информация, усиливаются, что обеспечивает более быстрый доступ к ней. Вот почему иногда бывает, что вы знаете что-то, но не можете вспомнить – откуда. Это означает, что в мозге нет четких маршрутов к хранящейся информации. Если у вас спросили ваш номер телефона, вы можете сразу же назвать его, не затратив на это никакой энергии. Но если вас спросят, как звали вашего учителя в третьем классе, может потребоваться значительно больше времени и энергии, чтобы вспомнить это. Хотя и та, и другая информация хранится в мозге, путь к номеру телефона является более четким и усиленным, поэтому вспомнить его можно быстро и легко. В ANN связи между двумя модулями, которые часто используются, также могут усиливаться, что является одной из форм обучения.

Известно, что события, которые произошли с человеком, находящемся в высокоэмоциональном состоянии, будут им запомнены гораздо лучше, скорее всего, он запомнит все до мельчайших подробностей. Например, Анна прекрасно помнит события своего дня рождения в 30 лет, на который она получила много подарков и провела его в компании друзей. Но она плохо помнит, что происходило в ее день рождения в 31 год, когда ее муж просто подарил ей открытку. Зато она прекрасно помнит события своего дня рождения в 32 года, когда ее муж вообще забыл ее поздравить, что едва не привело к разводу. Причиной того, что некоторые воспоминания являются более яркими, чем другие, заключается в том, что с этими воспоминаниями связано гораздо больше эмоций или они более значительны. В искусственных нейронных сетях некоторые входы также имеют больший вес, по сравнению с другими входами, что усиливает значение или важность получаемой через них информации, аналогично тому, что делают эмоции с людьми.

Интуиция – это качество, которое очень сложно повторить в электрических схемах и логических элементах. Для реализации интуиции, прогнозирования, догадок применяется нечеткая логика (fuzzy logic) и другие математические дисциплины. Эти подходы работают с вероятностями, используя математику и анализируя членство в различных множествах. Простым примером использования нечеткой логики, является стиральная машина, обладающая некоторым интеллектом. После того, как вы загрузили в нее грязную одежду, и бак заполнился водой, машина, основываясь на принципах нечеткой логики, посылает лучи света из одной части бака в другую. В зависимости от того, сколько света фактически было получено в другой части бака, она может определить, насколько грязна загруженная в нее одежда, учитывая плотность грязи в воде. Чтобы определить тип загрязнений (жирные или сухие) и проверить другие аспекты, могут использоваться дополнительные проверки. Стиральная машина берет всю эту информацию, анализирует ее, и на основании полученных результатов выбирает температуру стирки и нужное количество порошка. Это обеспечивает более эффективную стирку, т.к. экономится вода и порошок, стирка выполняется при наиболее правильной температуре, в течение нужного количества времени. Стиральная машина не может знать всех фактов, она не может быть уверена в точности всей собранной информации, однако она способна строить догадки на основе этой информации, которые будут довольно близки к реальности.

ANN разрабатываются таким образом, чтобы они могли принимать решения и обучаться, чтобы улучшить свою функциональность, принимая множество решений методом проб и

ошибок.

Обычная компьютерная система обладает бинарной логикой – черное или белое, она не может видеть оттенки серого между ними. Для этого необходима нечеткая логика. Она не может отличить хорошее от плохого, не различает понятия мало и много. Нечеткая логика – это метод, позволяющий компьютеру использовать такие неопределенные понятия, имеющие значение для людей, но ничего не значащие для компьютеров.

Прогнозирование фондового рынка, оценка страховых и финансовых рисков являются примерами областей, в которых требуется нечеткая логика и в которых она может быть наиболее полезна. Эти области требуют использования большого числа переменных и принятия решений на основе информации, полученной от экспертов в этих областях. Система, используя принципы нечеткой логики, может указать, какие страховые или финансовые риски являются приемлемыми, а какие – нет, не заставляя пользователя вводить большой набор условий, зависимостей «если-то» и значений переменных.

Традиционные компьютерные системы видят мир черно-белым и работают в мире точных значений. Нечеткая логика позволяет компьютеру включить неточность в язык программирования, что открывает совершенно новый мир для вычислений и решения сложных вопросов. Исследователи нейронных сетей пытаются больше узнать о работе мозга, и о природе его возможностей, чтобы усовершенствовать ANN, позволяя им решать все более сложные задачи, по сравнению с традиционными вычислительными средствами.

15. Безопасность веб-приложений

Многие ситуации и угрозы, связанные с сетью Интернет и веб-приложениями, являются уникальными. К примеру, в сети Интернет нередко нужно учитывать угрозы, связанные с возможным вандализмом. Риски мошенничества при использовании веб-приложений значительно выше, что вызвано их всеобщей доступностью. С помощью Интернета, мы можем предоставить свой продукт или услугу максимально возможной аудитории. Наученные горьким опытом, мы размещаем веб-серверы в DMZ, поэтому злоумышленники, получившие несанкционированный доступ к этим веб-серверам, не получают прямого доступа к ресурсам нашей внутренней сети. Но мы вынуждены разрешать доступ к веб-серверам через Интернет по портам, на которых работают наши веб-приложения (обычно 80 и 443), чтобы пользователи могли их использовать. Для этого мы открываем указанные порты на межсетевом экране, а это позволяет проводить атаки на веб-сервера через эти порты.

Сами приложения часто являются сложными и непонятными для интернет-продавцов. Если вы хотите продавать через Интернет домашние пироги, вам нужно будет разместить на сайте их фотографии, цены, указать способы связи с вами (телефон, электронная почта). Вам понадобится создать на сайте некую корзину для покупок, если вы собираетесь брать деньги за свои пироги, для этого вам потребуется работать с соответствующими сервисами доставки и обработки платежей... И все это для того, чтобы просто продавать пироги! Но если вы просто пекарь, вы вряд ли являетесь веб-мастером, поэтому вам придется положиться на кого-то, чтобы он создал и настроил веб-сайт для вас и установил необходимые приложения. Нужно ли вам разрабатывать для этого собственное веб-приложение на PHP или JAVA? У созданного специально для вас приложения может быть много преимуществ, т.к. оно будет максимально автоматизировать именно ваш бизнес, но нужно учитывать и риски, связанные с разработкой собственного приложения (особенно, если вы сталкиваетесь с этим впервые), а также сложность процессов и методологии, которые нужно будет организовать для этого: собственно процесс разработки, управление изменениями, управление версиями. Кроме того, нужно будет заняться выявлением уязвимостей и оценкой рисков... Стоит ли все это того, чтобы просто продавать пироги? Теперь вы понимаете, почему многие просто продают их у обочины дороги – никакой головной боли от веб-приложений!

Альтернативой разработке собственных веб-приложений является использование различных

уже готовых продуктов. Существует множество коммерческих и бесплатных продуктов почти для любого вида электронной торговли. Они написаны на различных языках программирования, разными компаниями и частными лицами, но кому мы можем верить? Есть ли у этих разработчиков процессы, о которых мы говорили выше? Учитывались ли надлежащим образом вопросы безопасности при разработке и тестировании этих приложений? Какие уязвимости есть в этих приложениях? Понимает ли наш веб-мастер, рекомендуя использовать определенное приложение, все связанные с ним риски безопасности? Теми же проблемами озабочены не только те, кто хочет продавать через Интернет пироги, но и финансовые организации, различные аукционы – все, кто занимается электронной коммерцией.

Помня обо всех этих вопросах, давайте попробуем определить наиболее опасные угрозы, связанные с использованием веб-сервера, подключенного к Интернету.

15.1. Вандализм

Эта атака, как правило, представляет собой изменение злоумышленником размещенного на веб-сайте контента – текста, заголовков, графики. Вам может показаться удивительным, что одной из угроз вашему сайту является вандализм. На самом деле, многие скрипт-кидди взламывают сайты исключительно в целях самоутверждения. Взлом вашего магазина пирогов может быть не столь впечатляющим для хакерской элиты, как дефейс сайтов *.gov или *.mil, но защищать нужно любой сайт. Здесь следует также учитывать и возможный репутационный риск: хотя многие технически грамотные клиенты смогут понять, что хакерам удалось изменить только текст на главной странице вашего сайта, но все остальные будут считать, что хакеры получили доступ ко всей базе данных ваших клиентов. Помните об общественном мнении!

15.2. Финансовое мошенничество

Деньги – сильный мотиватор для тех, кто хочет получить что-нибудь бесплатно. При проведении финансовых операций, существуют потенциальные возможности для мошенничества, особенно в анонимной среде, такой как Интернет. При этом люди, которые не осмелятся украсть газету в сломанном торговом автомате в общественном месте, легко пойдут на совершение кражи товаров и услуг через Интернет.

15.3. Привилегированный доступ

Сеть Интернет и ее участники распределены по всей планете. Организуя свою торговлю пирогами через Интернет, ваши сотрудники, которые будут печь пироги реально могут жить в Талсе, веб-мастер может жить в Сингапуре, а серверы, на которых будет размещен ваш сайт, физически будут находиться в Лондоне. При этом должен быть создан механизм, позволяющий выполнять удаленное администрирование вашего сайта, что несет в себе риск того, что кто-то другой получит административные полномочия в вашей системе. Если злоумышленник получит административный доступ, вы больше не сможете доверять системе, журналам регистрации событий и транзакциям.

15.4. Кража информации о транзакциях

Для получения денег, доставки товаров, а также чтобы просто отличать одного клиента от другого, вам потребуется собирать и хранить данные ваших клиентов. Разумеется информация о транзакциях будет мишенью для хакеров, которые будут пытаться похитить идентификационные данные клиентов, данные их платежных карт, чтобы продавать эту информацию организованным преступным группировкам, либо использовать ее самостоятельно для совершения мошеннических действий.

15.5. Кража интеллектуальной собственности

Если веб-сервер будет взломан, он может быть использован для атаки на внутреннюю сеть и

любую подключенную к ней систему. Получив доступ к веб-серверу, атакующий находится всего в шаге от баз данных и файлового хранилища, где хранятся все секреты компании. Можете ли вы позволить себе потерять свое конкурентное преимущество?

15.6. Атаки «отказ в обслуживании»

Одной из старейших в репертуаре хакеров атак является атака «отказ в обслуживании» (DoS-атака – Denial-of-Service attack). Это простая, но эффективная методика подавления системы или службы, основанная на отправке ей огромного количества запросов на соединение, что перегружает ее ресурсы и каналы связи, не позволяя реальным запросам дойти до веб-сервера. Некоторые виды DoS-атак вызывают сбой системы или службы, их переход в неуправляемое состояние и не позволяющее обрабатывать поступающие запросы – не очень хорошее состояние для веб-сервера.

Как мы уже говорили выше, веб-серверы и работающие на них веб-приложения обычно широко доступны, ведь мы хотим, чтобы пользователи имели к ним доступ из любого места на планете. Раньше широкую огласку получали случаи выявления уязвимостей программного обеспечения самих веб-серверов (например, Microsoft IIS 4.0), большинство современных атак направлено на веб-приложения, работающие на самом верхнем – прикладном – уровне. При этом, в процессе проведения атаки, чтобы усложнить ее выявление, на веб-сервер часто направляется множество запросов, что существенно затрудняет или делает невозможным журналирование событий (и, конечно же, их последующий анализ). Межсетевые экраны разрешают трафик, поступающий на 80-й порт вашего веб-сервера, т.к. это необходимо для его работы. Некоторые веб-мастера считают, что использование для всех соединений протокола SSL (Secure Sockets Layer), работающего на 443-м порту, обеспечит их защиту. Однако использование SSL позволит только шифровать передаваемый трафик и защитит его от перехвата, но шифроваться при этом будет в том числе и трафик злоумышленника, скрывая его от систем выявления вторжений и не позволяя ничего сделать для защиты самого веб-приложения. Если система выявления вторжений размещается в DMZ, то анализ ее журналов регистрации событий займет все рабочее время нескольких сотрудников. К тому же, если это стандартный IDS уровня сети, он в любом случае не окажет существенной помощи. Это, конечно, не означает, что не нужно выполнять журналирование событий, использовать SSL, межсетевые экраны и системы IDS, это означает только, что эффективность каждого из этих защитных механизмов должна быть оценена с точки зрения стратегии обеспечения безопасности вашей компании.

Ниже рассмотрен ряд защитных мер и средств, позволяющих снизить риски безопасности для веб-приложений, которые могут использоваться в дополнение к рассмотренным выше мерам и средствам.

15.7. Организация процесса обеспечения качества

Процесс обеспечения качества (quality assurance process) довольно эффективен для обеспечения уверенности в том, что серверы, на которых размещены ваши веб-приложения, настроены должным образом. Даже самые безопасные веб-приложения могут быть взломаны через уязвимости операционной системы. Указанный процесс должен учитывать все особенности сервера, начиная от установки обновлений операционной системы и программного обеспечения веб-сервера, удаления нежелательных сервисов, ненужной документации и библиотек. Чтобы убедиться, что система отвечает установленным требованиям, проводят его внешнее (из сети Интернет) и внутреннее (из локальной сети) сканирование, что должно быть сделано до перевода системы в промышленную эксплуатацию.

15.8. Межсетевые экраны для веб-приложений

В отличие от традиционных межсетевых экранов, которые смотрят только на адреса

получателя / отправителя и номера портов, межсетевые экраны прикладного уровня выполняют глубокий анализ пакетов, что позволяет им выявлять и блокировать определенное поведение, связанное с проведением атак, а также аномалии и нежелательные команды протоколов (например, команду POST в протоколе HTTP).

15.9. Системы предотвращения вторжений

Система предотвращения вторжений (IPS – Intrusion Prevention System), в отличие от систем выявления вторжений (IDS), может реально *предотвращать* выявленные ей атаки. Такие системы обычно устанавливаются «в разрыв» (inline), т.е. весь трафик проходит через них и проверяется, прежде чем он достигнет серверов, размещенных за системой IPS. Это может вызвать сложности с точки зрения производительности, что, как правило, приводит к увеличению стоимости и аппаратных требований. Некоторые считают, что такие системы являются расширенными вариантами систем IDS.

15.10. Реализация SYN-прокси на межсетевом экране

Наиболее распространенным вариантом DoS-атак является SYN-флуд. При проведении атаки SYN-флуд атакующий отправляет поддельные запросы на соединение (пакеты SYN) на сервер, пытаясь перегрузить его и не позволить обслуживать запросы реальных пользователей. После получения от злоумышленника пакетов SYN, сервер будет в течение определенного периода времени ждать завершения фиктивных соединений (которые, конечно, никогда не завершатся), а запросы реальных пользователей будут игнорироваться сервером.

Реализация SYN-прокси на межсетевом экране позволит ему управлять подключениями к серверу. SYN-прокси постоянно контролирует количество запросов SYN в единицу времени и сравнивает его с заранее определенным пороговым значением (например, 500 в секунду). Если пороговое значение было достигнуто, а запросы все продолжают поступать, межсетевой экран может удалить самые старые из запросов, по которым так и не было установлено соединения, тем самым позволяя серверу продолжать обслуживать запросы на подключения от реальных пользователей. Не все межсетевые экраны имеют такую функциональность, обычно для этого нужны межсетевые экраны, поддерживающие «таблицы состояний» для соединений.

Все эти решения являются прекрасными, но в основе всего этого остается веб-приложение. Истинная безопасность веб-приложений должна начинаться с разработки и внедрения безопасных прикладных сервисов и позволять использовать другие виды защитных механизмов для снижения рисков. Теперь мы более детально рассмотрим отдельные угрозы и уязвимости, связанные с этой темой.

15.11. Специфические угрозы веб-среде

В следующих разделах рассматриваются наиболее распространенные виды уязвимостей, угроз и возникающих сложностей в веб-среде. Будут рассмотрены следующие аспекты:

- Сбор информации
- Административные интерфейсы
- Аутентификация и управление доступом
- Управление конфигурациями
- Проверка входных данных
- Проверка параметров
- Управление сессиями

Сбор информации

Сбор информации (information gathering) является, как правило, первым шагом в методологии злоумышленника. Собранная информация может позволить злоумышленнику сделать выводы (догадаться) о дополнительной информации, которая может быть использована им для нарушения безопасности систем. К сожалению, значительную часть информации можно получить из общедоступных источников. Крупные поисковые системы упрощают злоумышленнику задачу сбора информации, поскольку они накапливают информацию, хранят ее в своих кэшах и могут возвращать результаты поиска из кэша поисковой системы, что не требует не только выполнения атаки, но даже подключения к целевому веб-серверу компании.

В большинстве случаев виновными в раскрытии информации оказываются разработчики и веб-администраторы сервера, которые просто пытаются делать свою работу. Комментарии в исходном тексте программы, указанные разработчиком для пояснения работы процедур, или файлы резервных копий, которые администратор хранит на самом веб-сервере, сами по себе не являются вопиющими проблемами безопасности, но если злоумышленник получит доступ к ним, они помогут ему узнать гораздо больше, чем хотелось бы компании. Даже сообщения об ошибках, возвращаемые сервером в ответ на некорректные запросы, могут содержать физический путь к базе данных, номер версии сервиса и т.п., что может быть использовано злоумышленником в качестве отправной точки для получения несанкционированного доступа к системе.

Более профессиональные атакующие выходят за пределы поисковых систем, чтобы изучить содержимое всех доступных файлов на сервере в поисках возможных ключей к пониманию структуры внутренней сети или строк соединений, используемых веб-сервером для подключения к серверу базы данных.

Для того чтобы веб-сервер предоставлял активное содержимое и единый интерфейс, что необходимо современным веб-пользователям, серверы должны обращаться к источникам данных, обрабатывать код и возвращать результаты веб-клиентам. Для работы этих механизмов, должен быть написан соответствующий код и передан веб-браузеру в соответствующем формате. Одной из технологий, позволяющей веб-разработчикам повторно использовать одно и то же содержимое, вставляя его в несколько веб-документов, является SSI (Server Side Includes - Включения на стороне сервера). Обычно это предполагает использование включения в код необходимых команд и информации из файлов (.inc). Однако если к этим файлам получит доступ злоумышленник, то он сможет увидеть код и изменить его для «включения» других файлов, содержащих критичную информацию. Другие технологии, такие как ASP (Active Server Pages - Активные серверные страницы) (страницы с расширением файла *.asp) используются для создания «активной» пользовательской среды. Эти файлы могут раскрыть любой содержащийся в них критичный код, если есть возможность их просмотра. Разработчики должны избегать размещения критичного кода в файлах SSI или ASP (например, строки подключения к базе данных или текст программы, реализующий бизнес-логику, которая является объектом интеллектуальной собственности), при этом даже в случае, если злоумышленник сможет получить доступ к такому файлу, в нем не будет ничего важного. В прошлом уже было найдено множество уязвимостей, позволяющих получить доступ к таким файлам, поэтому есть все основания полагать, что риск их прочтения злоумышленником достаточно велик.

Другим советом, позволяющим разработчикам предотвратить раскрытие физического размещения компонентов или паролей, используемых для подключения к базе данных, является использование имени источника данных (DSN – Data Source Name). Это логическое имя хранилища данных, а не буква диска и каталог, в котором размещена база данных. Такое логическое имя может быть использовано при программировании для интерфейса ODBC. При использовании ODBC DSN для хранения таких значений, хранение реальных физических путей осуществляется в реестре системы, а не в коде программы. Кроме того, эта технология упрощает изменение кода, поскольку при этом строки соединений являются

переменными, хранящимися в системном реестре. Поэтому это является хорошей практикой.

Основной контрмерой против методов сбора информации является попытка сделать так, чтобы атакующий мог получить только ту информацию, которую вы осознанно делаете общедоступной, либо минимально ограничиваете ее доступность. Разработчики должны понимать, что существует потенциальная возможность, что их код будет просмотрен кем-то за пределами компании, а администраторы должны регулярно проверять ссылки поисковых систем на веб-сайты компании, адреса электронной почты, ссылки на файлы и хранилища данных. Многие веб-сайты и целые книги посвящены вопросам сбора информации из общедоступных баз данных, поэтому такие проверки будут хорошим примером проявления должной осмотрительности.

Административные интерфейсы

Каждый хочет работать из кафе или из дома, сидя в пижаме за собственным компьютером. Веб-мастера и веб-разработчики особенно любят такую работу. В некоторых системах установлены жесткие ограничения, позволяющие выполнять администрирование только с локального терминала, однако в большинстве систем таких ограничений нет, и интерфейс для администрирования системы предоставляется удаленно, в том числе через Интернет. Хотя это может быть очень здорово и удобно для веб-мастеров, это также является точкой входа в систему для неуполномоченных пользователей.

В большинстве случаев, использование *административного веб-интерфейса* является плохой идеей. Он может использоваться только в том случае, если определены его уязвимости, для них реализованы защитные меры, делающие административный веб-интерфейс не менее (а лучше даже более) безопасным, чем само администрируемое через него веб-приложение, приняты остаточные риски, на которые компания готова осознанно пойти.

Очень плохой привычкой, которая иногда оказывает свое влияние даже в условиях повышенной безопасности, является жесткое указание в коде аутентификационной информации для соединений с интерфейсами управления или предоставление опции «Запомнить пароль». Это упрощает жизнь администратора, но дает слишком много прав доступа тому, кто получает эту информацию вопреки желанию администратора.

Но давайте смотреть фактам в лицо, большинство коммерческого программного обеспечения и серверов веб-приложений по умолчанию устанавливают те или иные виды административной консоли. Знания этого факта и методов сбора информации, рассмотренных выше, должно быть вполне достаточно для компании, чтобы всерьез рассматривать связанные с этим угрозы. Если интерфейс не нужен, он должен быть отключен. Если он нужен, следует рассмотреть вопросы его использования в политике и процедурах компании.

Простой контрмерой против этой угрозы является простое удаление или блокирование внешних интерфейсов управления, но это может расстроить ваших администраторов. Если внешние интерфейсы управления все же необходимы, следует использовать механизмы строгой аутентификации, это значительно лучше, чем просто имя пользователя и пароль. Другим хорошим вариантом является настройка ограничений на подключение к административным интерфейсам. Многие системы позволяют настроить разрешения доступа только для конкретных IP-адресов или сетевых идентификаторов, при этом возможность использования административных интерфейсов будет предоставляться только с этих компьютеров.

И, наконец, наиболее безопасным вариантом является интерфейс управления, доступ к которому предоставляется только по отдельному соединению, вне основного диапазона адресов, для подключения к которому используется отдельный канал связи. Это позволяет избежать влияния любых уязвимостей, присутствующих в среде, в которой работает система.

Примером такого канала является использование модема, подключенного к веб-серверу, через который можно подключиться напрямую и выполнить настройку сервера с помощью его локального интерфейса, вместо подключения через Интернет и выполнения настроек с помощью веб-интерфейса. При этом такое модемное соединение должно осуществляться по зашифрованному каналу, например, с использованием SSH.

Аутентификация и управление доступом

Если вы уже имеете опыт использования систем дистанционного банковского обслуживания, онлайн-покупок, использования сервисов дистанционного обучения, вам наверняка приходилось не раз регистрироваться в веб-приложениях. Как для клиента, так и для поставщика услуги, тема **аутентификации и управления доступом** является очевидной проблемой. Клиенты, конечно, хотят, чтобы механизмы управления доступом обеспечивали безопасность и конфиденциальность, они хотят работать в доверенной среде, но при этом они совершенно не хотят, чтобы этот процесс был обременителен для них самих. С другой стороны, провайдер услуги хочет обеспечить максимальный уровень безопасности для клиента, насколько позволяет производительность используемого оборудования и программного обеспечения, действующие требования и бюджет. В результате, для доступа к веб-приложениям, как правило, по-прежнему используются имена и пароли.

Причиной, по которой пароли остаются самым распространенным способом аутентификации, является доступность. Безусловно, доступность – это прекрасно, но только если все использующие ваш сайт являются законными пользователями. Однако доступность, при которой любой злоумышленник может анонимно получить несанкционированный доступ к вашему сайту, уже совсем не прекрасна. Пароли не могут обеспечить многого. Они продолжают использоваться, поскольку они остаются дешевым и достаточно эффективным способом аутентификации, но реально они не могут гарантировать, что пользователь «Jsmith» на самом деле является Джоном Смитом, они просто говорят о том, что лицо, использующее учетную запись Jsmith, ввело правильный пароль. Но это может быть кто угодно! Разве вы сами никогда не использовали чужую учетную запись для каких-либо целей?

Было бы вполне логично предположить, что система, на которой хранится критичная информация (медицинская, финансовая и т.д.), может стать мишенью для атак. Распространенной практикой является сбор имен пользователей через поисковые системы с последующей попыткой использования их для входа в веб-приложения, либо использование для этих целей часто встречающихся имен (таких как, Ivanov). Кроме того, пользователи слишком часто при регистрации на различных сайтах используют одни и те же имя и пароль. Вспомните, какие учетные данные вы использовали последний раз, когда регистрировались на каком-нибудь сайте, чтобы просто скачать какой-нибудь бесплатный документ? Атакующие могут создавать веб-сайты специально для того, чтобы собирать через них имена и пароли пользователей. Такие сайты могут выглядеть вполне дружелюбно и безобидно, предлагая вам узнать свой IQ, прочитать чужие SMS, поучаствовать в лотерее, но их реальная цель – сбор учетных данных, которые затем будут использоваться злоумышленниками для попытки регистрации на других ресурсах. Помните, что необученный и неосведомленный о таких угрозах пользователь является большой угрозой для компании!

Еще одним слабым местом аутентификации (особенно, если используются пароли) является то, что злоумышленники (также, как и законные пользователи) могут блокировать учетные записи, выполняя последовательно несколько попыток входа с неправильным паролем. Система, настроенная в соответствии с рекомендациями по противодействию подбору паролей, после нескольких неудачных попыток регистрации автоматически заблокирует учетную запись. Противодействие подбору паролей, безусловно, также необходимо для веб-приложений, как и для традиционных приложений. Блокировка учетных записей при

выявлении попыток подбора пароля является эффективной контрмерой. Но представьте, что перебор осуществляется не для паролей, а для имен пользователей. Если злоумышленник последовательно пытается войти в систему под каждой учетной записью, используя при этом неправильные пароли, фактически он заблокирует учетные записи всех пользователей. Каковы будут последствия, если несколько тысяч клиентов банка, использующие систему интернет-банкинга, вдруг не смогут войти в систему? Справится ли служба технической поддержки с таким потоком звонков и заявок? Это может привести к пересмотру политики вашей компании в отношении создания учетных записей и сброса паролей. Как вы будете проводить аутентификацию пользователей, забывших свой пароль? Какой пароль вы установите при сбросе забытого пароля? Вы будете использовать пароль по умолчанию при создании новых учетных записей? Все это должно быть определено в политике и процедурах. Помните, что общий уровень вашей безопасности равен уровню безопасности самого слабого звена.

Решением для защиты от массовой атаки на учетные записи с целью их блокировки может быть только использование временной блокировки учетных записей, т.е. после нескольких неудачных попыток регистрации учетная запись блокируется на ограниченное время (от 30 минут для сайтов с низким риском, до трех часов или даже до суток для сайтов с высоким уровнем рисков), после чего она автоматически разблокируется. Ваша компания должна определить, какой уровень риска она готова принять. Использование механизмов многофакторной аутентификации не обязательно исключит атаки такого типа, однако это позволит значительно повысить защиту от несанкционированного доступа и сделает его менее вероятным.

Чтобы выявить атакующую систему (или системы), следует анализировать журналы регистрации событий, хотя это вряд ли будет реальный компьютер, принадлежащий злоумышленнику. Анализ журналов регистрации событий позволит вам увидеть сам факт атаки – вы увидите многочисленные попытки регистрации, исходящие от отдельного компьютера (или нескольких компьютеров). Это поможет вам внести соответствующие изменения в правила межсетевого экрана и других систем безопасности.

И наконец, хорошей практикой является обмен всей аутентификационной информацией с использованием защитных механизмов. При этом, как правило, выполняется шифрование паролей и учетных данных, либо шифрование всего канала связи. В наше время было бы неразумно использовать незащищенный доступ на критичный веб-сайт, т.к. преимущества от реализации доступа посредством SSL значительно превышают стоимость покупки сертификата для своего веб-сайта и расходы на дополнительную обработку, связанную с зашифрованием и расшифрованием информации на каждой стороне. Некоторые крупные сайты, предоставляющие критичные сервисы, продолжают работать без выполнения шифрования информации аутентификации и подвергают себя и своих пользователей угрозе, поскольку злоумышленники могут перехватить имена пользователей и их пароли.

Управление конфигурациями

Управление конфигурациями (configuration management) – это просто концепция управления конфигурациями ваших систем. Учетные записи по умолчанию и их пароли, файлы примеров в системе, а также интерфейсы управления – все это должно быть выявлено и учтено при определении базовых требований безопасности в веб-среде. Прежде чем система будет переведена в промышленную эксплуатацию, должна быть проведена проверка ее соответствия действующей политике. Хотя, конечно, на практике такой подход используется не часто.

На практике, при разработке веб-приложений, обычно все ограничивается тем, что разработчик устанавливает приложение в «тестовую» среду, которая эмулирует промышленную среду (с такой же операционной системой и веб-сервером), после чего разработчик пишет код, проверяющий функциональность приложения, и готовит отчет об

успешном тестировании. По результатам этого тестирования, без каких-либо дополнительных проверок, приложение переносится в промышленную эксплуатацию. При этом очень распространенной ошибкой является дальнейшее использование тестовой среды, в которой была проверена работоспособность приложения, в качестве промышленной среды. Конечно, приложение в такой среде точно будет работать, однако нужно понимать, что разработчик почти наверняка создавал тестовую среду без учета даже минимальных требований безопасности, необходимых для промышленной среды. Слишком часто в процессе тестирования веб-приложений основное внимание уделяется доступности, а не целостности и конфиденциальности. Все думают (ошибочно): «Давайте сначала посмотрим, работает ли это приложение вообще, и тогда уже будем ограничивать его функциональность». Но про необходимость ограничений потом забывают, что часто становится причиной компрометации системы.

Если у компании нет своей собственной команды разработчиков и она просто покупает приложение, она должна реализовать процесс, позволяющий выявить уязвимости этого приложения и убедиться в том, что приложение внедрено безопасно. Приложения, которые просто были установлены с использованием настроек по умолчанию, редко бывают безопасными. Многие приложения при установке по умолчанию создают административные учетные записи со стандартными паролями, известными хакерскому сообществу. Они устанавливают страницы ошибок с подробной информацией и файлы примеров, предназначенные для помощи в процессе внедрения, что часто является слабостью, которой могут воспользоваться злоумышленники. Вас может удивить, разве атакующий сможет найти все эти вещи на вашем веб-сервере? Ответ на этот вопрос вы можете найти в вашей любимой поисковой системе.

Поисковые системы (например, Google) очень хорошо анализируют веб-страницы и ссылки в них, они проходят по всем ссылкам и заносят в свой каталог все, что могут найти. Это отличная возможность, поскольку она значительно упрощает поиск кулинарных рецептов или цитат из фильмов, но этой возможностью могут воспользоваться и хакеры, которые используют хитро сформулированные поисковые запросы, позволяющие им найти интерфейсы управления, страницы ошибок со сведениями о конфигурации, а иногда и разделы вашего сайта, которые вы не собирались выставлять на всеобщее обозрение.

Решением всех этих проблем является наличие четких политик и процедур. Процедуры должны предусматривать обязательное удаление создаваемых по умолчанию настроек, учетных записей и паролей, своевременное применение патчей безопасности, выпускаемых производителем. При переносе приложения из тестовой среды в промышленную, следует удалять или пересматривать все заполненные на этапе тестирования списки контроля доступа (ACL) в приложении, что позволит обеспечить правильное разграничение прав доступа (в особенности к критичным файлам и данным) и надлежащий уровень безопасности. Перед началом промышленной эксплуатации новой системы, из нее должно быть удалено все, к чему вы не хотели бы давать доступ всему миру через поисковые системы. В частности, следует удалить файлы документации, файлы примеров и стандартные страницы ошибок.

Проверка входных данных

Веб-серверы не такие умные, они просто делают то, что им говорят. Они предназначены для обработки запросов, переданных посредством определенного протокола. Термин «протокол» означает правила, которым нужно следовать в определенной ситуации, чтобы выполнять необходимые коммуникации. Пользователь использует веб-браузер. Когда он вводит в адресную строку `http://www.website.com/index.htm` и нажимает Enter, он использует протокол HTTP (Hypertext Transfer Protocol), чтобы запросить файл «index.htm» с сервера «www» в пространстве имен «website.com». Запрос в такой форме называется URL (Uniform Resource Locator - Единый указатель ресурсов), он похож на то, как мы говорим (ну, по

крайней мере, мы можем легко прочитать его). Как и во многих других случаях, в компьютерном мире существует несколько различных способов оформления такого запроса, поскольку компьютеры могут «говорить» на нескольких различных «языках» (используя, например, двоичную, шестнадцатеричную и другую кодировку), каждый из которых интерпретируется и обрабатывается системой в качестве команд. Проверка таких запросов является частью **проверки входных данных** (input validation), обычно она связана с некоторыми запрограммированными правилами проверки. Тот факт, что эти правила должны быть заранее запрограммированы в коде программы означает, что вероятно могут существовать некоторые «хитрые» запросы, которые могут обойти такие правила проверки.

Вот некоторые такие хитрые примеры:

- **Обход пути или каталога** (Path or directory traversal). Эту атаку иногда называют «dot dot slash», поскольку она выполняется путем многократной вставки символов «../» в адрес URL для прохода в каталоги, которые не должны быть доступны из сети Интернет. Команда «. ./» в командной строке говорит системе, что нужно вернуться в предыдущий каталог (попробуйте в командной строке набрать, «cd ../»). Если каталогом по умолчанию веб-сервера является «C:\Inetpub\WWW», передача ему URL-запроса
`<http://www.website.com/scripts/../../../../../../../../windows/system32/cmd.exe?/C+dir+C:\>` даст ему команду вернуться на несколько каталогов выше, проходя весь путь до корневого каталога диска, а затем перейти в каталог операционной системы (Windows\System32), запустить утилиту cmd.exe и вывести содержимое диска «C:».
- **Кодировка Unicode** (Unicode encoding). Unicode – это стандартный механизм, разработанный для того, чтобы позволить использовать весь диапазон возможных символов (языки разных стран используют суммарно более 100 000 текстовых символов). Unicode поддерживает различные наборы символов (например, китайский), и, в настоящее время, многие приложения поддерживают его по умолчанию. Если мы внесли в настройки наших систем запрет на обработку команд «. ./» в запросах для описанного выше обхода каталогов, злоумышленник может воспользоваться кодировкой Unicode, чтобы дать веб-серверу ту же команду «/», но с использованием одного из вариантов Unicode-представлений этого символа (существует три: %c1%1c, %c0%9v и %c0%af). Такой запрос может проскользнуть незамеченным и будет обработан веб-сервером.
- **Кодирование URL** (URL encoding). Может быть, вы когда-нибудь обращали внимание, что в адресной строке веб-браузера «пробел» преобразуется в «%20» (шестнадцатеричный код символа «пробел»), поскольку «пробел» не является допустимым символом в запросе URL. Аналогично атаке с использованием Unicode-символов, злоумышленник может использовать представление символов в виде их кодов, чтобы обойти фильтрацию и передать серверу нужный запрос.

Помимо простого предоставления пользователям статичных файлов, почти любому веб-приложению требуется принимать тот или иной ввод от пользователей. При использовании интернет-ресурсов вас постоянно просят ввести какую-нибудь информацию – имя пользователя, пароль, данные банковской карты и т.п. Для веб-приложений это просто входные данные, которые должны быть обработаны наравне с остальными частями кода приложения. Обычно полученные таким образом входные данные записываются в переменную, которую код программы обрабатывает, основываясь на некоторой, заранее запрограммированной логике, например, ЕСЛИ [поле ввода имени пользователя] = X И [поле ввода пароля] = Y ТОГДА аутентифицировать. Это будет хорошо работать, при условии, что в поля ввода всегда помещается адекватная информация, но что

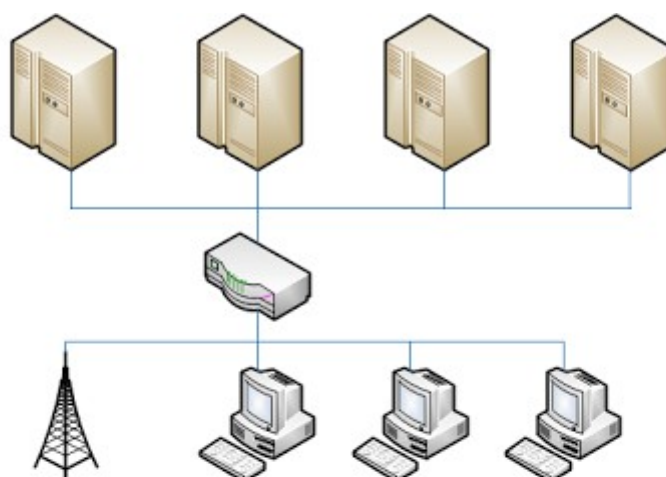
если пользователь введет неадекватную информацию? Разработчики должны предусмотреть любые варианты. Они должны предположить, что возможны ситуации, когда программой будут получены неправильные входные данные, и программа должна обработать их надлежащим образом. Для этого код процедур должен предусматривать подобные ситуации и указывать, что делать системе, если получено не то, что ожидалось.

Переполнение буфера – это, пожалуй, наиболее известная ошибка проверки входных данных. Буфер – это область, зарезервированная приложением для хранения в ней неких данных, таких как пользовательский ввод. После получения входных данных приложением, указатель команд говорит приложению, что нужно делать с полученными данными, сохраненными в буфере. Переполнение буфера происходит в тех случаях, когда приложение ошибочно принимает неправильные входные данные, объем которых превышает объем отведенного под них буфера, что приводит перезаписи указателя команд в коде. Это позволяет выполнить любой код, на который теперь ссылается указатель команд (например, код, переданный приложению в качестве пользовательского ввода и теперь находящийся в буфере), причем этот код будет выполнен в контексте безопасности приложения.

Проверка на стороне клиента (client-side validation) – это контроль входных данных, выполняющийся клиентской частью приложения перед их передачей на сервер для обработки. Например, если при заполнении веб-формы вы пропустили какое-то важное поле, при нажатии кнопки подтверждения ввода вы сразу же получите сообщение об ошибке, информирующее вас, что вы забыли заполнить одно из обязательных полей. Это пример проверки, выполняющейся на стороне клиента. Это гораздо лучше, чем передавать неполные данные на сервер, чтобы затем получить от него такую же ошибку. Проблема возникает в тех случаях, когда проверка на стороне клиента является единственной проверкой данных. В таком случае сервер надеется, что клиентская часть приложения хорошо сделала свою работу и уже убедились, что введенные данные – правильные. В обычной ситуации этого действительно должно быть достаточно. Однако, если злоумышленник может перехватить трафик между клиентом и сервером и внести в него изменения, либо просто напрямую передавать некорректные запросы на сервер без использования клиентской части, нарушение безопасности весьма вероятно.

В системе, в которой достаточно слабо реализованы проверки входных данных, злоумышленник будет пытаться вводить нужные ему команды операционной системы в поля ввода вместо значений, которые эта система ожидает получить (например, вместо имени пользователя и пароля), пытаясь обмануть систему и заставить ее выполнить эти команды. Помните, что компьютеры делают то, что им говорят, поэтому, если злоумышленник сможет получить возможность запуска команд операционной системы, они выполнят их, поскольку будут считать, что их запуск инициирован самим приложением. Если веб-приложение обращается к базе данных, в большинстве случаев существует угроза выполнения SQL-инъекций (SQL-Injection), когда вместо правильных входных данных злоумышленник помещает в поля ввода команды для работы с базой данных, которые при получении разбираются и выполняются приложением. SQL-выражения могут быть использованы злоумышленником, в частности, для обхода аутентификации и получения всех записей в базе данных.

Помните, что различные уровни системы (см. Рисунок 9-18) имеют собственные уязвимости, которые должны быть выявлены и исправлены.



Политики безопасности	Политика использования паролей, политика журналирования событий, политика предоставления доступа к системам, политика распределения прав доступа ...
Веб-приложения	Apache, Internet Explorer, Firefox, Microsoft IIS, Tomcat, WebLogic, ColdFusion, SSH, Telnet ...
Сторонние приложения	Lotus Notes, Microsoft Exchange, Adobe Acrobat, Windows Media, Sendmail ...
Базы данных	Oracle, Microsoft SQL Server, MySQL, IBM DB2, IBM DB/400, Sybase, Lotus Domino ...
Операционные системы	Microsoft Windows, Linux, Unix, Solaris, Mac OS, BSD, AIX, AS/400, Novell NetWare ...
Сети	IPSec, PPTP, Сетевые файловые системы, DHCP, DNS, LDAP, SNMP ...
Аппаратное обеспечение	Маршрутизаторы, Коммутаторы, Беспроводные точки доступа, Аппаратные межсетевые экраны ...

Рисунок 9-18. Атаки могут происходить на различных уровнях

Похожим образом выполняются атаки межсайтового скриптинга (XSS – cross-site scripting), которые пришли на смену переполнению буфера в качестве самой большой угрозы для веб-приложений. XSS-атака – это атака, для выполнения которой злоумышленник отыскивает на веб-сайте уязвимость, которая позволит ему внедрить вредоносный код в веб-приложение. При этом этот вредоносный код может быть выполнен в веб-браузере ничего не подозревающих пользователей, как только они зайдут на этот сайт. Отключение в настройках браузера пользователя выполнения любых сценариев легко решило бы эту проблему, но тогда пользователь не смог бы работать со множеством веб-приложений.

Все эти атаки, рассмотренные в этом разделе, связаны ошибочным предположением разработчиков, что входные данные всегда корректны. Эффективными контрмерами для борьбы с этой проблемой являются фильтрация всех запросов для выявления в них заведомо «вредоносных», применение правила не доверять входящей информации от пользователя без ее предварительной тщательной проверки, внедрение строгой политики, обеспечивающей обязательное включение необходимых проверок входных данных во всех приложениях.

Проверка параметров

Вопрос проверки параметров сродни вопросу проверки входных данных, рассмотренному ранее. **Проверка параметров** (parameter validation) – это проверка полученных приложением значений на предмет их нахождения в определенном диапазоне, перед тем, как передать их серверу приложения для обработки. Основное отличие проверки параметров от проверки входных данных заключается в том, что проверяются значения, полученные из переменной среды, определенной приложением, а не данные, введенные пользователем. Проверка

параметров нужна для противодействия атакам, связанным с манипулированием значениями, которые может настраивать пользователь, и которые система учитывает в процессе своей работы. В основном это относится к настройкам, изменение которых не предусмотрено интерфейсом приложения. Никому нельзя слепо доверять, тем более если речь идет о компьютерах, у которых нет здравого смысла, которым наделены люди.

Чтобы создать приложение для широкого круга пользователей, разработчики веб-приложения должны использовать механизмы для отслеживания тысяч пользователей, которые могут подключаться к приложению в любое время. Протокол HTTP сам по себе не позволяет управлять состоянием пользовательских соединений, фактически, они просто подключаются к серверу, получают нужные им объекты (файлы *.htm, графические файлы и т.д.), запрошенные в коде HTML (HTTP Markup Language), а потом отключаются (либо соединение завершается по таймауту). Но если браузер пользователя сразу отключается от сервера, как сервер сможет распознать того же пользователя, когда он вернется? Вероятно, пользователь будет сильно раздражен, если ему придется повторно вводить все свои данные только из-за того, что он слишком долго смотрел список рейсов, чтобы забронировать через Интернет билет на самолет, и соединение его браузера с веб-сервером завершилось по таймауту. Чтобы этого не случилось, веб-разработчики используют куки, которые передаются клиенту, чтобы потом по ним сервер мог «вспомнить» этого клиента и все сведения о состоянии соединения с ним. Куки (cookie) – это не программа, это просто набор данных, которые передаются веб-браузеру пользователя и хранятся в памяти браузера (так называемые сеансовые куки), либо локально на компьютере пользователя в виде файла (называемые постоянными куки), чтобы в любой момент передать информацию о состоянии подключения пользователя обратно на сервер. Примером использования куки является корзина покупок на сайте интернет-магазина. Когда вы кладете покупки в вашу виртуальную корзину, соответствующие сведения передаются вашему веб-браузеру путем обновления сеансового куки на вашей системе. Такие сайты не могут работать без использования куки.

Поскольку сеансовые куки хранятся в памяти, как правило, большинство пользователей не может получить к ним доступ, поэтому веб-разработчики при проектировании систем часто не считают, что эти куки могут представлять серьезную угрозу. К примеру, разработчики часто реализуют функцию блокировки учетной записи после определенного числа неудачных попыток регистрации (то, о чем мы говорили ранее). Если разработчик реализует эту функцию таким образом, что она хранит количество оставшихся у пользователя попыток входа в сеансовом куки, может возникнуть уязвимость. В таком случае, если приложение настроено на блокировку учетной записи после трех неудачных попыток входа, сервер может передать клиенту сеансовый куки, записав в него значение, типа «Допустимое количество входов = 3». После каждой неудачной попытки, сервер уменьшает это количество в куки пользователя на единицу. Когда оставшееся количество становится равным нулю, пользователь перенаправляется на страницу с указанием, что «Ваша учетная запись заблокирована».

Злоумышленник может перехватить и внести изменения в передаваемую между браузером пользователя и веб-сервером информацию, используя для этого программное обеспечение, называемое веб-прокси (web proxy). Существуют свободно распространяемое, общедоступное программное обеспечение веб-прокси (например, Achilles или Burp Proxy), которым он может воспользоваться для этих целей. Когда сервер передает пользователю сеансовый куки, в котором записано «Допустимое количество входов = 3», злоумышленник перехватывает эту информацию с помощью такого веб-прокси и изменяет значение, например, на 50000. Это позволит ему эффективно выполнить брутфорс-атаку и подобрать пароль пользователя, если эта система не использует дополнительные механизмы проверки.

Использование веб-прокси может также позволить воспользоваться скрытыми полями в веб-

страницах. Скрытые поля не отображаются в пользовательском интерфейсе, но они содержат некоторые значения, передаваемые на сервер при отправке веб-формы. Например, разработчик может использовать скрытые поля для указания цен на товары, показанные на веб-странице, вместо того, чтобы делать ссылку на прайс-лист на сервере. В таком случае злоумышленник, используя веб-прокси, может перехватить передаваемую от пользователя информацию и внести изменения в стоимость товаров (например, установить стоимость всех товаров 1 рубль), прежде чем эта информация попадает на сервер. Это совсем не сложно сделать, если в приложении не реализованы другие проверки.

Контрмерой, позволяющей снизить риски, связанные с такими угрозами, является *надлежащая проверка параметров*. Эта проверка может состоять из проверок на входе и проверок на выходе. В клиент-серверной среде, проверка на входе может быть реализована на стороне клиента, до передачи запросов на сервер. Однако даже при реализации такой проверки на клиентской стороне, сервер также должен выполнять проверку входных данных до начала их обработки, т.к. компьютер пользователя обычно защищен хуже, чем сервер, и его безопасность может быть нарушена.

- **Проверка на входе (Pre-validation).** Проверяет, что входные данные имеют правильный формат и соответствуют требованиям приложения, и только после успешной проверки данные передаются приложению. Примером может быть проверка вводимых в поле формы символов, которая не позволяет пользователю ввести буквы в цифровое поле.
- **Проверка на выходе (Post-validation).** Проверяет, что данные, выходящие из приложения, соответствуют ожиданиям (то есть находятся в пределах заранее определенного разумного диапазона).

Управление сеансами

Как было отмечено ранее, управление несколькими тысячами различных клиентов, подключающихся к веб-приложению, является непростой задачей. Вопрос *управления сеансами* (session management) должен быть рассмотрен и решен до размещения приложения в сети Интернет. Чаще всего используется способ управления пользовательскими сеансами, при котором каждому соединению присваивается уникальный идентификатор сеанса. Идентификатор сеанса – это значение, которое отправляется клиентом на сервер вместе с каждым запросом, что позволяет серверу или приложению однозначно идентифицировать клиента. В случае если злоумышленник сможет получить или угадать идентификатор сеанса аутентифицированного клиента и направить его на сервер в качестве собственного идентификатора сеанса, он сможет «обмануть» сервер и получить доступ к сеансу клиента.

Старое правило «никогда ничего не отправлять в открытом виде», безусловно, применимо и здесь. Трафик HTTP по умолчанию не зашифрован, в нем не предусмотрено ничего для противодействия перехвату злоумышленником идентификаторов сеансов из передаваемых по сети данных. Поскольку идентификаторы сеансов передаются, как правило, по протоколу HTTP, следует обеспечить их защиту.

Возможность угадать или предсказать идентификаторы сеансов, также представляет угрозу в такой среде. Поэтому использование идентификаторов сеансов с последовательными номерами является большой ошибкой. Более целесообразно использовать случайные идентификаторы сеансов нужной длины, что не позволит предсказать значение идентификатора. Следует включать в передаваемые запросы некие штампы времени, что позволит предотвратить атаку повтора (replay attack), при которой злоумышленник перехватывает трафик легитимного сеанса, а затем повторяет его, чтобы аутентифицировать свой сеанс. Кроме того, любой куки, используемый для хранения информации о состоянии соединения, должен быть зашифрован.

16. Мобильный код

Мобильным кодом (mobile code) называется код, который может передаваться по сети, для запуска на другой системе или устройстве. Существует множество вполне реальных причин для использования мобильного кода – например, апплеты веб-браузера, которые могут выполняться в фоновом режиме для просмотра дополнительного контента на веб-страницах, в частности, так работают плагины, позволяющие просматривать видео на веб-страницах.

Если веб-сайт пытается заставить браузер загрузить и исполнить некий код, обычно веб-браузер выводит соответствующее предупреждение и просит пользователя подтвердить это действие, поскольку такой код может оказаться вредоносным и нарушить безопасность компьютера пользователя. Злоумышленники часто пытаются взломать веб-сайты, чтобы разместить на них вредоносный код и использовать взломанные сайты в качестве платформы для совершения атак на их посетителей. Контрмерой против таких атак является настройка веб-браузера на высокий уровень безопасности, либо полное отключение выполнения различных скриптов и активных веб-компонентов.

В следующих разделах рассматриваются некоторые из распространенных видов мобильного кода.

16.1. Java-апплеты

Java – это объектно-ориентированный независимый от платформы язык программирования. Этот язык может использоваться для написания как полноценных программ, так и небольших скриптов, называемых **апплетами** (applet), которые выполняются в браузере пользователя.

Другие языки компилируются в объектный код для конкретной операционной системы и процессора. Поэтому, например, скомпилированное для Windows приложение не может работать на Macintosh. В отличие от них, Java не зависит от платформы, поскольку он создает промежуточный код (bytecode – байт-код), который не зависит от конкретного процессора. Виртуальная машина Java (JVM – Java Virtual Machine) затем конвертирует байт-код в машинный код, который понимает процессор на данной конкретной системе. Давайте вкратце рассмотрим процесс создания и выполнения Java-апплетов:

1. Программист создает Java-апплет и выполняет его компиляцию.
2. Компилятор Java преобразует исходный код в байт-код (не зависящий от конкретного процессора).
3. Пользователь загружает Java-апплет.
4. JVM конвертирует байт-код в машинный код (для соответствующего процессора, установленного на компьютере пользователя).
5. Апплет запускается при обращении к нему.

Для запуска апплета, JVM создает виртуальную машину в рамках пользовательской среды, называемую **песочницей** (sandbox). Эта виртуальная машина является замкнутой средой, в которой апплет выполняет свои действия. Апплеты обычно отправляются по запросам от веб-страниц, поэтому апплет выполняется сразу, как только он приходит. Такой апплет может выполнять вредоносную деятельность намеренно или случайно, если разработчик апплета сделал что-то неправильно. Поэтому песочница строго ограничивает доступ апплета к любым системным ресурсам. JVM является посредником между апплетом и ресурсами системы, перехватывая, проверяя и выполняя запросы апплета к системным ресурсам и оставляя при этом сам апплет внутри песочницы. Компоненты этого процесса показаны на Рисунке 9-19.

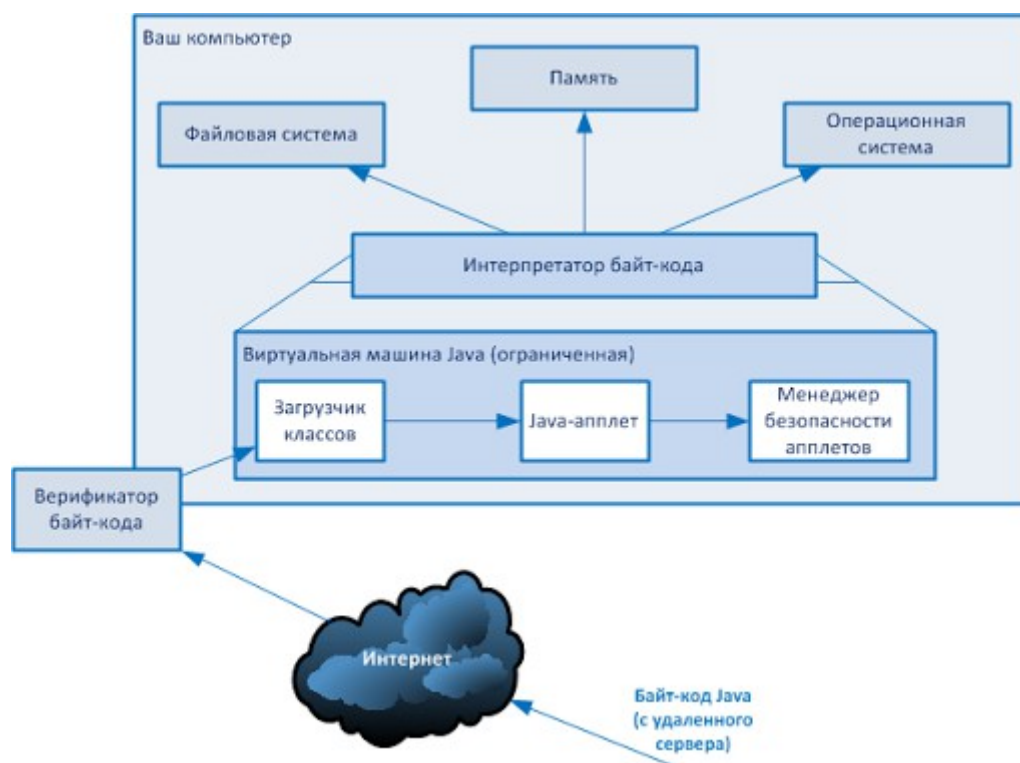


Рисунок 9-19. Модель безопасности Java

ПРИМЕЧАНИЕ. Язык Java сам предоставляет защитные механизмы, такие как сбор мусора, управление памятью, проверка использования адресов, а также содержит компонент, который проверяет соблюдение заранее определенных правил.

Настройки браузера. В отношении Java-апплетов и действий, которые они выполняют, могут быть установлены ограничения и функции контроля при помощи специальных настроек браузера. Эти настройки не влияют на полноценные приложения Java, используемые за пределами окна браузера.

Однако плохие парни выяснили, как обойти эти ограничения песочницы. Программисты выяснили, как создавать апплеты, позволяющие коду получать доступ к жестким дискам и системным ресурсам, которые должны быть защищены схемой безопасности Java. Эти методы могут использоваться при создании вредоносного кода, который может нарушить безопасность пользовательских систем.

16.2. Элементы управления ActiveX

ActiveX – это технология Microsoft, состоящая из набора технологий объектно-ориентированного программирования и инструментов, основанных на COM и DCOM. Программист использует эти инструменты для создания элементов управления (компонентов) ActiveX, которые являются самодостаточными программами, аналогично Java-апплетам. Элементы управления ActiveX могут повторно использоваться различными приложениями в рамках одной системы, или различными системами в сети. Эти элементы управления могут быть загружены с веб-сайтов для добавления дополнительной функциональности (например, для просмотра анимации на веб-странице), но кроме этого они являются компонентами операционной системы Windows (динамически подключаемыми библиотеками (DLL)) и выполняют обычные задачи операционной системы.

Для обеспечения безопасности технологии ActiveX не применяется хранение элементов управления ActiveX в безопасных местах, вместо этого выполняется информирование пользователя об источнике получения такого элемента. Получив уведомление, пользователь решает, следует ли доверять элементу из этого источника или нет.

Технология ActiveX обеспечивает различные уровни безопасности и аутентификацию, позволяя пользователям управлять безопасностью загружаемых элементов управления

ActiveX. В отличие от Java-апплетов, элементы управления ActiveX загружаются на жесткий диск пользователя, если он решает добавить функциональность, реализуемую элементом управления. Это означает, что элемент управления ActiveX имеет гораздо более широкий доступ к системе пользователя по сравнению с Java-апплетом.

Настройка уровня безопасности в браузере пользователя определяет, может ли элемент управления ActiveX загружаться автоматически, либо пользователь должен сначала подтвердить такое действие после получения соответствующего предупреждения. Уровень безопасности настраивается пользователем с помощью настроек браузера. По мере повышения уровня безопасности, также увеличивается уровень чувствительности браузера, в отношении подписанных и неподписанных компонентов, элементов управления, а также инициализации скриптов ActiveX.

Основное различие между Java-апплетами и элементами управления ActiveX в подходах к обеспечению безопасности заключается в том, что Java выполняет код апплетов в песочнице, ограничивая, таким образом, доступ кода к системным ресурсам компьютера пользователя, а ActiveX использует технологию Authenticode, основанную на цифровых сертификатах и доверенных центрах сертификации (вопросы, связанные с цифровыми подписями, сертификатами и центрами сертификации подробно описаны в Домене 06). Хотя обе эти технологии являются хорошими и интересными, им присущи определенные недостатки. Java не может гарантировано обеспечить сохранение всего кода в песочнице, что может стать причиной различных видов нарушения безопасности. Этим пользуются разработчики вредоносного программного обеспечения. В то же время ActiveX в действительности не всегда обеспечивает безопасность, поскольку частое появление предупреждающих сообщений раздражает пользователей, большинство пользователей не понимают организацию работы этой технологии и связанные с ней риски. Все это приводит к тому, что они нажимают кнопку «Ok», даже не читая текст предупреждения.

16.3. Вредоносное программное обеспечение

Существует несколько видов вредоносного кода (malicious code) или вредоносного программного обеспечения (malware), такого как вирусы, черви, троянские кони и логические бомбы. Как правило, они ведут себя тихо, пока не будут активированы событием, инициированным пользователем или системой. Они могут распространяться по электронной почте, через совместно используемые носители информации (сменные диски, флеш-накопители), при совместном использовании документов и програм, либо при загрузке файлов из сети Интернет. Кроме того, они могут быть умышленно установлены злоумышленником.

Соблюдение простого правила – не открывать вложения в сообщения электронной почты, полученные от неизвестных отправителей, является одним из лучших способов борьбы с вредоносным кодом. Однако современные вирусы и черви научились, заражая компьютер, использовать сохраненные на нем адресные книги для отправки вредоносных сообщений от имени владельца зараженного компьютера, поэтому это правило уже не гарантирует защиту системы от вредоносного кода. В таком случае, отправляемые зараженные сообщения выглядят достаточно правдоподобно, поскольку они отправляются от имени человека, известного получателю. А так как получатель знаком с отправителем, вполне вероятно, что он откроет сообщение и дважды щелкнет мышью на вложении... Теперь и его компьютер заражен и рассылает вредоносные сообщения от его имени людям, указанным в его адресной книге.

Для противодействия вирусам, на компьютере должно быть установлено антивирусное программное обеспечение, которое будет выявлять вирусы на основе известных сигнатур, а также программная система выявления вторжений на уровне узла, которая может контролировать действия работающего на компьютере программного обеспечения и выявлять подозрительное поведение (например, доступ к критичным файлам, внесение

изменений в системный реестр и т.п.), что также поможет выявить вредоносный код.

Вредоносный код может быть обнаружен по следующим косвенным признакам:

- Увеличение размера файла
- Неожиданно большое количество обращений к диску
- Изменение штампа времени последнего изменения файла
- Резкое сокращение объема свободного пространства на жестком диске
- Неожидаемые и странные действия приложений
- Резкое увеличение сетевой активности

В следующем разделе мы кратко рассмотрим несколько видов вредоносного кода.

Вирусы

Вирус (virus) представляет собой небольшую программу, заражающую другие программы. Одной из основных функций вируса является самовоспроизведение («размножение»), а это требует наличия приложения-носителя. Иными словами, вирусы не могут «размножаться» самостоятельно. Вирус заражает файлы, вставляя или добавляя свою копию в каждый файл определенного типа. Другой функцией вируса является собственно вредоносное действие, которым может быть нарушение работы компьютера пользователя (например, путем удаления системных файлов), отображения графических изображений, внесение изменений в настройки системы пользователя и т. п.

Макросы – это программы, написанные на специальных языках, таких как Visual Basic или VBScript, часто используемых при работе с продуктами Microsoft Office. Макрос автоматизирует определенные повторяющиеся задачи, которые иначе пользователям пришлось бы выполнять вручную. Пользователи могут создать макрос, который будет выполнять ряд действий или повседневных рутинных задач просто по нажатию кнопки, избавляя пользователя от необходимости выполнять каждую из этих задач по отдельности вручную. Макрос может быть разработан для выполнения полезных действий, но он может быть создан и в качестве вредоносной программы. **Макро-вирус** (macro virus) – это вирус, написанный на одном из таких макро-языков и являющийся независимым от платформы. Макро-вирусы заражают файлы шаблонов и документов, используя их для своего «размножения». Макро-вирусы встречаются достаточно часто, поскольку их совсем несложно написать, а офисные продукты используются очень широко.

Некоторые вирусы заражают загрузочный сектор жесткого диска компьютера, перенося оригинальные данные загрузочного сектора в другое место на диске, либо просто перезаписывая поверх них новую информацию. Такие вирусы называются **загрузочными вирусами** (boot sector virus). Некоторые загрузочные вирусы размещают непосредственно в загрузочном секторе части своего кода, которые могут запустить вирус, а остальные части кода размещаются в секторах жесткого диска, которые они помечают как "плохие". Такие секторы операционная система и приложения не будут пытаться использовать и злоумышленник может не опасаться, что они будут перезаписаны.

Другие виды вирусов сжимают исполняемые файлы на компьютере и добавляют сжатые файлы к своему вредоносному коду, используя предоставленные пользователю разрешения (**сжимающие вирусы** – compression virus). Если пользователь запускает зараженный исполняемый файл, вирус распаковывает оригинальный исполняемый файл, сохраняет его во временный файл и запускает на выполнение, а сам параллельно делает свои грязные дела.

Вирус-невидимка (stealth virus) скрывает изменения, которые он произвел в файлах или загрузочной записи. Это может реализовано путем мониторинга вызовов системных функций, используемых для чтения файлов или секторов и выдачи результатов. При этом,

когда антивирусная программа пытается прочитать зараженный файл или сектор, ей предоставляется первоначальное содержимое неинфицированного файла вместо реального содержимого инфицированного файла. Также вирус может попытаться скрыть свое присутствие, временно переместив свой код в другое место, пока антивирусная программа выполняет процедуру сканирования.

Таким образом, вирус-невидимка – это вирус, который скрывает свои следы после заражения системы. Он вносит определенные изменения, чтобы компьютер выглядел точно так же, как раньше. К примеру, вирус может показывать первоначальный размер файла, который он инфицировал, а не новый, увеличившийся размер, пытаясь скрыть свое присутствие от пользователя и антивирусных средств.

Полиморфный вирус (polymorphic virus) создает отличающиеся друг от друга, но при этом полнофункциональные копии своего кода. Он делает это, чтобы перехитрить антивирусный сканер. Даже если сканер сможет обнаружить и заблокировать 1-2 копии вируса, остальные экземпляры могут остаться в системе и продолжить свою работу.

Полиморфный вирус может использовать различные методы зашифрования, требующие различных процедур расшифрования. Чтобы обнаружить такой вирус, антивирусный сканер должен проверять каждый файл с использованием каждого из возможных методов расшифрования, чтобы выявить все копии этого вируса.

Такие вирусы могут изменять последовательность своих команд, включая «шум» – фиктивные команды вперемешку с реальными командами. Также они могут использовать для изменения последовательности своих команд алгоритмы мутации и генератор случайных чисел, пытаясь защитить себя от обнаружения. Полиморфный вирус обладает способностью изменять свой собственный код, что позволяет ему иметь сотни или даже тысячи различных вариантов. Это может привести к тому, что антивирусный сканер не сможет найти вирус и оставит его в покое.

Составной вирус (multipart virus) заражает как загрузочный сектор жесткого диска, так и исполняемые файлы. В случае запуска такого вируса, он сначала заражает загрузочный сектор диска, а затем приступает к заражению всей системы.

Самоискажающий вирус (self-garbling virus) пытается скрыться от антивирусного программного обеспечения, искажая свой собственный код. По мере распространения вируса, он изменяет способ форматирования своего кода. Небольшая часть кода вируса при его активации выполняет декодирование искаженной части кода.

Мем-вирусы (Meme virus) фактически не являются компьютерными вирусами – это разновидность сообщений электронной почты, которые постоянно пересылаются по всему Интернету. Они могут быть цепочкой писем, поддельным предупреждением о почтовом вирусе, сообщениями религиозного характера или сообщениями финансовых пирамид. Они пересылаются людьми, а не программами, они могут расходовать трафик и сеять панику. Некоторые письма предупреждают об опасных (но не существующих в действительности) вирусах. Прочитав такое сообщение, люди верят ему и считают своим долгом переслать их своим друзьям и знакомым, чтобы рассказать им об этом, хотя реально они были обмануты и фактически распространяли мем-вирус.

Скриптовые вирусы (Script virus) весьма популярны и опасны в последние годы. Скрипты – это файлы, выполняемые интерпретатором, например, Microsoft Windows Script Host, интерпретирующим различные виды скриптовых языков. Использование скриптов позволило сделать веб-сайты более динамичными и интерактивными. Чаще всего для этих целей используются Visual Basic (VBScript) и Java (JScript), и другие скриптовые языки, встроенные в HTML. Когда веб-страница, содержащая такие скрипты, запрашивается веб-браузером, эти скрипты исполняются. Если оказывается, что скрипты были вредоносными, последствия этого могут быть очень плохими. Такой скрипт может выполнять, например,

рассылку вредоносного кода в сообщениях электронной почты, адресованных всем знакомым пользователя инфицированного компьютера, записанным в его адресной книге. Также, такой скрипт может удалить или внести изменения в критичные файлы. Применение скриптов является еще одним вектором заражения, используемым вирусописателями для выполнения своих грязных дел.

ПРИМЕЧАНИЕ. Одним из наиболее известных вирусов, причинивших огромный ущерб, является LoveLetter. Он был написан на VBScript.

Другим видом вирусов является **туннелирующий вирус** (tunneling virus), который пытается установить себя ниже антивирусной программы (на уровне системы). Он перехватывает запросы к функциям операционной системы. Когда антивирусная программа выполняет проверку критичных файлов, их размеров, дат модификации и т.д., обращаясь за этой информацией к функциям операционной системы, вирус перехватывает этот запрос и отвечает на него самостоятельно, предоставляя информацию, которая свидетельствует о том, что все хорошо и никаких признаков заражения нет.

ПРИМЕЧАНИЕ. Для тестирования антивирусного программного обеспечения используется тест EICAR. Этот тест выполняется с помощью специального СОМ-файла, признаваемая вредоносным всеми антивирусными продуктами (хотя в действительности никакой угрозы он не представляет и при запуске просто выводит на экран текстовую строку). В базах любого антивирусного продукта есть сигнатура файла EICAR.com. Вы можете провести этот тест самостоятельно, чтобы проверить реакцию на вирус используемых в вашей компании антивирусных продуктов. Для этого после завершения настройки антивирусного программного обеспечения, просто запишите такой файл в систему (его можно создать с помощью Блокнота Windows).

Компоненты вредоносного программного обеспечения. Ниже перечислены шесть основных элементов вредоносных программ, хотя не обязательно, что любая из них должна обладать всеми этими элементами.

- **Установка ("заражение").** Инсталлирует себя в систему жертвы.
- **Скрытие.** Использует методы, позволяющих избежать обнаружения.
- **Самоуничтожение.** Удаляет собственный код после выполнения вредоносных действий.
- **Репликация ("размножение").** Создает свои копии, распространяясь на системы других жертв.
- **Триггер.** Использует события, чтобы инициировать исполнение своих вредоносных действий.
- **Полезная нагрузка (payload).** Выполняет определенную функцию, для которой был предназначен вредоносный код (т.е. удаляет файлы, устанавливает бэкдор и т.п.)

Черви

Черви (worm) отличаются от вирусов тем, что они могут воспроизводить себя самостоятельно, не используя для этого какое-либо приложение на зараженном компьютере, и являются автономными программами. Червь может распространять свои копии через электронную почту, через веб-сайты и т.д. В настоящее время определения червей и вирусов все больше сливаются, различия между ними становятся все более размытыми. Червь ILOVEYOU был одной из первых таких программ, он использовал для своего распространения программы Outlook и Outlook Express. При запуске пользователем вложения в сообщении электронной почты, автоматически порождалось несколько процессов. Червь отправлял свои копии на все адреса, найденные в адресной книге жертвы. Некоторые файлы на жестком диске удалялись и заменялись на другие. При их открытии, червь производил повторную рассылку своих копий.

Троянские программы

Троянская программа (Trojan horse) – это программа, которая скрывается под другой программой. Например, троянская программа может называться Notepad.exe и иметь такую же иконку, как обычная программа «Блокнот». Однако при запуске поддельной Notepad.exe,

эта программа может удалить системные файлы. Троянские программы выполняют полезные функции в дополнение к вредоносной функциональности, выполняющейся в фоновом режиме. Троянская программа, названная «Notepad.exe» может также запустить и обычную программу «Блокнот» для пользователя, однако в фоновом режиме она будет удалять файлы или выполнять иные вредоносные действия. Для противодействия троянским программам может использоваться система IDS уровня узла, которая может быть настроена для контроля определенных файлов и выявления фактов увеличения их размеров, что часто является признаком троянской программы. Если оригинальный файл Notepad.exe имеет размер 50KB и вдруг его размер вырос до 2 MB, это может означать, что эта программа заражена трояном.

Трояны удаленного доступа (RAT – Remote Access Trojan) – это вредоносные программы, которые при работе на системе жертвы, позволяют злоумышленнику получить удаленный доступ к этой системе. Они имитируют функциональность обычных программ удаленного доступа, применяемых для удаленного администрирования, однако они используются во вредоносных целях. Трояны удаленного доступа обычно разрабатывают таким образом, чтобы они могли незаметно устанавливаться в систему и также незаметно работать в ней. Обычно их скрывают в различном мобильном коде, например, в Java-апплетах или элементах управления ActiveX, которые загружаются с веб-сайтов.

Хакерам доступно множество подобных программ (Back Orifice, SubSeven, Netbus и др.). Как только такая программа запускается на системе жертвы, злоумышленник может скачать или загрузить на нее файлы, передать ей команды, установить на нее программное обеспечение (например, для подключения к ботсети) и использовать зараженную систему по своему усмотрению.

Логические бомбы

Логическая бомба (logic bomb) запускает программу или строку кода, когда происходит определенное событие или наступает определенная дата и время. Например, логическая бомба может быть настроена на событие запуска пользователем программного обеспечения для доступа к своему банковскому счету. При этом логическая бомба запускает программу, которая выполняет копирование реквизитов доступа к счету пользователя. Другим событием, на которое может быть настроена логическая бомба, является подключение пользователя к сети Интернет. При наступлении этого события она может послать злоумышленнику сообщение через Интернет, говоря ему, что пользователь подключен к сети и может быть атакован.

Ботсети

Слово «бот» – это сокращение от слова «робот». Бот представляет собой фрагмент кода, который выполняет некоторую функциональность для своего хозяина, являющегося автором этого кода. **Боты** (bot) являются разновидностью вредоносных программ, они установлены на тысячах компьютеров. Компьютер, на котором установлен бот, называют *зомби* (zombie). Бот получает команды от своего хозяина и заставляет зараженный компьютер выполнять их. Такими командами может быть рассылка спама, вирусов или проведение атак. Злоумышленник предпочитает выполнять такие действия с использованием ботов, а не своего компьютера, поскольку это позволяет ему избежать обнаружения и идентификации.

Совокупность скомпрометированных злоумышленником зомби-компьютеров, на которых установлены боты, называется **ботсетью** (botnet). Для создания ботсети хакеры взламывают тысячи систем, рассылая вредоносный код множеством различных методов: в виде вложений в сообщения электронной почты, через скомпрометированные веб-сайты, с помощью рассылки ссылок на вредоносные сайты, вложенных в сообщения электронной почты и т.д. В случае успешной установки на компьютере пользователя, вредоносный код направляет злоумышленнику сообщение о том, что система была взломана и теперь доступна злоумышленнику, который может использовать ее по своему желанию. Например, он может

использовать созданную ботсеть для проведения мощной DDoS-атаки или сдавать ее в аренду спамерам. При этом большинство компьютеров, входящих в ботсеть, являются домашними компьютерами ничего не подозревающих пользователей.

Хозяин этой ботсети управляет входящими в нее системами удаленно, как правило, посредством протокола IRC (Internet Relay Chat).

Основные шаги создания и использования ботсетей приведены ниже:

1. Хакер различными способами направляет потенциальным жертвам вредоносный код, который содержит в себе программное обеспечение бота.
2. После успешной установки на системе жертвы, бот устанавливает контакт с управляющим сервером ботсети, связываясь с ним через IRC или специальный веб-сервер, в соответствии с тем, что указано в его коде. После этого управляющий сервер берет на себя управление новым ботом.
3. Спамер платит хакеру за использование систем его ботсети, хакер передает на управляющий сервер соответствующие команды, а управляющий сервер, в свою очередь, дает команду всем зараженным системам, входящим в ботсеть, рассылать спам.

Спамеры используют этот метод потому, что это значительно повышает вероятность достижения их сообщениями получателей, в обход установленных у них спам-фильтров, т.к. такие сообщения будут отправляться не с одного адреса, который быстро будет заблокирован или добавлен во все «черные списки», а со множества реальных адресов владельцев взломанных компьютеров.

Для создания ботсети, ее будущий хозяин либо все делает сам, либо оплачивает хакерам разработку и распространение вредоносных программ для заражения систем, которые станут частью его ботсети. А потом к хозяину ботсети будут обращаться и платить ему те, кто хочет рассказать вам о своих новых продуктах, а также те, кому нужно провести атаку на конкурентов, своровать личные данные или пароли пользователей и многие другие.

16.4. Антивирусное программное обеспечение

Традиционное антивирусное программное обеспечение использует сигнатуры для обнаружения вредоносного кода. Сигнатуры – это «отпечатки пальцев» вредоносного кода, созданные производителем антивирусного программного обеспечения. Сигнатура представляет собой фрагменты кода, извлеченные из самого вируса. Антивирусная программа сканирует файлы, сообщения электронной почты и другие данные, проходящие через определенные протоколы, и сравнивает их со своей базой вирусных сигнатур. При выявлении совпадений, антивирусная программа выполняет заранее настроенное действие, которым может быть отправка зараженного файла в карантин, попытка «вылечить» файл (удалить вирус), отображение окна с предупреждением для пользователя и/или запись события в журнал регистрации событий.

Выявление вредоносного кода на основе сигнатур – это эффективный способ обнаружения вредоносного программного обеспечения, однако при этом существуют определенные задержки в части реагирования на новые угрозы. После первого обнаружения вируса, производитель антивируса должен изучить этот вирус, разработать и протестировать новые сигнатуры, выпустить обновление базы сигнатур, а все пользователи должны загрузить это обновление. Если вредоносный код просто рассылает ваши фотографии всем вашим друзьям, такая задержка не столь критична. Однако если вредоносная программа похожа на червя Slammer, ущерб от такой задержки может быть катастрофическим.

ПРИМЕЧАНИЕ. Червь Slammer появился в 2003 году. Он использовал уязвимость в СУБД Microsoft SQL Server 2000, позволяющую провести атаку переполнения буфера и вызвать отказ в обслуживании. По некоторым оценкам Slammer нанес ущерб на сумму свыше 1 млрд. долларов.

Поскольку новые вредоносные программы создаются ежедневно, производителям антивирусного программного обеспечения трудно не отставать. Технология использования вирусных сигнатур позволяет обнаруживать вирусы, которые уже были выявлены, и для которых была создана сигнатура. Но в связи с тем, что вирусописатели очень плодовиты, а многие вирусы могут изменять свой код, очень важно, чтобы антивирусное программное обеспечение имело и другие механизмы, позволяющие обнаружить вредоносный код.

Другим методом, который используют почти все антивирусные программные продукты, является обнаружение вредоносного кода на основе *эвристического анализа* (heuristic detection). Этот метод анализирует общую структуру вредоносного кода, оценивает выполняемые кодом инструкции и алгоритмы, изучает типы данных, используемые вредоносной программой. Таким образом, он собирает большой объем информации о фрагменте кода и оценивает вероятность того, что он имеет вредоносный характер. Он использует некий «счетчик подозрительности», который увеличивается по мере того, как антивирусная программа находит в нем новые потенциально опасные (подозрительные) свойства. При достижении заранее определенного порогового значения, код считается опасным, и антивирусная программа инициирует соответствующие защитные механизмы. Это позволяет антивирусному программному обеспечению распознавать неизвестные вредоносные программы, а не только полагаться на сигнатуры.

Рассмотрим следующую аналогию. Иван – полицейский, он работает, чтобы поймать плохих парней и запереть их. Если Иван собирается использовать метод сигнатур, он сравнивает стопки фотографий с каждым человеком, которого он видит на улице. Когда он видит совпадение, он быстро ловит плохого парня и сажает его в свою патрульную машину. Если он собирается использовать эвристический метод, он следит за подозрительными действиями. Например, если он видит человека в лыжной маске, стоящего перед входом в банк, он оценивает вероятность того, что это грабитель, а не просто замерзший парень, выпрашивающий мелочь у посетителей банка.

ПРИМЕЧАНИЕ. Бездисковые рабочие станции так же уязвимы для вирусов, несмотря на отсутствие у них жесткого диска и полноценной операционной системы. Они могут быть заражены вирусами, которые загружаются и живут в памяти. Такие системы могут быть перезагружены дистанционно (удаленная перезагрузка), чтобы очистить память и вернуть ее в исходное состояние, т.е. вирус кратковременно живет в такой системе.

Некоторые антивирусные продукты создают искусственную среду, называемую виртуальной машиной или песочницей, и позволяют некоторой части подозрительного кода выполняться в защищенной среде. Это дает антивирусной программе возможность увидеть код в действии, что дает гораздо больше информации для принятия решения, является ли он вредоносным или нет.

ПРИМЕЧАНИЕ. Виртуальную машину или песочницу иногда называют *буфером эмуляции* (emulation buffer). Это то же самое, что защищенный сегмент памяти, поэтому даже если код действительно окажется вредоносным, система все равно останется в безопасности.

Анализ информации о части кода называется *статическим анализом*, если выполняется запуск части кода на виртуальной машине, это называется *динамическим анализом*. Оба этих метода считаются эвристическими методами обнаружения.

Вакцинация. Другой подход, который использовали некоторые антивирусные программы, называется *вакцинацией* (immunization). Продукты с этой функциональностью вносили изменения в файлы и области диска, чтобы они выглядели так, как будто уже были инфицированы. При этом вирус может решить, что файл (диск) уже заражен и не будет вносить никаких дополнительных изменений, перейдя к следующему файлу.

Программа вакцинации, как правило, нацелена на конкретный вирус, поскольку каждый из них по-разному проверяет факт заражения и ищет в файле (на диске) разные данные (сигнатуры). Однако число вирусов и другого вредоносного программного обеспечения постоянно растет, растет и количество файлов, которые необходимо защищать, поэтому такой подход в настоящее время не применим на практике в большинстве случаев, и производители антивирусов больше его

не используют.

В настоящее время, даже с учетом всех этих сложных и эффективных подходов, нет стопроцентной гарантии эффективности антивирусных средств, поскольку вирусописатели очень хитры. Это постоянная игра в кошки-мышки, которая продолжается каждый день. Антивирусная индустрия находит новый способ обнаружения вредоносных программ, а вирусописатели на следующей неделе находят, как обойти этот новый способ. Это заставляет производителей антивирусных средств постоянно увеличивать интеллектуальность своих продуктов, а пользователям приходится ежегодно покупать их новые версии.

Следующим этапом эволюции антивирусного программного обеспечения называют **поведенческие блокираторы** (behavior blocker). Антивирусное программное обеспечение, выполняющее блокировку на основе поведения, фактически позволяет подозрительному коду выполняться в незащищенной операционной системе и следит за его взаимодействием с операционной системой, обращая внимание на подозрительные действия. В частности, антивирусное программное обеспечение следит за следующими видами действий:

- Запись в автоматически загружаемые при запуске системы файлы или в разделы автозапуска в системном реестре
- Открытие, удаление или изменение файлов
- Включение скриптов в сообщения электронной почты для отправки исполняемого кода
- Подключение к сетевым ресурсам или общим папкам
- Изменение логики исполняемого кода
- Создание или изменение макросов и скриптов
- Форматирование жесткого диска или запись в загрузочный сектор

Если антивирусная программа выявляет некоторые из этих потенциально опасных действий, она может принудительно завершить такую программу и сообщить об этом пользователю. Новое поколение поведенческих блокираторов в действительности анализирует последовательность выполнения таких действий, прежде чем решить, что система заражена (поведенческие блокираторы первого поколения срабатывали просто на отдельные действия, что приводило к большому числу ложных срабатываний). Современное антивирусное программное обеспечение может перехватывать выполнение опасных частей кода и не позволяет им взаимодействовать с другими запущенными процессами. Также они могут обнаруживать руткиты. Некоторые из таких антивирусных программ позволяют выполнить «откат» системы до состояния, в котором она была перед заражением, «стирая» все изменения, выполненные вредоносным кодом.

Казалось бы, что поведенческие блокираторы могут полностью решить все проблемы, связанные с вредоносным кодом, однако у них есть один недостаток, который требует выполнения такого мониторинга вредоносного кода в режиме реального времени, в противном случае система все-таки может быть заражена. К тому же постоянный мониторинг требует большого количества системных ресурсов...

ПРИМЕЧАНИЕ. Эвристический анализ и блокирование на основе поведения считаются проактивными методами, они могут обнаруживать новые вредоносные программы, иногда называемые атаками «нулевого дня». Обнаружение вредоносного кода на основе сигнатур не может выявить новые вредоносные программы.

Большинство антивирусных программ, используют сочетание всех этих технологий, чтобы обеспечить максимальную защиту, насколько это возможно. Отдельные решения по противодействию вредоносному программному обеспечению показаны на Рисунке 9-20.

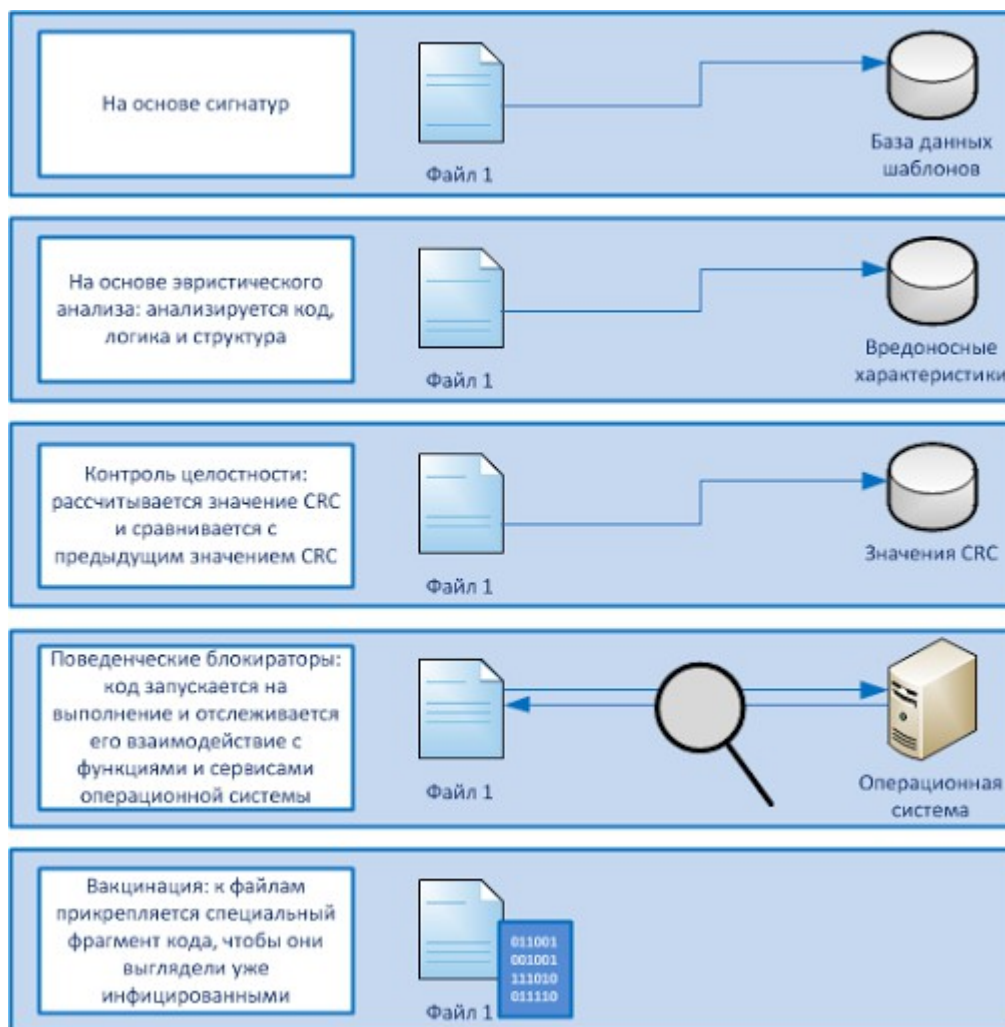


Рисунок 9-20. Производители антивирусного программного обеспечения используют различные методы обнаружения вредоносного кода

16.5. Выявление спама

Все мы очень устали от сообщений электронной почты, которые предлагают нам купить что-нибудь ненужное. Такие письма называются **спамом** (spam) – это нежелательные сообщения электронной почты. Спам не только отвлекает его получателей от их дел, но потребляет значительную пропускную способность сети, а также может являться источником распространения вредоносных программ. Многие компании используют спам-фильтры на своих почтовых серверах, а пользователи могут настроить правила фильтрации спама в своих почтовых клиентах. Но спамеры, также как и вирусописатели, постоянно выдумывают новые хитроумные способы обхода спам-фильтров.

Эффективное распознавание спама стало настоящей наукой. Один из используемых методов называется **Байесовской фильтрацией** (Bayesian filtering). Много лет назад, господин по имени Томас Байес (математик) разработал эффективный способ предсказания вероятности происхождения каких-либо событий с помощью математики. Теорема Байеса позволяет определить вероятность того, что произошло какое-либо событие при наличии лишь косвенных тому подтверждений (данных), которые могут быть неточны. Концептуально это не так уж трудно понять. Если вы трижды ударились головой о кирпичную стену и каждый раз падали, вы можете сделать вывод, что и дальнейшие попытки приведут к тем же болезненным результатам. Более интересно, когда эта логика применяется к действиям, содержащим намного больше переменных. Например, как работает спам-фильтр, который не пропускает к вам письма с предложением купить виаргу, но при этом не препятствует доставке почты от вашего друга, который очень интересуется этим препаратом и пишет вам

сообщения о его свойствах и воздействии на организм? Фильтр Байеса применяет статистическое моделирование к словам, из которых состоят сообщения электронной почты. Над этими словами выполняются математические формулы, позволяющие в полной мере понять их отношение друг к другу. Фильтр Байеса выполняет частотный анализ каждого слова, а затем оценивает сообщение как целое, чтобы определить, спам это или нет.

Такой фильтр не просто ищет слова «Виагра», «секс» и т.п., он смотрит на то, как часто используются эти слова, и в каком порядке, чтобы установить, является ли сообщение спамом. К сожалению, спамеры знают, как работают такие фильтры, и манипулируют словами в строке темы и теле сообщения, чтобы попытаться обмануть спам-фильтр. Именно поэтому вы можете получать спам-сообщения с ошибками или словами, в которых используются символы вместо букв. Спамеры очень заинтересованы в том, чтобы вы получали их сообщения, потому что они зарабатывают на этом большие деньги.

16.6. Противодействие вредоносному коду

Защита компаний от большого списка различных вредоносных программ требует большего, чем просто антивирусное программное обеспечение. Как и с другими компонентами программы безопасности, требуется внедрить и поддерживать некоторые дополнительные административные, физические и технические защитные меры и средства.

У компании должна быть отдельная антивирусная политика, либо вопросы антивирусной защиты должны учитываться в общей политике безопасности. Должны быть разработаны стандарты, определяющие необходимые для использования в компании виды антивирусного и анти-шпионского программного обеспечения, а также основные параметры их конфигурации.

Сведения о вирусных атаках, используемых средствах антивирусной защиты, а также об ожидаемом от пользователей поведении должны быть предусмотрены в программе повышения осведомленности. Каждый пользователь должен знать, что он должен делать и куда обращаться, если на его компьютере будет обнаружен вирус. В стандарте должны быть рассмотрены все вопросы, касающиеся действий пользователя, связанных с вредоносным кодом, должно быть указано, что должен сделать пользователь и что делать ему запрещается. В частности, стандарт должен содержать следующие вопросы:

- На каждую рабочую станцию, сервер, коммуникатор, смартфон должно быть установлено антивирусное программное обеспечение.
- Для каждого из этих устройств должен быть реализован способ автоматического обновления антивирусных сигнатур, который должен быть включен и настроен на каждом устройстве.
- У пользователя не должно быть возможности отключить антивирусное программное обеспечение.
- Должен быть заранее разработан и спланирован процесс удаления вирусов, должно быть определено и назначено контактное лицо, на случай выявления вредоносного кода.
- Все внешние диски (USB-накопители и т.п.) должны сканироваться автоматически.
- Должны сканироваться файлы резервных копий.
- Должен проводиться ежегодный пересмотр антивирусных политик и процедур.
- Используемое антивирусное программное обеспечение должно обеспечивать защиту от загрузочных вирусов.
- Антивирусное сканирование должно независимо выполняться на шлюзе и на каждом отдельном устройстве.

- Антивирусное сканирование должно выполняться автоматически по расписанию. Не нужно рассчитывать на то, что пользователи будут запускать сканирование вручную.
- Критичные системы должны быть физически защищены таким образом, чтобы локальная установка на них вредоносного программного обеспечения была невозможна.

Поскольку вредоносное программное обеспечение может нанести многомиллионный ущерб (в виде операционных расходов, потери производительности), многие компании устанавливают антивирусные решения на всех точках входа в сеть. Антивирусный сканер может быть интегрирован в программное обеспечение почтового сервера, прокси-сервера или межсетевого экрана. Такой антивирусный сканер проверяет весь входящий трафик на наличие в нем вредоносного кода, чтобы обнаружить и остановить его заранее, еще до того, как он попадет во внутреннюю сеть. Продукты, реализующие такую функциональность, могут сканировать трафик SMTP, HTTP, FTP, а также, возможно, и других протоколов. Но важно понимать, что такой продукт следит только за одним или двумя протоколами, а не за всем входящим трафиком. Это является одной из причин, по которой на каждом сервере и рабочей станции также должно быть установлено антивирусное программное обеспечение.

17. Управление патчами

Производители часто слишком торопятся выпустить продукт на рынок. Это приводит к тому, что в таком продукте обязательно остаются некоторые «дыры». Такие «продукты» могут привести к нарушению безопасности компании, прерыванию ее функционирования и нанести катастрофический ущерб. Для примера, выберите свою любимую СУБД и посмотрите на сайте ее производителя список исправлений (патчей), а затем патчей-для-патчей, которые были выпущены в течение последних пяти лет. Спросите у вашего администратора баз данных (DBA), как часто при установке обновлений перестают работать некоторые специфические (недокументированные) функции, необходимые для ваших приложений. Часто оказывается, что именно эти функции обожают разработчики приложения, и, конечно же, без них приложение работать не может.

Вы можете принять меры для снижения влияния патчей. Наилучший способ состоит в разработке, внедрении и повышении уровня зрелости *процесса управления патчами* (patch management process), целью которого является обеспечение внедрения патчей и обновлений в промышленную среду управляемым способом и наличие плана для их «отката». Эффективный процесс управления патчами структурирован и выполняется в соответствии со следующей методологией.

17.1. Методология управления патчами

Шаг 1: Инфраструктура

Вам необходимо создать основу - *инфраструктуру* для процесса управления патчами. Это не просто физическая инфраструктура, состоящая из коммутаторов, маршрутизаторов, кабелей и всего остального, что позволяет распространять патчи, но также и всего того, что позволит выполняться самому этому процессу. Нужно разработать стратегию управления патчами, которая подойдет именно вашей компании, а затем сформировать команду, которая будет выполнять этот процесс и вести учет установки патчей в рамках всей компании. Кто должен быть включен в эту команду? Членами команды должны стать не только отдельные системные администраторы, но и представители разработчиков и/или специалисты технической поддержки систем, на которые предполагается устанавливать патчи, а также систем, работающих совместно с ними, и приложений, работающих на их основе. При этом речь идет о любом программном обеспечении, независимо от того, установлено ли оно на жесткий диск или записано в «прошивку». Любое программное обеспечение нуждается в *установке патчей*, а аппаратное обеспечение при необходимости *заменяется* или *модернизируется*. Независимо от того, что мы обновляем (программное или аппаратное

обеспечение), необходимо следовать определенному процессу, который обеспечивает безопасное внесение изменений в среду.

Шаг 2: Изучение

Нередко случается, что на систему устанавливается неправильный патч. Администратор, в спешке пытаясь остановить атаки скрипт-кидди на уязвимый сервер, не читая последнюю часть имени патча, загружает патч от другой версии программного обеспечения и устанавливает его ... полностью выводя из строя приложение. Или, что еще хуже, администратор находит патч не на официальном сайте производителя, загружает его, и, не позаботившись о проверке аутентичности и целостности файла, устанавливает. Это прекрасный способ установить в систему троян или другую вредоносную программу.

Единственный способ предотвратить такие нежелательные происшествия – сначала изучить патч, убедиться, что источник его загрузки является подлинным, а файлы патча не повреждены, выполнив проверку целостности. Многие файлы публикуются вместе с цифровым отпечатком или цифровой подписью, созданной с помощью алгоритма хэширования, например, MD5 или SHA1. Такие подписи позволяют проверить целостность файлов путем перерасчета значения хэша и сравнения результатов.

Шаг 3: Тестирование

Перед установкой патча на системы, находящиеся в промышленной эксплуатации, важно проверить отсутствие любых неожиданных последствий, к которым может привести это. Такое тестирование лучше всего проводить в тестовой среде, которая как можно точнее отражает промышленную среду. Для проведения тестирования должен разрабатываться план тестирования, который работает как сценарий, предусматривающий последовательный проход по всем известным функциям и процедурам системы. План тестирования должен предусматривать выполнение действий, имитирующих стандартную работу в системе, работающей в промышленной среде. Кроме того, любой процесс управления патчами должен вписываться в Процесс управления изменениями, работающий в компании. Установка патчей, безусловно, *изменяет* промышленную среду, а Процесс управления изменениями осуществляет контроль за такими изменениями, снижая вероятность возникновения неожиданных проблем, и предусматривая стратегию «отката» на случай, если проблемы все же возникнут.

Шаг 4: Подготовка плана "отката"

Даже если вы внимательно изучили и протестировали патч, установили его в соответствии с практикой управления изменениями, вы все равно можете столкнуться с проблемами, вызванными установкой этого патча. Единственным выходом из такой ситуации, позволяющим сохранить функционирование промышленных систем, является «откат» изменений, позволяющий отменить все произведенные при установке патча изменения и вернуть промышленную среду в состояние, в котором она находилась до установки патча. Лучшим способом минимизации последствий подобных проблем является предварительная подготовка к их возможному возникновению. Для этого целесообразно разработать план «отката», который будет описывать все шаги, необходимые для возврата системы (среды) в рабочее состояние, как перед установкой патча.

Шаг 5: Установка патча

Когда вы будете готовы к установке патча на системы, находящиеся в промышленной эксплуатации, лучшее, что вы можете сделать, это сразу установить его на самую критичную систему... не так ли? Конечно, нет! Большинство компаний по возможности применяют поэтапный подход к развертыванию патчей, начиная с *экспериментальной группы*, состоящей из наименее важных систем. По прошествии определенного периода времени, если на этих системах не возникло никаких проблем, патч применяется к следующей группе, состоящей из более критичных систем. И только в самом конце патч устанавливается на

самые критичные системы. Часто при реализации стратегии развертывания патчей, особенно когда речь идет о большом количестве систем, применяются автоматизированные скрипты или инструменты для такого развертывания. Они помогают минимизировать вероятность возникновения проблем, связанных с человеческими ошибками, последовательно выполняя на каждой системе заранее определенную последовательность действий. Наилучшим подходом к развертыванию патчей является выполнение установки патчей по расписанию в пределах заранее определенного временного окна. Это временное окно должно выбираться таким образом, чтобы оно не пересекалось со временем пиковой нагрузки на системы, но совпадало с рабочим временем сотрудников, выполняющих установку патчей, и сотрудников, осуществляющих техническую поддержку систем, на которые они устанавливаются.

Шаг 6: Журналирование, Проверка и Подготовка отчета

В заключение, нужно предусмотреть ту или иную форму журналирования событий, чтобы можно было отслеживать действия, выполнявшиеся в промышленной среде, а также собственно факты установки патчей, с указанием какие патчи, когда и где устанавливались. Вся эта информация должна записываться в журналы регистрации событий, должны быть разработаны необходимые для этого документы и стандарты, а конфигурации должны обновляться с учетом новых патчей. Для завершения процесса развертывания патча необходимо получить подтверждение, что на все системы, предназначенные для установки патча, патч действительно установлен. Соответствующую проверку можно провести вручную, обойдя все системы, либо с помощью специализированного инструмента, выполняющего автоматическое сканирование систем. Такие инструменты могут работать с использованием агентов, локально устанавливаемых на все проверяемые системы, либо они могут удаленно обращаться к этим системам, чтобы узнать, установлен ли на них патч. После успешного завершения этой проверки и сбора всех необходимых данных, должен быть подготовлен отчет, который может потребоваться в дальнейшем (например, при проведении аудиторской проверки).

17.2. Проблемы при установке патчей

Наличие в вашей компании зрелого процесса установки патчей, включающего в себя все шесть этапов, описанных выше, не гарантирует успеха. Проблемы при управлении патчами могут быть вызваны неисправностями в системах или инфраструктуре, участвующей в процессе установки патчей, или недостатками методов установки патчей. Часто сам процесс установки патча просто занимает больше времени, чем планировалось, и выходит за пределы окна времени установки патчей, в результате чего на некоторые системы патч не устанавливается и они остаются уязвимыми. Что еще хуже, обновленные системы могут оказаться несовместимыми с системами, на которые патч не был установлен, а другие системы могут полностью или частично выйти из строя после установки патча. Зачастую превышение запланированного времени установки патчей происходит из-за избыточной нагрузки на сеть, обладающую недостаточной пропускной способностью, необходимой для доставки файлов с патчами.

17.3. Лучшие практики

Хорошая практика управления патчами включает в себя определение наиболее правильного решения для конкретной компании и ее среды. Это может быть достаточно сложной задачей. Любая компания может купить самые современные средства управления патчами, но найти нужный инструмент, лучше всего подходящий именно вашей компании, и максимально эффективно использовать его – вот то, что часто вызывает проблемы.

Некоторые производители предлагают собственные подходы к организации процессов управления изменениями и контроля изменений. Многие публикуют документы, в которых рассказывают о методологии управления изменениями, тогда как другие просто выпускают

патчи в соответствии с предсказуемым графиком, а также конкретные процедуры для развертывания этих патчей.

Насколько бы вы не были уверены в эффективности вашего процесса управления патчами и плана «отката», вы всегда должны предусматривать выполнение резервного копирования систем и данных до установки патчей. Это не значит, что вы должны выполнять резервное копирование только тогда, когда собираетесь устанавливать патчи. Конечно, нет! Резервное копирование должно быть частью ежедневных процедур эксплуатации и администрирования системам компании. Тем не менее, перед установкой патчей лучше сделать внеплановую резервную копию, чтобы иметь актуальную копию на случай возникновения проблем, что существенно упростит и ускорит процесс восстановления систем.

Кроме того, хорошей практикой является поддержка в актуальном состоянии перечня всего программного и аппаратного обеспечения, используемого в компании, а также конфигурационной информации, дистрибутивов и инструкций для него. Это сложнее, чем может показаться. Поддержка библиотеки, забитой инсталляционными компакт-дисками, всевозможными накопителями, документацией и файлами обновлений, может потребовать большого объема работы, не говоря уже о месте для хранения всего этого. Однако такая библиотека имеет огромное значение, и преимущества от ее наличия почти всегда покрывают издержки на ее поддержку.

Чтобы снизить риски компрометации систем, независимо от того, установлены на них актуальные патчи или нет, вокруг этих систем должны быть установлены компенсирующие защитные механизмы, которые позволят снизить риски эксплуатации уязвимостей системы. Для этих целей выполняется укрепление серверов, при котором блокируются ненужные сервисы, максимально ограничивается доступ, предоставляется минимум привилегий, необходимых пользователям и/или процессам. Помимо этого, системы могут быть защищены межсетевыми экранами и другими средствами обеспечения безопасности сетевого узла, которые ограничивают возможное влияние на систему различных атак, связанных с попытками эксплуатации ее уязвимостей. При этом никакие из перечисленных защитных мер не исключают необходимость выполнения процессов управления патчами, однако они повышают их эффективность и дают компании больше времени на установку патчей.

17.4. Атаки

В этом разделе рассказывается, как недостатки (уязвимости) программного обеспечения и ошибки используются различными методиками проникновения в системы и/или сети. Такие недостатки могут находиться в приложениях, операционной системе, протоколах и сетевом стеке.

Отказ в обслуживании

Сетевой стек является частью операционной системы, он позволяет устройствам обмениваться информацией по сети. Он позволяет создавать сетевые пакеты и отправлять их по проводам, а также принимать сетевые пакеты и обрабатывать их. Различные операционные системы и производители по-разному интерпретируют RFC (Request for Comments) сетевых протоколов, что приводит к несколько различающимся реализациям сетевых стеков в разных системах. Такие особенности могут иметь собственные недостатки, которыми могут воспользоваться хакеры для проведения атаки **«отказ в обслуживании»** (DoS – Denial-of-Service). Такие атаки осуществляются путем направления системе неправильно сформированных пакетов, при этом система не понимает их формата и не знает, как их обрабатывать. Это может привести к нарушению работы системы или прекращению обработки других сетевых пакетов (отказ в обслуживании).

DoS-атаки ежегодно приводят к многомиллионным убыткам компаний, вызванных простоем их систем, потерей доходов и производительности работы, ущербом репутации компании, а

дополнительными затратами человеческих и финансовых ресурсов для выявления возникших проблем и их исправления. DoS-атаки могут привести к отключению отдельных сервисов, либо к полной потере возможности принимать запросы реальных пользователей на доступ к необходимым им ресурсам системы.

DoS-атаки могут заполнить всю ширину полосы пропускания сети жертвы. Для этого у злоумышленника должен быть канал с большей пропускной способностью, либо требуется совместная работа нескольких атакующих, вместе заполняющих каналы связи жертвы. Если атака выполняется с нескольких различных систем, каждая из них усиливает суммарный эффект, что позволяет подавить сегмент сети жертвы.

Другой вид DoS-атаки использует все ресурсы системы жертвы вместо пропускной способности сети. Такими ресурсами может быть процессорное время, свободное место на диске, оперативная память. В следующих разделах рассматриваются некоторые из возможных вариантов DoS-атак.

Smurf

Протокол ICMP представляет собой мини IP-мессенджер, он используется для выяснения функционирования систем с помощью утилиты Ping. ICMP сообщает о текущем статусе и ошибках. Когда пользователь пингует другой компьютер с помощью утилиты Ping, фактически он посылает ему сообщение ICMP ECHO REQUEST. При этом, если этот компьютер включен и работает, он отвечает ему сообщением ECHO REPLY. Это представляет из себя примерно такой диалог: *«Привет, компьютер 10.10.10.1! Ты включен и работаешь?»*, на что этот компьютер отвечает *«да»*.

Smurf-атака требует участия трех игроков: атакующего, жертвы и усиливающей сети. Атакующий изменяет исходящий IP-адрес в заголовке пакета ICMP ECHO REQUEST, указывая в нем IP-адрес системы жертвы (это называется спуфингом). Измененный пакет ICMP ECHO REQUEST в виде широковещательного запроса передается в усиливающую сеть, которая направляет в ответ множество сообщений системе жертвы (т.к. в заголовке указан ее адрес и системы в усиливающей сети думают, что запрос пришел от нее). Такой мощный поток сообщений может перегрузить систему жертвы.

Функция ECHO в протоколе ICMP предназначена лишь для того, чтобы определить, работает ли определенный компьютер и принимает ли он запросы. Однако отсутствие в этом протоколе необходимых мер безопасности позволяет злоумышленникам использовать его для проведения атак.

Контрмеры

- Отключите функции прямой широковещательной рассылки на пограничных маршрутизаторах, чтобы не позволить использовать расположенные за ними сети в качестве усиливающих сетей.
- Настройте правила на маршрутизаторах периметра таким образом, чтобы они отклоняли любые входящие пакеты, в которых в качестве IP-адреса источника указан внутренний IP-адрес сети компании. Это поддельные пакеты, измененные с помощью спуфинга.
- Разрешите только необходимые пакеты ICMP на входе и выходе из сети.
- Используйте IDS уровня сети для выявления подозрительной активности.
- Некоторые системы более чувствительны к определенным видам DoS-атак, но для большинства из них уже были выпущены патчи. Эти патчи должны быть установлены.

Fraggle

Fraggle – это атака, подобная Smurf, но вместо ICMP она использует протокол UDP. Злоумышленник передает подделанный пакет UDP в усиливающую сеть, которая, в свою очередь, направляет свои ответы на систему жертвы. Чем больше усиливающая сеть, тем больший объем трафика направляется на систему жертвы.

Для различных пакетов ICMP и UDP должны быть установлены ограничения по их пропуску во внутреннюю сеть. На это есть масса причин. Злоумышленники часто используют эти протоколы, чтобы узнать о топологии сети, найти в ней маршрутизаторы, узнать о типах используемых в сети систем. Поскольку мы хотим ограничить объем доступной для злоумышленников информации, на маршрутизаторах периметра сети должны быть выполнены указанные ниже настройки.

Контрмеры

- Отключите функции прямой широковещательной рассылки на маршрутизаторах периметра, чтобы не позволить использовать расположенные за ними сети в качестве усиливающих сетей.
- Настройте правила на маршрутизаторах периметра таким образом, чтобы они отклоняли любые входящие пакеты, в которых в качестве IP-адреса источника указан внутренний IP-адрес сети компании. Это поддельные пакеты, измененные с помощью спуфинга.
- Разрешите только необходимые пакеты UDP на входе и выходе из сети.
- Используйте IDS уровня сети для выявления подозрительной активности.
- Некоторые системы более чувствительны к определенным видам DoS-атак, но для большинства из них уже были выпущены патчи. Эти патчи должны быть установлены.

SYN-флуд

Поскольку TCP является протоколом с предварительным установлением соединения, он должен создавать виртуальное соединение между двумя компьютерами. Это виртуальное соединение используется как при «рукопожатии», так и при непосредственном использовании протокола TCP. Для этого требуется трехсторонний процесс. Если компьютеру Comp1 нужно взаимодействовать с компьютером Comp2, Comp1 направляет пакет синхронизации (SYN) на определенный порт Comp2, находящийся в состоянии LISTEN (прослушивание, ожидание приема входящих соединений). Если Comp2 включен, работает и принимает вызовы, он ответит Comp1 пакетом подтверждения SYN/ACK. После получения этого пакета, Comp1 отправит Comp2 пакет ACK, после чего соединение будет установлено.

Системы и их сетевые стеки готовы одновременно обслуживать только определенное ограниченное количество таких соединений, для чего они выделяют определенный объем ресурсов, необходимый для выполнения такого рода функций. Если система получает слишком большое количество запросов на подключение (пакетов SYN), системе становится недостаточно ресурсов для обработки новых запросов на подключение.

Злоумышленники могут воспользоваться этим недостатком, постоянно отправляя жертве сообщения SYN с использованием поддельных пакетов. Компьютер жертвы будет выделять для каждого из них необходимый для установления соединения объем ресурсов, направлять свои ответы о готовности к установлению соединений в пакетах SYN/ACK и ожидая получить пакеты ACK в ответ. Однако жертва так и не получит сообщений ACK, т.к. пакеты поддельные и система жертвы направляет свои пакеты SYN/ACK на несуществующий компьютер. Таким образом, система жертвы послушно выделяет необходимые ресурсы, получая пакеты SYN, означающие, что другая система хочет установить с ней соединение.

Новое соединение ставится в очередь, ожидая получения пакета ACK, но вместо него злоумышленник направляет ей следующий пакет SYN. При этом система жертвы выделяет все больше и больше ресурсов, что в конечном итоге приводит к их недостатку для открытия следующего соединения. Для этого злоумышленнику нужно будет отправить десятки или сотни запросов SYN, прежде чем система жертвы перестанет реагировать на новые запросы. Это делает компьютер жертвы недоступным для подключения реальных пользователей, он отказывает им в обслуживании.

Производители различных систем выпустили патчи, которые увеличивают очередь соединений и/или уменьшают период времени ожидания установления соединения, что позволяет системе оперативно очищать свою очередь несостоявшихся подключений.

Контрмеры

- Уменьшите период времени ожидания установления соединений (это снижает воздействие от такой атаки).
- Увеличьте размер очереди соединений в стеке IP.
- Установите соответствующие патчи от производителя, на системах, в отношении которых возможно проведение такой атаки.
- Используйте IDS уровня сети, который будет отслеживать такие действия и предупреждать ответственных лиц в случае выявления атаки.
- Установите межсетевой экран, для контроля такого вида атак, оповещения о них администратора или сброса таких соединения.

Teardrop

При прохождении пакетов через различные сети, может потребоваться выполнение их фрагментации и пересборки в зависимости от сетевой технологии, используемой каждой конкретной сетью. Каждая сетевая технология имеет свое значение максимального размера передаваемого блока (MTU – maximum transmission unit), который указывает на максимальный размер пакета, который может быть в ней обработан. Некоторые системы проверяют, что пакеты не превышают этот размер, но не проверяют, что пакеты слишком малы. Система жертвы получает фрагменты пакетов и пытается собрать из них целые пакеты, но эти фрагменты сделаны злоумышленником таким образом, что они не могут быть правильно собраны. Многие системы не знают, как бороться с такой ситуацией. Злоумышленники могут воспользоваться этим недостатком системы и отправить ей очень маленькие пакеты, что может вызвать приостановку работы системы или ее перезагрузку. Эти недостатки используются при проведении *атаки Teardrop*.

Контрмеры

- Установите необходимый патч или обновите операционную систему.
- Запретите вход в сеть неправильно сформированных фрагментов пакетов.
- Используйте маршрутизатор, который собирает все фрагменты в полный пакет перед тем, как отправить его системе получателя.

Распределенная атака отказ в обслуживании

Распределенная атака отказ в обслуживании (DDoS – Distributed Denial-of-Service) является логическим продолжением DoS-атаки, которая в процессе проведения атаки использует больше компьютеров. Обычная DoS-атака подавляет компьютеры с помощью только одного атакующего компьютера, отправляя жертве «плохие» пакеты или постоянно отправляя запросы на подключение, пока ресурсы системы жертвы не закончатся и она не сможет отвечать на другие запросы. При проведении DDoS-атаки злоумышленник

использует сотни или тысячи компьютеров, которые посылают запросы на обслуживание серверу (или ферме серверов), пока он не перестанет функционировать.

В процессе такой атаки могут использоваться компьютеры и тех, кто сознательно участвует в атаке, но чаще всего для этого используются *зомби-компьютеры*, владельцы которых даже не подозревают, что они участвуют в атаке. Злоумышленник создает Управляющие серверы (master controller), которые управляют зомби-компьютерами, входящими в *ботсеть* злоумышленника. Управляющие серверы – это системы, на которые у злоумышленника есть административные права доступа, которые он использует для установки на них специального программного обеспечения. Это программное обеспечение позволяет злоумышленнику удобно управлять всем множеством зомби-компьютеров ботсети, оно ожидает инструкций злоумышленника, чтобы дать соответствующие команды зомби-компьютерам. Зомби-компьютеры занимают третий уровень в этой иерархии. На них установлено программное обеспечение бота, которое ожидает команд от управляющих серверов и исполняет их при получении. Когда атакующему требуется выполнить какое-либо действие с помощью своей ботсети, например, провести DDoS-атаку на веб-сайт, он передает его адрес управляющим серверам, а те дают соответствующую команду всем зомби-компьютерам ботсети, которые начинают все вместе одновременно атаковать указанный злоумышленником веб-сайт. Пример такой атаки показан на Рисунке 9-21.

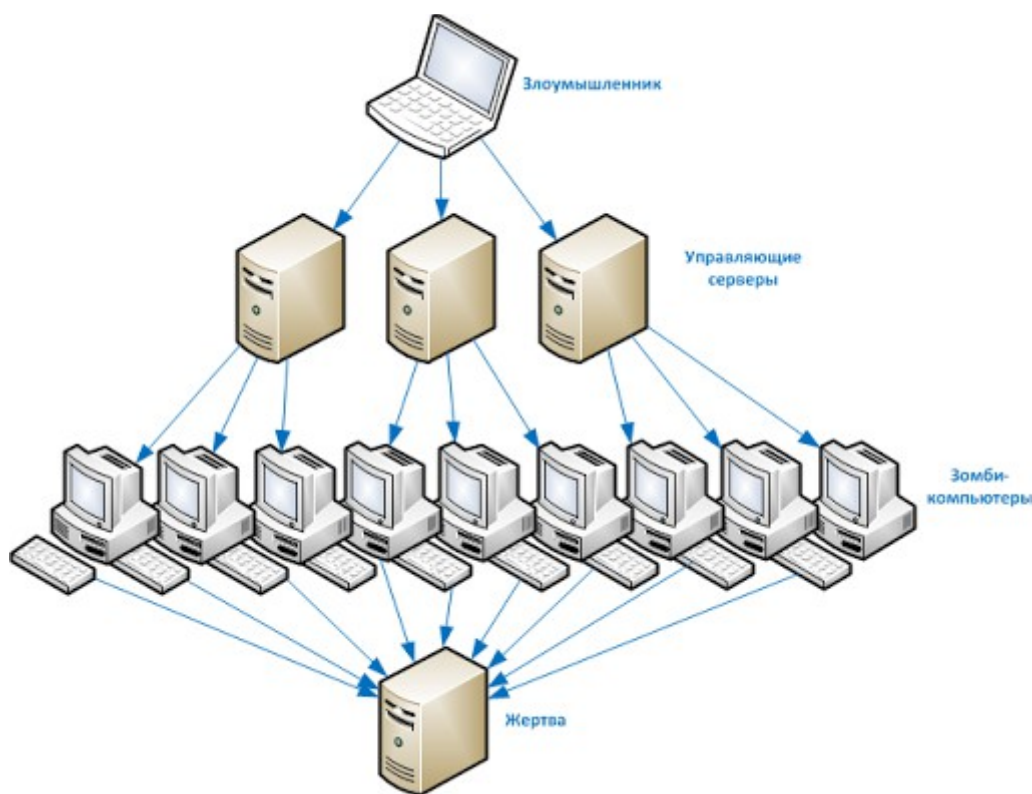


Рисунок 9-21. Для проведения DDoS-атаки, злоумышленник использует управляющие серверы, чтобы передать команду зомби-компьютерам начать одновременно направлять запросы системе жертвы

Контрмеры

- Используйте маршрутизаторы периметра, чтобы ограничить ненужный трафик по протоколам UDP и ICMP.
- Используйте IDS уровня сети для мониторинга такого вида подозрительной деятельности.
- Отключите неиспользуемые подсистемы и сервисы на компьютерах.

- Переименуйте учетную запись администратора и организуйте строгое управление паролями, чтобы вашими системами не могли воспользоваться неизвестные.
- Настройте правила маршрутизаторов периметра таким образом, чтобы они отклоняли любые входящие сообщения с пакетами, содержащими IP-адреса внутренней сети в качестве адреса отправителя. Это поддельные пакеты.

18. Резюме

Хотя при разработке программного обеспечения функциональность имеет первостепенное значение, было бы весьма полезным добавить безопасность в эту смесь еще до начала проекта и интегрировать ее в каждый этап процесса разработки. Многие компании пока не считают этот подход к разработке программного обеспечения выгодным, однако они меняют свое мнение при возникновении необходимости в разработке все большего количества патчей и исправлений, устраняющих проблемы безопасности в уже работающем у их покупателей программном обеспечении. У покупателей начинает появляться спрос на более безопасные продукты.

Разработка программного обеспечения представляет собой сложную задачу, особенно учитывая стремительное развитие технологий, изменяющихся со скоростью света, эволюционирующих сред, и возрастающих ожиданий производителей, каждый из которых хочет быть «царем горы» в рамках своего сегмента рынка программного обеспечения. Все это также усложняет внедрение в программные продукты эффективной безопасности. На протяжении многих лет, программистам и разработчикам не нужно было учитывать вопросы безопасности в своем коде, но эта тенденция меняется. Образование, опыт, осведомленность, требования, а также запросы потребителей – все это является необходимыми частями для применения более безопасных подходов и технологий в программном коде, которым мы все пользуемся.

Тест

Вопросы экзамена CISSP являются концептуальными, поэтому они сформулированы соответствующим образом. Задачей кандидата является выбор наилучшего из всех представленных вариантов ответа. Среди вариантов ответа может не быть идеального ответа на поставленный вопрос - кандидат должен выбрать лучший ответ из имеющихся вариантов.

1. Что является завершающим этапом процесса управления изменениями?

- ☐ A. Правильная настройка оборудования
- ☐ B. Обновление документации и инструкций
- ☐ C. Информирование пользователей об изменении
- ☐ D. Отчет руководству об изменении

2. Что из перечисленного ниже лучше всего описывает логическую бомбу?

- ☐ A. Она используется для перемещения информационных активов с одного компьютера на другой
- ☐ B. Она выполняет какое-либо действие при наступлении определенных условий
- ☐ C. Она может самостоятельно «размножаться»
- ☐ D. Она выполняет как полезные действия, так и вредоносные

3. Из сети Интернет загружена утилита, выполняющая очистку диска и удаление ненужных временных файлов. В действительности эта утилита, помимо указанных действий, также перехватывает вводимые пользователем пароли и отправляет их по некоторому адресу. К какому виду вредоносного программного обеспечения относится такая утилита?

- ☐ A. Вирус
- ☐ B. Троянская программа
- ☐ C. Червь
- ☐ D. Логическая бомба

4. Почему макро-вирусы так распространены?

- ☐ A. Они быстро распространяются
- ☐ B. Они могут заражать любую платформу
- ☐ C. Язык, на котором пишутся макросы, очень прост в использовании

☐ D. Они активируются по событиям, обычно происходящим на любой системе

5. Какое из перечисленных ниже действий не является частью процесса управления конфигурациями?

- ☐ A. Передача официального запроса
- ☐ B. Конфигурирование и настройка операционной системы
- ☐ C. Конфигурирование оборудования
- ☐ D. Конфигурирование и настройка приложения

6. Зачем для автоматизации анализа журналов регистрации событий безопасности применяются экспертные системы?

- ☐ A. Для предотвращения вторжений
- ☐ B. Чтобы убедиться, что используется наилучший метод доступа
- ☐ C. Для выявления вторжений
- ☐ D. Чтобы собрать статистику отклонений от базового уровня

7. Какой вид вредоносного программного обеспечения «размножается» с использованием ресурсов зараженной системы??

- ☐ A. Червь
- ☐ B. Вирус
- ☐ C. Троянская программа
- ☐ D. Составной вирус

8. Экспертная система использует все перечисленные элементы, за исключением _____.

- ☐ A. Автоматическая логическая обработка
- ☐ B. Общие методы поиска решений проблем
- ☐ C. Механизм логических выводов
- ☐ D. Механизм циклических рассуждений

9. Какой из перечисленных ниже видов вредоносного программного обеспечения «размножается», добавляя свой код к другим программам??

- ☐ A. Червь
- ☐ B. Вирус
- ☐ C. Троянская программа
- ☐ D. Любой вредоносный код

10. В чем заключается важность логических выводов в экспертной системе?

- ☐ A. База знаний состоит из фактов, но должна существовать возможность комбинировать эти факты для получения на их основе новой информации и решений
- ☐ B. Машина логических выводов позволяет эффективно бороться с составными вирусами
- ☐ C. База знаний должна работать с единицами, имитирующими нейроны в мозге
- ☐ D. Необходимо контролировать доступ, чтобы предотвратить несанкционированный доступ

11. На программу за последнее время несколько раз устанавливались патчи, но недавно ее основной исполняемый файл был заражен опасным вирусом. Антивирусная программа сообщает, что лечение зараженного файла приведет к его повреждению. Какое действие является наиболее правильным в этом случае??

- ☐ A. Выполнить лечение файла и обратиться к производителю для его восстановления
- ☐ B. Сделать резервную копию и затем выполнить лечение файла
- ☐ C. Заменить файл на его копию, сделанную вчера
- ☐ D. Восстановить с резервной копии незараженную версию файла с установленными патчами

12. Что из перечисленного ниже является централизованным средством управления базой данных и различными аспектами самих данных?

- ☐ A. Хранилище данных
- ☐ B. База данных
- ☐ C. Словарь данных
- ☐ D. Контроль доступа

13. Каким образом реализуется многоэкземплятность (polyinstantiation)?

- ☐ A. Ограничивается возможность доступа субъектов с низким уровнем допуска к информации, требующей более высокого уровня допуска
- ☐ B. Создается копия объекта и в копии изменяются отдельные атрибуты
- ☐ C. Создается другой объект, который будет по-другому реагировать на те же входные данные

☐ D. Создаются различные объекты, которые унаследуют атрибуты от родительского класса

14. Если в базе данных происходит сбой, что позволяет начать обработку с момента, предшествующего сбою?

- ☐ A. Контрольная точка
- ☐ B. Словарь данных
- ☐ C. Метаданные
- ☐ D. Инструмент интеллектуального анализа данных (data-mining)

15. Какой тип защитных мер реализуют представления баз данных?

- ☐ A. Детективный
- ☐ B. Корректирующий
- ☐ C. Превентивный
- ☐ D. Административный

16. Если одно из подразделений компании может посмотреть записи по работе сотрудников компании, а другие подразделения – не могут, примером чего это является?

- ☐ A. Контекстно-зависимое управление доступом
- ☐ B. Контентно-зависимое управление доступом
- ☐ C. Разделение обязанностей
- ☐ D. Мандатное управление доступом

17. Что из перечисленного ниже применяется в базах данных для предотвращения атак на основе предположений (inference attack)?

- ☐ A. Разделение базы данных, скрытие ячеек, шум и пертурбация
- ☐ B. Контроль доступа к словарю данных
- ☐ C. Разделение базы данных, скрытие ячеек, наборы небольших запросов
- ☐ D. Разделение базы данных, шум, пертурбация и наборы небольших запросов

18. В чем заключается недостаток использования контекстно-зависимого управления доступом для баз данных?

- ☐ A. Возможны обращения к другим адресам памяти
- ☐ B. Может вызвать проблемы конкуренции (concurrency problem)
- ☐ C. Может привести к увеличению загруженности компьютерных ресурсов
- ☐ D. Может привести к взаимным блокировкам (deadlock)

19. Если безопасность не является частью процесса разработки базы данных, каким образом она обычно реализуется?

- ☐ A. С помощью скрытия ячеек
- ☐ B. С помощью доверенного серверного приложения
- ☐ C. С помощью доверенного фронтального интерфейса
- ☐ D. С помощью представлений

20. В чем заключается преимущество использования контентно-зависимого управления доступом в базах данных?

- ☐ A. Снижается нагрузка на компьютерные ресурсы
- ☐ B. Обеспечивается параллельная обработка
- ☐ C. Запрещается блокировка данных
- ☐ D. Обеспечивается более детальный контроль

21. Что из перечисленного ниже применяется в технологиях Распределенных вычислительных сред?

- ☐ A. Глобально уникальный идентификатор (GUID)
- ☐ B. Универсальный уникальный идентификатор (UUID)
- ☐ C. Универсальный глобальный идентификатор (UGID)
- ☐ D. Глобальный универсальный идентификатор (GUID)

22. На каком этапе проекта впервые должны быть учтены вопросы безопасности?

- ☐ A. На этапе функционального проектирования
- ☐ B. На этапе интеграционного тестирования
- ☐ C. На этапе разработки технического задания
- ☐ D. На этапе внедрения

23. Что из перечисленного ниже должно сделать веб-приложение при выявлении неправильной транзакции?

- ☐ А. Выполнить откат и восстановить первоначальные данные
- ☐ В. Прервать выполнение транзакций, пока неправильная транзакция не будет исправлена
- ☐ С. Сформировать отчет об ошибке
- ☐ D. Создать контрольную точку

24. Что является заключительным этапом жизненного цикла процесса разработки системы?

- ☐ А. Сертификация
- ☐ В. Тестирование модулей
- ☐ С. Разработка
- ☐ D. Аккредитация

25. Что из перечисленного ниже является строками и столбцами в реляционной базе данных?

- ☐ А. Строки и записи
- ☐ В. Атрибуты и строки
- ☐ С. Ключи и представления
- ☐ D. Записи и атрибуты

Домен 10. Операционная безопасность.

Операционная безопасность (operations security) имеет отношение ко всему тому, что делается для сохранения доступности и обеспечения защиты сетей, компьютерных систем, приложений и сред. В частности, операционная безопасность обеспечивает правильную настройку привилегий и прав доступа пользователей и приложений к тем ресурсам, для работы с которыми они имеют необходимые полномочия, а также выполнение журналирования событий, мониторинга и анализа журналов, подготовки необходимой отчетности. После того, как сеть создана и введена в эксплуатацию, в ней начинают выполняться различные операции, в т.ч. операции по сопровождению и постоянной поддержке функционирования сети и повседневной деятельности в ней. Это действия, которые позволяют сети и системам в ней работать правильно и безопасно.

Сети и вычислительные среды являются динамичными. То, что они безопасны сегодня, совершенно не означает, что они будут безопасны через месяц. Многие компании нанимают консультантов по безопасности, чтобы они дали рекомендации по улучшению их инфраструктуры, политик и процедур. Компания может потратить миллионы на реализацию предложений консультанта: закупить, установить и правильно настроить межсетевые экраны, системы обнаружения вторжений (IDS), антивирусные средства и системы управления патчами. Однако если для ее IDS и антивирусных средств не выполняется постоянное обновление сигнатур, не устанавливаются обновления и новые версии самих этих средств, если межсетевые экраны и маршрутизаторы не проверяются на наличие уязвимостей, если новые программы появляются в сети, минуя планы по обеспечению безопасности, компания может легко скатиться обратно в небезопасное состояние. Это происходит из-за того, что компания не поддерживает в актуальном состоянии свою операционную безопасность.

Большинство вопросов, связанных с операционной безопасностью, были рассмотрены в предыдущих Доменах. Они были интегрированы в темы этих Доменов, и не помечались особо в качестве вопросов операционной безопасности. В этом Домене мы будем ссылаться на уже рассмотренные вопросы и более детально остановимся на еще не рассмотренных вопросах операционной безопасности, имеющих большое значение для компаний и кандидатов CISSP.

1. Роль Департамента эксплуатации

Необходим постоянный контроль, чтобы сохранять уверенность в том, что внедрены правильные политики, процедуры, стандарты и инструкции, что им действительно следуют сотрудники компании. Это является важной частью проявления *должной заботы* (due care) и *должной осмотрительности* (due diligence), которые обязана выполнять компания. Должная забота и должная осмотрительность основаны на понимании «разумного человека». Разумный человек считается ответственным, внимательным, осторожным и практичным. Это полностью применимо и к компании, проявляющей должную заботу и должную осмотрительность. Необходимо выполнить правильные шаги для достижения необходимого уровня безопасности, сохраняя при этом баланс простоты использования, соблюдения установленных требований и минимизации затрат. Необходимо предпринимать постоянные усилия и обеспечивать дисциплину, чтобы сохранить надлежащий уровень безопасности. Операционная безопасность – это все то, что делается для обеспечения адекватной защиты людей, приложений, оборудования, а также среды, в которой они работают.

Хотя операционная безопасность основана на постоянной поддержке защитных мер и средств для сохранения необходимого уровня безопасности среды, при выполнении этих задач также должны учитываться обязательства компании и ее юридическая ответственность. Компании и их высшее руководство часто имеют закрепленные на законодательном уровне обязательства по обеспечению защиты активов, реализации мер безопасности, проверке их эффективности, чтобы убедиться, что они реально обеспечивают

необходимый уровень защиты. Если эти обязательства по операционной безопасности не будут выполнены, у компании могут возникнуть серьезные проблемы.

Компания должна учесть множество угроз, в т.ч. возможное разглашение конфиденциальной информации, кражу имущества, повреждение данных, прерывание отдельных сервисов, а также полное разрушение физической или логической среды. Важно определить, какие системы и операции обрабатывают информацию ограниченного доступа (что означает, что должна быть обеспечена защита конфиденциальности обрабатываемой в них информации), а какие являются критичными для работы самой компании (что означает, что они должны быть доступны постоянно) (вопросы законодательства и ответственности, связанные с безопасностью, были рассмотрены в Домене 08).

Важно также отметить, что, несмотря на то, что значительная часть операционной деятельности компаний связана с компьютерными ресурсами, компании по-прежнему зависят и от физических ресурсов, включая бумажные документы и данные, хранящиеся на дисках, лентах и других съемных носителях. Значительная часть операционной безопасности посвящена вопросам, связанным с физическими проблемами и проблемами внешней среды, такими как температура и контроль влажности, повторное использование носителей информации, утилизация и уничтожение носителей информации, содержащих критичную информацию. Также операционная безопасность связана с конфигурациями, производительностью, отказоустойчивостью, безопасностью, ведением учета, проведением проверок соблюдения действующих операционных стандартов и требований.

2. Административное управление

Административное управление (administrative management) является очень важной частью операционной безопасности. Одним из аспектов административного управления являются кадровые вопросы. Они включают в себя разделение обязанностей и ротацию обязанностей. Целью *разделения обязанностей* (separation of duties) является обеспечение того, чтобы один человек не мог в одиночку нарушить безопасность компании каким-либо образом. Высокорискованные операции (действия) должны быть разбиты на несколько частей и распределены между разными людьми или подразделениями. При этом компании не нужно принимать опасно высокий уровень доверия к отдельным лицам. Для реализации мошеннических действий потребуется сговор, т.е. в этом должны принимать участие несколько человек, предварительно договорившись между собой, что менее вероятно, чем действия отдельного злоумышленника. Таким образом, разделение обязанностей является превентивной мерой, которая не позволяет нарушить безопасность компании без специального сговора нескольких ее сотрудников.

В Таблице 10-1 перечислены распространенные в компаниях роли и их основные обязанности. Для каждой роли в компании должны быть разработаны полные и понятные должностные инструкции. Сотрудники безопасности должны руководствоваться этими должностными инструкциями при назначении прав доступа и разрешений, чтобы обеспечить наличие у каждого из пользователей доступа только к тем ресурсам, которые необходимы им для выполнения возложенных на них задач.

Организационная роль	Основные обязанности
Контролер	Проверяет информацию, полученную от аналитиков, администраторов и пользователей, а затем передает ее различным группам пользователей
Системный аналитик	Проектирует потоки данных между системами на основе операционных и пользовательских требований
Прикладной программист	Разрабатывает и сопровождает прикладные системы
Специалист технической поддержки	Решает проблемы, возникающие у конечных пользователей, технические проблемы и эксплуатационные проблемы в системах
ИТ-инженер	Выполняет ежедневные эксплуатационные задачи в системах и приложениях
Администратор баз данных	Создает новые таблицы баз данных и управляет базами данных
Администратор сети	Устанавливает и сопровождает компоненты среды LAN/WAN
Администратор безопасности	Определяет, настраивает и сопровождает механизмы безопасности, обеспечивающие защиту компании
Библиотекарь данных на магнитных лентах (Tape Librarian)	Получает, записывает, стирает и защищает резервные копии файлов систем и приложений на внешних носителях информации, таких как магнитная лента или диски
Контролер качества	Может включать в себя как Обеспечение качества (Quality Assurance), так и Контроль качества (Quality Control). QA обеспечивает соблюдение заранее разработанных стандартов, касающихся сопроводительной документации и номенклатуры. QC обеспечивает, что деятельность, сервисы, оборудование и персонал работают в рамках принятых стандартов

Таблица 10-1. Роли и связанные с ними задачи

Таблица 10-1 содержит всего несколько ролей с несколькими задачами для каждой роли. Компания должна разработать полный список ролей, работающих в ее среде, с указанием задач и обязанностей каждой роли. Этот список должен использоваться в дальнейшем владельцами данных и сотрудниками безопасности при определении, кто должен иметь доступ к определенным ресурсам и какой именно доступ.

Разделение обязанностей позволяет избежать ошибок и свести к минимуму конфликты интересов, которые могут произойти, если один человек выполняет некую задачу от начала до конца. Например, программист не должен быть единственным, кто протестирует разработанный им код. Тестирование его кода на предмет функциональности и целостности должен произвести другой человек, имеющий другие задачи, поскольку сам программист может быть совершенно уверен в своей программе, у него может быть собственное понимание, как программа должна работать, поэтому он проведет тестирование только нескольких функций и нескольких вариантов входных значений и только в определенных средах.

Другим примером разделения обязанностей являются различия в функциях пользователя компьютера и системного администратора. Между их обязанностями должна проходить четкая линия. Конкретные обязанности пользователей и администраторов могут различаться в различных средах и в зависимости от требуемого уровня безопасности в этих средах. Как правило, системные администраторы и администраторы безопасности выполняют такие функции, как резервное копирование и восстановление, разграничение прав доступа, добавление и удаление пользователей, создание пользовательских профилей. При этом пользователю компьютера может быть разрешено устанавливать программное обеспечение на свой компьютер, устанавливать первоначальный пароль, изменять настройки рабочего стола и некоторые параметры системы. Однако пользователь не должен иметь возможностей изменять свой профиль безопасности, добавлять и удалять глобальных пользователей, предоставлять и изменять права доступа к критичным сетевым ресурсам, т.к. это могло бы нарушить концепцию разделения обязанностей.

Ротация обязанностей (Job rotation) означает, что один и тот же человек не может постоянно выполнять задачи одной должности в компании. По прошествии определенного времени, он должен быть переведен на другую должность или на него должны быть возложены иные обязанности. Это позволяет компании иметь несколько сотрудников,

которые понимают и умеют выполнять задачи и обязанности такой должности, что обеспечивает наличие резерва, который может понадобиться в случае, если критичный для компании сотрудник уйдет из компании или будет отсутствовать на рабочем месте. Ротация обязанностей также помогает выявить мошеннические действия, поэтому ее можно считать детективной защитной мерой. Если Кейт раньше занимал должность Девида, он хорошо знает задачи и процедуры, которые входят в обязанности этой должности. Поэтому Кейт может лучше других выявить что-то необычное и подозрительное в деятельности Девида (вопросы, связанные с ротацией обязанностей рассматривались в Домене 02).

Приципы наименьших привилегий (least privilege) и «необходимо знать» (need to know) также являются административными защитными мерами, которые должны быть реализованы в операционной среде. Принцип **наименьших привилегий** означает, что человек должен иметь достаточно прав и полномочий для выполнения задач своей роли в компании, но не более того. Если человек имеет излишние права или полномочия, это может стать причиной злоупотреблений и привести к излишним рискам для компании. Например, если Александр в компании занимает должность технического писателя, ему не обязательно иметь доступ к исходным текстам программного обеспечения, разрабатываемого компанией. Поэтому механизмы, контролирующие доступ Александра к ресурсам, не должны позволить ему получить доступ к исходным текстам. Это будет правильной реализацией защитных мер, обеспечивающих операционную безопасность ресурсов.

Принципы наименьших привилегий и «необходимо знать» имеют общие черты. Каждый пользователь должен иметь доступ к тому, что ему «необходимо знать». Если Майку не требуется знать сумму налогов, которую компания заплатила в прошлом году, его права доступа в соответствующей системе не должны содержать разрешений на доступ к этим данным, что является примером реализации принципа наименьших привилегий. Использование нового программного обеспечения управления идентификацией (identity management software), которое сочетает в себе традиционные каталоги, системы контроля доступа и инициализации пользователей на серверах, в приложениях и системах, становится нормой для компаний. Это программное обеспечение позволяет в любой момент убедиться, что конкретным пользователям предоставлены только определенные права доступа, как правило, оно включает в себя расширенные функции аудита, которые могут быть использованы для контроля соблюдения действующих требований (внутренних и внешних).

Права доступа пользователя являются комбинацией атрибутов наименьших привилегий, уровня допуска пользователя, элементов, которые ему «необходимо знать», уровня критичности ресурса, а также режима безопасности, в котором работает компьютер. Система может работать в разных режимах в зависимости от критичности обрабатываемых данных, уровня допуска пользователей, и того, что эти пользователи имеют полномочия делать. Режим работы описывает условия, в которых система фактически работает. Мы рассматривали эти вопросы в Домене 03.

Обязательный отпуск (mandatory vacations) является другим видом административных защитных мер, хотя название может показаться немного странным на первый взгляд. В Домене 01 были рассмотрены причины, по которым следует убедиться, что сотрудники реально берут свои отпуска. Эти причины включают в себя возможности для выявления мошеннической деятельности и создание более благоприятных условий для выполнения ротации обязанностей. Если бухгалтер занимался мошенничеством, переводя по одному рублю с множества различных счетов на свой счет (атака «салями»), у компании будет больше шансов обнаружить это, если этот бухгалтер находится в отпуске неделю или больше. Когда один сотрудник находится в отпуске, другой сотрудник заменяет его. При этом он может найти сомнительные документы и указания на совершенные мошеннические действия, либо компания может заметить изменения в некоторых аспектах картины повседневной работы после того, как сотрудник ушел в отпуск.

Для проведения проверки лучше всего, чтобы проверяемый сотрудник взял отпуск и недели две отсутствовал на рабочем месте. Это даст достаточно времени для сбора улик, связанных с его мошеннической деятельностью. Сотрудники, которые сильно сопротивляются уходу в отпуск, должны вызывать подозрения, поскольку существует вероятность, что они не хотят идти в отпуск, т.к. боятся, что в их отсутствие могут быть обнаружены свидетельства их мошеннической деятельности.

2.1. Администратор безопасности и администратор сети

Администратор безопасности не должен отчитываться перед сетевым администратором, потому что их работа направлена на достижение различных целей. Администратор сети должен обеспечить высокую доступность и производительность сети и ресурсов, предоставление пользователям той функциональности, которую они запрашивают. Однако зачастую концентрация на производительности и удобстве для пользователей дорого обходится безопасности. Обычно внедрение механизмов безопасности приводит к снижению производительности, поскольку эти механизмы принимают активное участие в процессах обработки и передачи данных: контентная фильтрация, поиск вирусов, выявление и предотвращение вторжений, обнаружение аномалий и т.д. Поскольку обычно все это не входит в зону ответственности сетевого администратора, может возникнуть конфликт интересов. Поэтому администратор безопасности должен быть подчинен другому руководителю (по отношению к администратору сети), чтобы вопросы безопасности не игнорировались и им не устанавливался самый низкий приоритет.

Ниже перечислены задачи, которые должны выполняться администратором безопасности, а не сетевым администратором:

- **Внедрение и сопровождение устройств и программного обеспечения безопасности.** Несмотря на то, что некоторые поставщики утверждают, что их продукция будет эффективно обеспечивать безопасность при реализации по принципу «поставить и забыть», продукты безопасности требуют постоянного мониторинга и обслуживания, что необходимо для их полноценной работы. Для закрытия найденных в этих продуктах уязвимостей, а также для получения новых возможностей, может потребоваться установка патчей или обновление версий программного обеспечения этих продуктов.
- **Проведение оценки безопасности.** При проведении оценки безопасности, требуются знания и опыт администратора безопасности, необходимые для выявления уязвимостей в системах, сетях, программном обеспечении компании. Результаты оценки безопасности позволят руководству компании лучше понять риски, перед лицом которых стоит компания, и принять наиболее разумные решения по закупке и внедрению продуктов и услуг безопасности, выбрать эффективную стратегию снижения риска с точки зрения соотношения затрат и величины, на которую будут снижены риски, либо принять решение о принятии или переносе рисков (путем приобретения страховки) или их избежании (путем отказа от деятельности, вызывающей появление этого риска), если расходы на снижение риска до приемлемого уровня превысят вероятные выгоды от выполнения вызывающей его деятельности.
- **Создание и сопровождение профилей пользователей, внедрение и поддержка механизмов контроля доступа.** Администратор безопасности обеспечивает практическое применение требований политики безопасности в части соблюдения принципа наименьших привилегий, выполняет контроль существующих учетных записей, разрешений и прав, которые им предоставлены.
- **Настройка и сопровождение меток безопасности для среды мандатного управления доступом (MAC).** MAC применяется в основном в правительственных и

военных организациях. При его использовании объектам и субъектам доступа присваиваются метки безопасности. Решения о предоставлении доступа принимаются по результатам сравнения уровня классификации объекта с уровнем допуска субъекта. Более подробно это рассматривалось в Домене 02. Внедрение и сопровождение методов управления доступом является обязанностью администратора безопасности.

- **Установка первоначальных паролей для пользователей.** Новые учетные записи должны быть защищены от злоумышленников, которые могут знать шаблоны, на основе которых вырабатываются первоначальные пароли, либо могут найти новые учетные записи, для которых пароли пока не установлены. Они могут получить контроль над такой учетной записью, прежде чем уполномоченный пользователь воспользуется ей и сменит пароль. Администратор безопасности должен использовать автоматизированные генераторы паролей, либо вручную установку новых паролей, а затем передавать их уполномоченным пользователям. При этом пароли должны создаваться таким образом, чтобы злоумышленники не могли угадать или быстро подобрать их. Новые учетные записи не должны оставаться незащищенными.
- **Анализ журналов регистрации событий.** Хотя наибольшую безопасность обеспечивают превентивные защитные меры (такие как межсетевые экраны, которые блокируют неразрешенную сетевую активность), детективные меры, такие как анализ журналов регистрации событий, также необходимы. Например, межсетевой экран заблокировал за вчерашний день 60000 попыток несанкционированного доступа. Единственный способ для администратора безопасности (или используемого им автоматизированного средства) узнать, хорошо это или плохо – проанализировать журналы межсетевого экрана, чтобы найти в них признаки действий злоумышленника. Если эти 60000 заблокированных попыток доступа были обычным случайным низкоуровневым шумом сети Интернет, тогда (вероятно) это нормально, но если эти попытки доступа были более сложными и исходили из конкретного диапазона адресов, вероятно, было проведено спланированное (и, возможно, успешное) нападение. Анализ журналов регистрации событий администратором безопасности позволяет выявить различные плохие вещи по мере того, как они происходят, и (надеюсь) до того, как они причинят реальный ущерб.

2.2. Подотчетность

Доступ пользователей к ресурсам должен быть ограничен и надлежащим образом контролироваться, чтобы предотвратить нанесение ущерба компании и ее информационным ресурсам вследствие использования избыточных привилегий. Попытки доступа пользователей и их действия в процессе использования ресурсов должны журналироваться, должен быть организован мониторинг и регулярный анализ содержимого журналов регистрации событий. Каждому пользователю должен быть присвоен уникальный идентификатор, который должен записываться в журнал регистрации событий среди прочей информации о каждом событии, что позволит обеспечить персональную ответственность пользователей за свои действия. Каждый пользователь должен понимать свои обязанности при использовании ресурсов компании и нести ответственность за свои действия.

Ведение и мониторинг журналов регистрации событий помогают определить, действительно ли произошло нарушение, либо система просто нуждается в перенастройке для более качественного сбора событий, отражающих действия, которые выходят за пределы установленных разрешенных границ. Если действия пользователя не были записаны в журнал регистрации событий или не были проанализированы, очень трудно выявить факты наличия у пользователей избыточных привилегий или факты несанкционированного доступа.

Журналирование событий должно выполняться на постоянной основе. При этом кто-то должен регулярно просматривать содержимое журналов регистрации событий. Если никто

не просматривает на регулярной основе эти журналы, сбор этих журналов вообще теряет смысл. Журналы регистрации событий зачастую содержат слишком много непонятных или, наоборот, обычных событий, чтобы их можно было эффективно анализировать вручную. Для этих целей существуют специализированные продукты и сервисы анализа журналов регистрации событий, которые разбирают каждое событие и сообщают о важных находках. Журналы регистрации событий должны анализироваться с помощью автоматизированных средств, либо вручную, при невозможности автоматизированного анализа, для выявления подозрительных действий, изменения настроек или поведения системы по сравнению с базовыми настройками (поведением). Кроме того, именно анализ журналов регистрации событий может заранее предупредить администраторов о многих проблемах, прежде чем они станут слишком большими и выйдут из-под контроля (вопросы журналирования, мониторинга и анализа журналов регистрации событий были рассмотрены в Доменах 02 и 08).

При проведении мониторинга администраторы должны задаваться следующими вопросами в отношении пользователей, их действий, текущего уровня безопасности и доступа:

- *Получают ли пользователи доступ к информации или выполняют задачи, которые не являются необходимыми для выполнения их должностных обязанностей?* В случае положительного ответа, должна быть проведена оценка (и, возможно, изменение) имеющихся у пользователя прав и разрешений.
- *Повторяются ли одни и те же (или однотипные) ошибки?* В случае положительного ответа, следует провести дополнительное обучение пользователей.
- *Не слишком ли много пользователей имеют права и привилегии доступа к критичным данным и ресурсам ограниченного доступа?* В случае положительного ответа, должна быть проведена переоценка имеющихся прав доступа к этим данным и ресурсам. Возможно, следует сократить число лиц, которые имеют доступ к ним, и/или изменить границы их прав доступа.

2.3. Уровни отсечения

Компании могут заранее установить пороговые значения количества определенных видов ошибок, в пределах которых эти ошибки будут считаться допустимыми, однако при превышении порогового значения, вызвавшие их действия должны быть проанализированы. Такое пороговое значение является базовым уровнем, в пределах которого выполнение определенных нарушений считается допустимым, однако при его превышении включается сигнал тревоги. Этот базовый уровень называют **уровнем отсечения** (clipping level). После превышения этого уровня отсечения, дальнейшие нарушения должны быть обязательно сохранены в журнале для последующего анализа. В основном для отслеживания таких действий и шаблонов поведения используются системы IDS, т.к. человек просто не в силах постоянно мониторить стопку журналов регистрации событий и эффективно выявлять в них определенные шаблоны действий. При превышении уровня отсечения, IDS может отправить соответствующее уведомление администратору по электронной почте или на его мобильный телефон, либо просто добавить эту информацию в журнал регистрации событий, в зависимости от настроек программного обеспечения IDS.

Целью использования уровней отсечения, анализа и мониторинга журналов регистрации событий является обнаружение проблем до того, как они нанесут серьезный ущерб, а также для получения предупреждений о происходящих или готовящихся атаках.

ПРИМЕЧАНИЕ. Внедренные меры и средства безопасности должны иметь определенную степень "прозрачности". Это дает возможность пользователям выполнять свои задачи и обязанности без необходимости выполнения дополнительных шагов, вызванных введенными мерами безопасности. Прозрачность также позволяет не давать пользователям слишком много сведений об используемых защитных мерах, что не позволяет им придумать способы их обхода. Если защитные меры слишком очевидны, злоумышленнику будет проще выяснить, как можно их

обойти или преодолеть.

3. Уровень гарантий

При проведении оценки продуктов по уровню доверия и гарантий, которые они предоставляют, часто в состав такой оценки входят операционные гарантии (operational assurance) и гарантии жизненного цикла (life-cycle assurance). Оценка **операционных гарантий** сосредоточена на архитектуре продукта, встроенных возможностях и функциях, которые позволяют клиентам постоянно иметь необходимый уровень защиты при использовании продукта. В частности, в процессе оценки операционных гарантий, рассматриваются механизмы контроля доступа, разделение программного кода, исполняемого в реальном и защищенном режимах, возможности контроля и мониторинга, проверяется наличие скрытых каналов, а также механизмы восстановления доверенного состояния в случае возникновения непредвиденных обстоятельств.

Гарантии жизненного цикла относятся к тому, как продукт был разработан и сопровождается. К каждому этапу жизненного цикла продукта предъявляются определенные стандарты и ожидания, которым он должен соответствовать, чтобы считаться доверенным продуктом. Примерами стандартов гарантий жизненного цикла являются технические задания, настройки уровня отсечения, модульное и интеграционное тестирование, управление конфигурациями, доверенное распространение. Производители, собирающиеся достичь одного из высших рейтингов безопасности своей продукции, должны будут пройти через оценку и тестирование каждого из этих вопросов.

Следующие разделы рассматривают некоторые из видов операционных гарантий и гарантий жизненного цикла не только в части оценки, но и в части обязанностей компании, возникающих после внедрения продукта. Продукт – это просто рабочий инструмент компании, который используется для получения определенной функциональности и безопасности. Именно компания должна обеспечить, чтобы эта функциональность и безопасность были постоянно доступны, выполняя необходимые проактивные действия.

Ссылки по теме:

- NIST Security Configuration Checklists Program for IT Products
- An Introduction to Computer Security: The NIST Handbook
- “The Operations Security Connection,” by Arion N. Pattakos, PM (Jan.–Feb. 1999)

4. Эксплуатационные обязанности

Операционная безопасность охватывает защитные меры и контрмеры, предназначенные для защиты ресурсов и информации, а также оборудования, на котором размещаются эти ресурсы и информация. Целью операционной безопасности является снижение вероятности возникновения ущерба, вызванного несанкционированным доступом или раскрытием информации. Эта цель достигается за счет ограничения возможностей для злоупотреблений.

В некоторых компаниях организован отдельный Департамент эксплуатации, который отвечает за выполнение действий и процедур, необходимых для обеспечения бесперебойной работы сети и сохранения ее производительности на определенном уровне. В других компаниях для выполнения этих функций могут быть назначены несколько ответственных лиц. В любом случае люди, которые выполняют эти обязанности, несут ответственность за выполнение указанных действий и процедур, они обязаны контролировать и отслеживать определенные вопросы.

Выполнение операций в компьютерной среде может иметь отношение к программному обеспечению, персоналу и оборудованию, однако Департамент эксплуатации зачастую сосредотачивает свое внимание на аспектах программного и аппаратного обеспечения. Нужно понимать, что ответственность за поведение сотрудников и выполнение ими своих

обязанностей несет руководство компании, а сотрудники Департамента эксплуатации отвечают за обеспечение надлежащей защиты систем и их предсказуемую работу.

В задачи Департамента эксплуатации обычно входит предотвращение повторного возникновения проблем, снижение количества аппаратных и программных сбоев до приемлемого уровня, а также снижение влияния инцидентов и сбоев. Департамент эксплуатации должен расследовать любые необычные или непонятные случаи, неожиданные повышения загрузки систем, отклонения от стандартов и другие ненормальные ситуации, которые происходят в сети.

4.1. Необычные и необъяснимые события

Сети, а также аппаратное и программное обеспечение в них, могут быть сложными и динамичными. Происходящие в них события нередко вызывают удивление и могут казаться необъяснимыми. Именно Департамент эксплуатации должен расследовать такие ситуации, диагностировать проблемы и находить их логичное решение.

Рассмотрим пример. Несколько узлов в сети постоянно теряют сетевое соединение без каких-либо видимых причин. Для решения этой проблемы выделяется группа сотрудников Департамента эксплуатации, которая должна контролируемым образом устранить возникшие неполадки, убедившись, что не забыт ни один из возможных источников проблемы. Группа может выполнить проверку соединений между этими узлами и распределительным шкафом, концентраторами и коммутаторами, которые управляют этими соединениями, выполнить проверку возможных дефектов связывающих их сетевых кабелей. Члены группы должны работать методично, пока не найдут причину возникших проблем. Централизованный мониторинг систем и управление событиями могут помочь найти причину проблем и сэкономить массу времени и усилий.

ПРИМЕЧАНИЕ. Управление событиями (event management) обычно реализуется с помощью специализированного продукта, осуществляющего сбор и анализ различных журналов со всех устройств в сети. Такой продукт выявляет шаблоны поведения и потенциально опасные действия, которые человек, скорее всего, пропустит, т.к. объем таких журналов обычно очень велик.

4.2. Отклонения от стандартов

В данном случае «стандарты» относятся к доступному уровню вычислительных возможностей и способам его измерения. Для каждого устройства могут быть установлены определенные стандарты: время нахождения в режиме онлайн, минимальное количество запросов, которые должны обрабатываться в единицу времени, максимальная полоса пропускания, счетчики производительности и т.п. Эти стандарты используются в качестве базового уровня, позволяющего установить факт наличия проблем с устройством. Например, если устройство обычно обрабатывает около 300 запросов в минуту, но вдруг оно начинает тратить на обработку этого количества запросов три минуты, сотрудники Департамента эксплуатации должны изучить это отклонение от стандарта, обычно выполняемого этим устройством. Причиной такого отклонения может быть неисправность устройства или его нахождение под DDoS-атакой. Другой причиной может быть легитимная деятельность, но выполняющаяся способами (или в условиях), которые не были предусмотрены при вводе устройства в эксплуатацию.

Иногда стандарты необходимо пересматривать, чтобы они отражали реальный взгляд на уровень обслуживания, который может обеспечить такое устройство. Если, к примеру, в сервере был заменен процессор с Pentium II на Pentium IV, был в четыре раза увеличен объем памяти и добавлены три дополнительных жестких диска, уровень сервиса для этого сервера следует пересмотреть.

4.3. Внеплановая перезагрузка системы

Начальная загрузка системы (IPL – Initial program load) – это термин, использующийся в среде мэйнфреймов, он обозначает загрузку ядра операционной системы в оперативную память компьютера. Загрузка операционной системы на персональном компьютере, является эквивалентом IPL. Это необходимая операция для подготовки компьютера к работе.

Сотрудники Департамента эксплуатации должны исследовать компьютеры, перезагружающиеся без явных причин. Это может указывать на серьезные проблемы операционной системы, или заражение компьютера вредоносной программой.

4.4. Идентификация и управление активами

Управление активами (asset management) – это управление тем, «чем компания владеет». В магазине розничной торговли, это может быть названо управлением запасами, которое будет являться частью ежедневных операций, обеспечивающих точность записей по продажам и данных систем учета, а также позволяющих выявить факты воровства. Те же самые принципы могут применяться к ИТ-среде, но в ней существует гораздо больше аспектов, помимо физического и финансового.

Для уверенности в установке безопасных настроек для аппаратного и программного обеспечения, сначала необходимо знать о том, что они вообще существуют в среде. Управление активами включает в себя получение полного инвентаризационного перечня аппаратного (включая системы и сети) и программного обеспечения, а также дальнейшую поддержку актуальности этого перечня.

Может показаться, что управление активами представляет из себя знание о том, что компания владеет 600 настольными компьютерами от одного производителя, 400 настольными компьютерами от другого производителя и 200 ноутбуками от третьего производителя. Но достаточно ли этой информации для управления конфигурациями и безопасностью этих 1200 компьютеров? Нет.

Немного углубимся в эту информацию. Достаточно ли будет знать, что эти 600 настольных компьютеров являются моделями 123 от производителя А, 400 компьютеров – моделями 456 от производителя В, 200 ноутбуков – моделями 789 от производителя С? Пока еще нет.

Чтобы иметь полное представление обо всех компонентах среды, которые могут быть субъектом рисков безопасности, необходимо иметь полную информацию о каждом аппаратном устройстве, операционной системе, сетевом устройстве (и его операционной системе), а также каждом приложении, работающем в этой среде. Даже прошивка сетевой карты, установленной в компьютере, может содержать уязвимости, и уж, конечно, драйвера устройств в операционной системе, которые и приводят к тому, что уязвимость сетевой карты может представлять риск нарушения безопасности. Операционные системы представляют из себя относительно хорошо известный и управляемый аспект рисков безопасности. Менее известным аспектом, важность которого постоянно растет, являются приложения (программное обеспечение). К примеру, приложение может содержать неактуальную и уязвимую версию Java Runtime Environment. Или оно может копировать системную библиотеку операционной системы в нестандартное (и неконтролируемое) место, что может привести к наличию уязвимости даже после установки вами патча (ведь патч применится только к оригинальной библиотеке, хранящейся в системной папке операционной системы). Эту уязвимую библиотеку может обнаружить злоумышленник и нарушить безопасность системы, воспользовавшись старым проверенным эксплойтом.

Управление активами требует знания всего – оборудования, прошивок, операционных систем, библиотек времени выполнения, приложений и отдельных библиотек – в масштабах всей среды. Очевидно, что только с помощью автоматизированных средств можно

выполнить это в полной мере.

Наличие полной инвентаризации всего, что есть во всей среде, является необходимым, но не достаточным. Есть один простой принцип безопасности: если необходимость в компоненте отсутствует, лучше удалить этот компонент. Если компонента нет в среде, он не может создать угрозу для ее безопасности. Однако не всегда это просто сделать. Например, в отдельных случаях такие компоненты могут быть встроены в системы и приложения. В таких случаях, необходимо контролировать их наряду со всеми остальными компонентами, чтобы обеспечить их безопасность.

Стандарты конфигурации – это ожидаемая конфигурация, на соответствие которой проверяется фактическая конфигурация. Любые отклонения от ожидаемой конфигурации должны быть расследованы, поскольку это означает, что либо ожидаемая конфигурация потеряла актуальность, либо, что контроль над средой не является достаточным для предотвращения несанкционированных (или просто незапланированных) изменений в среде. Автоматизированные средства управления активами могут сравнить ожидаемые конфигурации с фактическими настройками компонентов среды.

Возвращаясь к принципу простоты, лучше всего свести количество стандартов конфигураций к разумному минимуму, обеспечивающему потребности бизнеса. Процессы Управления изменениями или Управления конфигурациями, должны охватывать любые изменения в среде, что позволит обеспечить поддержку соответствия стандартам конфигураций. Сведение количества стандартов конфигураций к разумному минимуму, позволит снизить общую стоимость Управления изменениями.

4.5. Системные защитные меры

Системные защитные меры (system controls) является еще одной частью операционной безопасности. В операционной системе, должны быть реализованы определенные защитные меры, обеспечивающие выполнение команд в правильном контексте безопасности. В системе есть механизмы, ограничивающие выполнение определенных типов команд, такие команды могут выполняться только тогда, когда операционная система работает в привилегированном или административном режиме. Это защищает безопасность системы в целом и ее функционирование, помогая обеспечить стабильную и предсказуемую работу.

Должны быть разработаны операционные процедуры, которые определяют, из чего состоит правильная работа системы или ресурса. Они должны включать в себя последовательности действий при запуске и выключении системы, процедуры обработки ошибок и восстановления из надежного источника.

Операционная система не позволяет процессам с низким уровнем привилегий (как правило, это процессы прикладных программ) получить прямой доступ к оборудованию. Если программе нужно передать команды аппаратному устройству, она направляет соответствующий запрос процессу, имеющему более высокие привилегии. Для выполнения привилегированных аппаратных команд, процесс должен работать в ограниченном и защищенном режиме. Это является неотъемлемой частью архитектуры операционной системы, которая определяет, какие процессы могут выполнять какие команды, основываясь на управляющих таблицах операционной системы.

Многие команды ввода/вывода (I/O) определены, как привилегированные, они могут быть выполнены только процессами ядра операционной системы. Если прикладной программе нужно выполнить какую-либо операцию ввода/вывода, она должна обратиться к привилегированным процессам ядра операционной системы, которые работают на внутренних кольцах защиты. Эти процессы (называемые системными сервисами) либо разрешают процессу прикладной программы выполнить эти действия и временно повышают уровень его привилегий, либо они сами выполняют запрошенные действия от имени прикладной программы (более детально эти вопросы рассматриваются в Домене 03).

4.6. Доверенное восстановление

При возникновении сбоя или «зависания» операционной системы или приложения, система не должна переходить в небезопасное состояние. Чаще всего, причиной сбоя системы является то, что она столкнулась с чем-то, что было ей воспринято как небезопасное или непонятное, в связи с чем она решила прекратить свою работу и перезагрузить компьютер, а не продолжать выполнение текущих задач.

Реакцией операционной системы какой-либо сбой может быть одно из следующих действий:

- Перезагрузка системы
- Аварийный перезапуск системы
- Холодный запуск системы

Перезагрузка системы происходит после ее управляемого (штатного) завершения работы в ответ на сбой на уровне ядра (доверенной компьютерной базы). Если система сталкивается с некорректной структурой данных объекта или у нее заканчивается место в некоторых критических таблицах, может произойти перезагрузка системы. Это позволяет освободить ресурсы, и вернуть систему в стабильное и безопасное состояние.

Аварийный перезапуск системы происходит неуправляемым (нештатным) образом после системного сбоя. Такой сбой может произойти на уровне ядра или может быть вызван неисправностью системного диска, либо попыткой доступа непривилегированного пользовательского процесса к сегментам памяти ограниченного доступа. Система видит, что это небезопасное действие, но она не может остановить его и вернуться в безопасное состояние без перезапуска. При этом состояние ядра и пользовательских объектов может быть нарушено, а данные могут быть потеряны или повреждены. Поэтому система переходит в режим обслуживания и "откатывает" выполненные действия. Затем она возвращается в целостное и стабильное состояние.

Холодный запуск системы выполняется в случае возникновения неожиданного сбоя на уровне ядра или системного диска, при котором обычная процедура восстановления не может восстановить целостность системы. Система, ее ядро и пользовательские объекты могут быть разрушены и для их восстановления потребуются вмешательство пользователя или администратора.

Важно обеспечить, чтобы система не перешла в небезопасное состояние в случае возникновения таких проблем, она должна правильно выключаться и надлежащим образом восстанавливать безопасное и стабильное состояние (более подробно вопросы, связанные с ТСВ, компонентами ядра и его работой, рассмотрены в Домене 03).

Ссылки по теме:

- Configuration Management Plans: The Beginning to Your CM Solution, by Nadine M. Bounds and Susan Dart, Software Engineering Institute, Carnegie Mellon University
- Configuration Management Resources from the Georgia Institute of Technology

После системного сбоя

Рано или поздно любая система может выйти из строя, поэтому важно, чтобы обслуживающий ее персонал знал, как устранять в ней неполадки и восстанавливать ее работу. Ниже перечислены шаги, которые должны быть предприняты для этого:

1. **Войдите в систему в однопользовательском режиме или безопасном режиме (safe mode).** Когда происходит холодный запуск системы из-за неспособности системы автоматически восстановить свое безопасное состояние, в этом процессе должен принимать участие администратор. Система может быть загружена автоматически, но только в «однопользовательском режиме» (single user mode), либо должна быть

загружена вручную с помощью «Консоли Восстановления». В этих режимах система не должна запускать пользовательские и сетевые службы, файловые системы обычно не монтируются, доступна только локальная консоль. Поэтому администратор должен либо физически находиться рядом с консолью, либо для доступа к ней должны быть внедрены специализированные технологии удаленного доступа, такие как защищенное соединение по модему (dial-in/dial-back), подключенному к последовательному порту этой системы, либо сетевой переключатель KVM (Keyboard Video Mouse – клавиатура/видео/мышь) подключенный к консоли.

2. **Исправьте возникшую проблему и восстановите файлы.** В однопользовательском режиме администратор восстанавливает поврежденную файловую систему (такие повреждения могут произойти, например, в результате внезапного нештатного выключения системы), а затем пытается определить причину сбоя, чтобы предотвратить его повторение. Иногда администратору также требуется произвести откат или, наоборот, отметить откат изменений в базе данных или другом приложении в однопользовательском режиме. В других случаях это происходит автоматически, когда администратор переводит систему из однопользовательского режима в обычный режим, либо выполняется администратором вручную до того, как приложения и службы вернуться в нормальное состояние.
3. **Проверьте критичные файлы и работу системы.** Если проводится расследование причин внезапного выключения и предполагается, что произошло повреждение (например, из-за программного или аппаратного сбоя, выполненного пользователем/администратором изменения настроек или в результате атаки), администратор должен проверить содержимое конфигурационных файлов и убедиться, что системные файлы (программные файлы операционной системы, файлы совместно используемых библиотек, возможно, файлы приложений и т.д.) являются подлинными. Для проверки системных файлов могут использоваться их криптографические контрольные суммы, проверенные с помощью таких программ, как Tripwire. Администратор должен проверить содержимое системных конфигурационных файлов на соответствие документации по системе.

Проблемы безопасности

- **Изменение последовательности загрузки (C:, A:, D:) должно быть запрещено.** Чтобы обеспечить восстановление системы в безопасном состоянии, система должна быть настроена таким образом, чтобы не позволить злоумышленнику изменить последовательность загрузки системы. Например, на рабочей станции или сервере Windows, только уполномоченные пользователи (администраторы) должны иметь доступ к настройкам BIOS, в т.ч. к настройке перечня устройств, с которых допускается загрузка системы, и последовательности использования этих устройств. Если загрузка системы допускается только с диска C: (основной жесткий диск), и использование для этого никаких других жестких дисков и съемных носителей (например, дискет, компакт-дисков, USB-накопителей) не допускается, настройки системы должны запрещать пользователю (и злоумышленнику) добавлять устройства в перечень загрузки и изменять их последовательность в этом перечне. Если злоумышленник может изменить перечень устройств, с которых возможна загрузка, или их порядок, и может вызвать перезагрузку системы (что всегда возможно при физическом доступе к системе), он может загрузить компьютер со своего собственного носителя информации и провести атаку на программное обеспечение и/или данные системы.
- **Запись действий в системные журналы регистрации событий должно быть невозможно обойти.** Должна быть обеспечена сохранность системных журналов регистрации событий и файлов состояния системы (system state file) посредством

использования разграничения обязанностей и контроля доступа. Иначе пользователи (злоумышленники) смогут скрыть свои действия или изменить состояние, в котором система окажется после следующей перезагрузки. Если какой-либо из конфигурационных файлов может быть изменен неуполномоченным пользователем, и этот пользователь может вызвать перезагрузку системы, он сможет таким образом применить новые (вероятно небезопасные) настройки системы.

- **Принудительное завершение работы системы не должно допускаться.** Чтобы снизить возможности применения несанкционированно измененных настроек системы, а также возможности вызвать отказ в обслуживании путем неправильного завершения работы системы, только администраторы должны иметь возможность завершать работу критичных систем.
- **Не должно существовать возможностей перенаправить диагностические данные.** Диагностические данные системы могут содержать критичную информацию. Содержимое файлов журналов регистрации диагностических событий (включая вывод этой информации на консоль) должны быть защищены с помощью контроля доступа, запрещающего чтение этих данных кому-либо, кроме уполномоченных администраторов. Неуполномоченные пользователи не должны иметь возможности изменить место для записи диагностических журналов и консольного вывода.

4.7. Контроль входных и выходных данных

Исходящие из приложения данные имеют прямую связь со входящими данными. Поэтому входные данные должны контролироваться на предмет ошибок и подозрительного содержимого. Если кассир в продовольственном магазине за любую покупку в магазине пробивает чек на 1 рубль, магазин в итоге может потерять значительную сумму денег. Подобные действия могут быть неумышленными, вызванными неумением кассира пользоваться кассовым аппаратом, что потребует проведения обучения кассира, а могут быть целенаправленными, что потребует применения дисциплинарных мер.

Поскольку деятельность большинства современных компаний в большой степени зависит от компьютеров и приложений, обрабатывающих их данные, контроль входных и выходных данных является очень важным. В Домене 08 были рассмотрены различные варианты мошенничества, при которых пользователь вносит изменения в данные, хранящиеся в программе или генерируемые ей на выходе, как правило, с целью получения финансовой прибыли.

Сами приложения также должны быть разработаны таким образом, чтобы принимать только определенные типы входящих данных и выполнять определенные процедуры логического контроля для проверки полученных входных значений. Если приложение запрашивает у пользователя ввод стоимости покупаемого по ипотеке дома, а пользователь вводит 10 рублей, приложение должно повторно запросить у пользователя ввод этой информации, чтобы не тратить время впустую, обрабатывая ошибочные входные данные. Аналогично, если в поле ввода числовых значений пользователь ввел «Иван», приложение должно вывести соответствующее сообщение об ошибке и предоставить пользователю возможность исправить ошибку. Различные контроли входных и выходных данных рассматривались нами более подробно в Домене 09.

Все упомянутые в предыдущих разделах защитные меры, должны быть внедрены и функционировать на непрерывной, безопасной и предсказуемой основе, чтобы обеспечить правильную работу систем, приложений и среды в целом. Давайте рассмотрим несколько вопросов, которые могут вызвать проблемы, если не уделять им достаточно внимания.

- Все онлайн-транзакции должны записываться в журнал с указанием штампа времени.
- Данные, вводимые в систему, должны иметь правильный формат и должны

проверяться, на предмет отсутствия в них вредоносных (подозрительных) данных.

- Следует убедиться в том, что выходные данные безопасным образом сохраняются (передаются) в правильное место назначения.
 - Перед предоставлением критичных выходных данных, всегда должен требоваться подписанный запрос.
 - Баннер в начале и конце должен указывать, кому предназначены данные.
 - После создания выходных данных, должно быть реализовано надлежащее управление доступом к ним, независимо от типа их носителя (бумага, диск, лента и т.п.).
 - Если отчет не содержит никакой информации (пустой отчет), он должен содержать строку, типа: «нет результатов».

Некоторых людей смущает последний пункт. Возникает логичный вопрос: «Если нет информации для записи в отчет, зачем создавать отчет без информации?». Рассмотрим, например, следующую ситуацию. Каждую пятницу вы готовите и отправляете своему руководителю отчет по выявленным за неделю инцидентам безопасности и принятым мерам. Однажды в пятницу руководитель не получил отчета от вас. Ему придется идти к вам и спрашивать, почему нет отчета. А если бы он получил пустой отчет, он бы знал, что задача выполняется, просто инцидентов за неделю не произошло.

Другим типом ввода в систему могут быть компоненты ActiveX, плагины, обновления конфигурационных файлов или драйверы устройств. Лучше всего, если они имеют криптографическую подпись, поставленную доверенной стороной перед их распространением. Это позволит администратору вручную (или системе автоматически) проверить, что файлы действительно получены от доверенной стороны (производителя, продавца, поставщика) и не были изменены, прежде чем использовать эти файлы на системе, находящейся в промышленной эксплуатации. Например, Microsoft, начиная с операционной системы Windows 2000, ввела проверку цифровой подписи файлов драйверов (Driver Signing), с помощью которой операционная система предупреждает пользователя, если происходит попытка установить драйвер, который не был подписан доверенным субъектом, имеющим сертификат доверенного Удостоверяющего центра. Современные операционные системы Windows (в т.ч. Windows Mobile) по умолчанию настроены на предупреждение пользователя в случае попытки установки неподписанного программного обеспечения. Обратите внимание, что факт наличия подписи у дистрибутивного файла или драйвера, вовсе не означает, что оно является безопасным и надежным. Наличие подписи дает высокую степень гарантий, что программное обеспечение или драйвер исходят от доверенного производителя, но не более того. Если пользователь не доверяет субъекту (компании или разработчику), который подписал программное обеспечение (драйвер), либо если программное обеспечение (драйвер) не содержит никакой подписи, то это должно остановить пользователя от использования этого программного обеспечения (драйвера) до тех пор, пока их безопасность и надежность не будут подтверждена по другим каналам.

4.8. Укрепление систем

Используемые для обеспечения безопасности защитные меры и средства могут являться физическими, административными или техническими. Считается, что если к критичному, с точки зрения безопасности, объекту может быть получен несанкционированный физический доступ, невозможно обеспечить безопасность этого объекта (поэтому критичные данные на портативных носителях информации должны быть зашифрованы). Иными словами, *«если я смогу получить доступ к консоли компьютера, я могу стать его владельцем»*. Очевидно, что дата-центр компании должен быть хорошо защищен физически. Для этого может использоваться охрана, ворота, заборы, колючая проволока, освещение, замки на дверях и

т.д. Все вместе это создает сильную физическую безопасность периметра вокруг здания, где хранится ценная информация.

Через дорогу от этого дата-центра находится офисное здание, в котором сотни или тысячи сотрудников сидят изо дня в день, работая с ценной информацией на своих настольных компьютерах, ноутбуках и мобильных устройствах с использованием сетей различных типов. В идеальном мире, приложения и технологии доступа к информации надежно защищали бы данные от любых сетевых атак, однако мир не идеален, и специалисты по безопасности должны обеспечивать защиту ценных данных в реальном мире. Поэтому физические компоненты, из которых состоят сети и через которые передаются потоки ценных данных, также должны быть защищены.

- Коммутационные шкафы (wiring closet) должны быть заперты.
- Сетевые коммутаторы и концентраторы, которые не могут быть по тем или иным причинам размещены в закрытых коммутационных шкафах, должны помещаться в запертые коробки.
- Сетевые розетки в общедоступных местах (например, переговорные комнаты, общедоступные компьютеры и даже телефоны) должны быть сделаны физически недоступными.

Ноутбуки, USB-накопители, портативные жесткие диски, мобильные телефоны / коммуникаторы и даже MP3-плееры могут хранить большое количество информации, часть которой может быть критичной и очень ценной. Пользователи не должны оставлять такие устройства, на которых хранится ценная информация, без контроля, и надежно хранить их, когда они не используются активно. Ноутбуки часто пропадают на контрольно-пропускных пунктах в аэропортах; флэш-накопители очень маленькие и их нередко теряют или забывают, мобильные телефоны, коммуникаторы и MP3-плееры воруют каждый день. Но если мы все-таки обеспечили надежную физическую безопасность, нуждаемся ли мы в дополнительных технических мерах защиты? Да.

Приложение, которое не установлено на компьютере, или отключенная системная служба не могут быть атакованы. Но даже если системная служба отключена, отдельные ее компоненты могут содержать уязвимости, которыми могут воспользоваться профессиональные атакующие, поэтому лучше ненужные компоненты полностью удалять из системы (среды). Компоненты, от установки которых нельзя отказаться в процессе установки системы, и которые нельзя удалить в связи с их глубокой интеграцией в систему, должны быть отключены. Полномочия на их включение должны быть только у уполномоченного системного администратора. Каждое установленное приложение, и, в особенности, каждая работающая служба, должны быть указаны в общей базе данных Управления конфигурациями, чтобы иметь возможность отслеживать уязвимости в этих компонентах.

Компоненты, которые нельзя ни удалить, ни отключить должны быть настроены с использованием самых безопасных настроек, которые, однако, позволят системе эффективно работать для выполнения функций, ради которых эта система была внедрена. "Движки" баз данных, например, должны запускаться от имени непривилегированного пользователя, а не от имени root или SYSTEM. Если система запускает несколько прикладных служб, каждая из них должна работать под собственной учетной записью, т.к. при этом компрометация одной службы в системе не позволит получить доступ к другим службам в этой системе. По аналогии с ненужными службами, из системы должны быть, по возможности, удалены ненужные части отдельных служб, либо отключены, если их удаление невозможно.

Вопросы лицензирования

Компании имеют этическое обязательство использовать только легально приобретенные программные приложения. Разработчики программного обеспечения и группы их отраслевых представителей, такие как Business Software Alliance (BSA), используют агрессивную политику

против компаний, которые используют пиратские (нелегальные) копии программного обеспечения.

Компании несут ответственность за обеспечение того, чтобы в их корпоративной среде использовалось только легальное программное обеспечение, и чтобы лицензии реально соблюдались. Обязанности по контролю этого требования возлагаются на Департамент эксплуатации. Автоматизированные системы управления активами (asset management system), или системы управления системами на более высоком уровне, могут сформировать отчет об установленном программном обеспечении во всей среде, в том числе о количестве установок каждого из приложений. Это количество должно на регулярной основе (например, ежеквартально) сравниваться с перечнем приложений, легально приобретенных компанией, и количеством приобретенных лицензий для каждого из них. В случае выявления установленного приложения, для которого не была приобретена лицензия, либо при выявлении количества установок одного приложения, превышающего число приобретенных для него лицензий, должно проводиться расследование. При обнаружении в среде приложения, для которого не был соблюден установленный в компании процесс управления изменениями и процесс закупок, оно должно быть взято под контроль, а сотрудники подразделения компании, которые приобрели/установили приложение с нарушением установленного в компании порядка должны быть обучены и поставлены в известность как в отношении юридических рисков, так и в отношении рисков информационной безопасности, которые могут быть вызваны их действиями. Обычно в таких случаях руководитель соответствующего бизнес-подразделения должен подписать документ, указывающий, что он понимает возникающие риски, принимает их и берет на себя персональную ответственность в случае их возможной реализации.

В случае выявления приложений, в отношении использования которых нет обоснованных потребностей бизнеса, такие приложения следует удалить, а лиц, которые их устанавливали, следует обучить и предупредить, что в будущем подобные действия могут привести к более серьезным последствиям.

Компании должны регламентировать **политику использования систем** (acceptable use policy), в которой, в частности, указывается, какое программное обеспечение пользователи могут установить самостоятельно, и которая информирует пользователей о регулярно проводимых проверках соблюдения этой политики. Для предотвращения несанкционированной установки пользователями неразрешенного программного обеспечения, должны быть внедрены соответствующие технические защитные меры.

Часто о компаниях, использующих нелегальное программное обеспечение, сообщают недовольные сотрудники этих компаний (в качестве мести).

4.9. Безопасность удаленного доступа

Удаленный доступ является важным элементом обеспечения работы компании, он очень помогает, когда компания сталкивается с различными авариями и чрезвычайными ситуациями. Если произошедшая авария или региональная катастрофа не позволяет большому числу сотрудников выполнять работу на своем обычном рабочем месте, но дата-центр (основной или удаленный резервный) продолжает работать, удаленный доступ к компьютерным ресурсам может позволить компании продолжать выполнять многие из повседневных операций почти как обычно. Удаленный доступ может также способствовать снижению операционных расходов за счет сокращения офисных площадей, находящихся в собственности компании или арендованных, снижении потребностей в мебели, кондиционировании, отоплении, уборке, а также на парковке, поскольку многие сотрудники будут работать из дома. Удаленный доступ может быть единственным вариантом для организации работы «мобильной рабочей силы», например, специалистов по продажам, постоянно переезжающим из одного города в другой, которым необходим постоянный доступ к корпоративным ресурсам, чтобы встречаться с существующими и потенциальными клиентами.

Как и все остальное, что позволяет бизнесу работать и получать прибыль, удаленный доступ также приносит и риски. Является ли лицо, удаленно регистрирующееся на сервере компании тем, за кого себя выдает? Может быть, кто-то физически или с помощью электронных средств "смотрит через плечо" пользователя, или перехватывает данные, передаваемые по линиям связи? Имеет ли клиентское устройство, с которого он выполняет

удаленный доступ, безопасную конфигурацию или его безопасность нарушена установленным на нем шпионским программным обеспечением, троянскими программами и прочим вредоносным кодом?

Удаленный доступ – это «бельмо на глазу» подразделения информационной безопасности и подразделения эксплуатации почти любой компании. Ведь действительно небезопасно разрешать удаленное подключение ко внутренней корпоративной сети компании, ничего не зная об обеспечении безопасности подключающихся устройств – были ли на них установлены актуальные патчи, обновлены ли на них вирусные сигнатуры, не инфицированы ли они вредоносным кодом и т.д. Такие точки доступа нередко используются атакующими для получения доступа во внутреннюю сеть компании. В связи с необходимостью защиты удаленного доступа, производители занялись разработкой технологий контроля безопасности подключающихся удаленно систем и организации карантина для небезопасных систем.

Удаленное администрирование. Чтобы получить преимущества от использования удаленного администрирования, не принимая на себя чрезмерные риски, удаленное администрирование должно выполняться безопасным образом. Ниже приводятся некоторые рекомендации для этого:

- Команды и данные не должны передаваться в открытом виде (т.е. они должны быть зашифрованы). Для этого должен использоваться защищенный протокол, такой как SSH, а не Telnet.
- Действительно критичные системы должны администрироваться локально, а не удаленно.
- Только небольшому числу администраторов должны быть предоставлены права выполнять свою работу удаленно.
- Должна быть внедрена строгая аутентификация для выполнения любых административных действий.

5. Управление конфигурациями

Каждая компания должна иметь политику, определяющую порядок осуществления изменений, группу сотрудников, которые могут вносить изменения, руководителей, которые могут давать разрешение на внесение изменений, а также определяющую порядок документирования изменений и доведения информации о них до сведения сотрудников компании. Без такой политики, сотрудники могут без получения соответствующих санкций вносить изменения, о которых другие не будут знать. Это может привести к путанице и полному нарушению работы компании. В жестко регулируемых отраслях, таких как финансы, медицина и энергетика, существуют очень строгие правила относительно того, что можно делать, когда и при каких условиях. Эти требования предназначены для того, чтобы избежать проблем, которые могут оказать негативное влияние на широкие слои населения или на нижестоящих партнеров. Без строгого контроля и нормативных документов, в среде могут возникнуть уязвимости. Выявить и «откатить» изменения после того, как они уже были сделаны, может быть очень сложной и практически невозможной задачей.

Изменения могут происходить при внедрении новых технологий, приложений, устройств или при внесении изменений в уже работающие системы и инфраструктуру, при этом они могут затрагивать сетевые настройки, параметры систем, конфигурации приложений. Управление изменениями является важным процессом не только для среды, но и для продукта в ходе его разработки и в рамках его жизненного цикла. Изменения должны быть эффективными и упорядоченными, поскольку постоянное, необдуманное и нецеленаправленное внесение изменений может привести только к пустым тратам времени и денег.

Некоторые изменения могут привести к серьезным нарушениям работы сети и повлиять на доступность систем. Поэтому любые изменения должны быть тщательно продуманы, одобрены, при их реализации должен применяться структурированный подход. На случай непредвиденных негативных последствий, могут потребоваться планы восстановления с

помощью резервных систем (копий). Например, если в здании меняется источник энергии, должен быть резервный генератор на случай, если переход пройдет не так гладко, как планировалось. В случае если сервер меняется на сервер другого типа, вероятно возникновение проблем совместимости, которые могут не позволить пользователям получить доступ к отдельным ресурсам, поэтому предварительно должны быть созданы резервные копии и организован резервный сервер, что позволит обеспечить постоянную доступность и приемлемую производительность систем и ресурсов.

5.1. Процесс управления изменениями

В компании должен быть организован хорошо структурированный процесс управления изменениями, чтобы помочь сотрудникам продолжать эффективно выполнять свои обязанности при осуществлении различных изменений в среде. Этот процесс должен быть изложен в политике управления изменениями (change control policy). Хотя изменения бывают разными, стандартный список процедур может помочь сохранить контроль над этим процессом и обеспечить его выполнение предсказуемым образом. Перечисленные ниже шаги являются примером процедур, которые должны быть частью любой политики управления изменениями:

1. **Оформление запроса на изменение** (request for a change). Запрос должен быть направлен лицу (или группе лиц), ответственному за утверждение изменений и надзор за деятельностью по проведению изменений в среде.
2. **Утверждение (одобрение) изменений**. В запросе на изменение заказчик должен обосновать причины, по которым это изменение необходимо, и показать явные преимущества и возможные проблемы выполнения изменения. Иногда от заказчика может потребоваться проведение специального (дополнительного) исследования, либо предоставление большего объема информации перед тем, как изменение будет одобрено.
3. **Документирование изменений**. После утверждения изменения, оно должно быть записано в журнал изменений. По мере выполнения изменения, информация в журнале должна дополняться.
4. **Тестирование и представление**. Изменение должно быть в полном объеме проверено, чтобы выявить любые непредвиденные результаты. В зависимости от сложности изменения и принятых в компании правил, может потребоваться представить информацию по изменению и его реализации Комитету по управлению изменениями (change control committee). Это поможет с разных сторон проанализировать цели и результаты изменения, а также возможные последствия.
5. **Реализация**. После того, как изменение полностью протестировано и утверждено, должен быть разработан график, содержащий предполагаемые этапы реализации (внедрения) изменения и необходимые контрольные точки. Выполнение этих шагов должно документироваться, процесс их выполнения должен контролироваться.
6. **Отчет руководству об изменении**. Должен быть разработан и направлен руководству полный отчет, резюмирующий информацию по выполненному изменению. Этот отчет может формироваться на периодической основе, чтобы руководство постоянно имело актуальную информацию и было уверено в выполнении этого процесса.

Как правило, эти шаги выполняются для крупных изменений, которые затрагивают всю компанию. Такие изменения обычно дороги и могут иметь долгосрочное влияние на компанию. Однако и менее значительные изменения должны выполняться в рамках определенного процесса управления изменениями. Если на сервер нужно установить патч, было бы не очень разумно просто подойти к нему и запустить установку патча, без проведения надлежащего тестирования, без получения разрешения руководителя

Департамента ИТ или администратора сети, без проведения предварительного резервного копирования и без разработки аварийного плана действий на случай, если патч окажет негативное влияние на работу сервера. Разумеется, что все изменения должны документироваться.

Как было сказано ранее, крайне важно, чтобы Департамент эксплуатации разработал и утвердил аварийные планы заранее, до того момента, когда изменения будут внесены в систему или сеть. Нередко изменения вызывают проблемы, которые не были выявлены раньше, до начала процесса их внедрения. Многие сетевые инженеры сталкивались с проблемами, вызванными плохо разработанными «исправлениями» или патчами, которые нарушали работу отдельных компонентов системы. Аварийный план нужен для того, чтобы даже в случае возникновения подобных проблем, продуктивности работы компании не был нанесен ущерб. В этом плане указываются действия, которые нужно будет выполнить, чтобы вернуть систему в первоначальное состояние, в котором она находилась до внедрения изменения.



5.2. Документация по управлению изменениями

Отсутствие документирования производимых в системах и сетях изменений приводит к различным неприятностям, т.к. никто не помнит, например, что было сделано на том сервере в DMZ шесть месяцев назад или как была восстановлена работа основного маршрутизатора, когда на нем произошел сбой в прошлом году. Изменения в настройках программного обеспечения и сетевых устройств в большинстве сред происходят довольно часто, сохранение детальной информации обо всех этих изменениях невозможно организовать без ведения журнала всех выполненных изменений.

В компании может происходить множество различных изменений, некоторые из которых приведены ниже:

- Установка новых компьютеров
- Установка новых приложений
- Внесение изменений в конфигурации
- Установка патчей и обновлений
- Интегрирование новых технологий
- Обновление политик, процедур и стандартов
- Внедрение новых правил и требований

- Выявление проблем в сети или системах и их решение
- Внесение изменений в сетевые конфигурации
- Установка в сети новых сетевых устройств
- Приобретение компании другой компанией или ее объединение с другой компанией

Этот список можно продолжать и дальше, он может быть высокоуровневым или, наоборот, детальным. Многие компании сталкивались с проблемами, которые оказывали существенное влияние на функционирование сети и продуктивность работы сотрудников. Сотрудники Департамента ИТ могут несколько часов или дней пытаться решить возникшую проблему методом проб и ошибок. Но если никто не задокументирует этот инцидент и принятые для его исправления действия, когда решение будет наконец найдено, компании, скорее всего, понадобится не меньше времени на решение аналогичной проблемы, которая возникнет через шесть месяцев.



6. Контроль носителей информации

Необходим контроль носителей информации и других устройств, которые используются в рабочей среде, для обеспечения их сохранности, а также защиты целостности, конфиденциальности и доступности хранящихся на них данных. Под «носителем информации» в данном разделе мы будем понимать как электронные (дискеты, компакт-диски, ленты, флеш-накопители и т.д.), так и неэлектронные (бумажные) носители информации. Мы будем рассматривать носители информации до записи на них информации, в процессе ее хранения и после удаления, а также при подключении к системе и после отключения от нее.

Существуют различные виды контроля носителей информации. Во-первых, это меры, предотвращающие несанкционированный доступ (защищающие конфиденциальность), которые могут быть, как обычно, физическими, административными и техническими. Если необходимо надежно защитить от несанкционированного доступа ленты, на которых хранятся резервные копии баз данных компании, нужно организовать их хранение в месте, в которое имеют право доступа только уполномоченные лица – например, в запертой серверной комнате или специальном территориально удаленном хранилище. Если носители информации должны быть защищены от воздействий окружающей среды, таких как

влажность, температура, пожар, стихийные бедствия (поддержка доступности), носители информации должны храниться в негорючем сейфе, установленном в помещении с регулируемым климатом. Более подробно эти вопросы были рассмотрены в Домене 04.

Компания может организовать библиотеку носителей информации и назначить библиотекаря, ответственного за их защиту. В таком случае, все описанные в этом разделе обязанности по защите конфиденциальности, целостности и доступности носителей информации возлагаются на библиотекаря. Пользователю может потребоваться отдельный носитель информации в библиотеке, но ему не нужен постоянный доступ ко всем носителям информации в библиотеке. Обычно в библиотеку помещаются, кроме всего прочего, дистрибутивные носители для лицензионного программного обеспечения. Использование библиотеки позволяет организовать учет (журнал регистрации) использования носителей информации, что может являться одним из подтверждений соблюдения компанией принципа должной осмотрительности при выполнении лицензионных соглашений, а также помочь в обеспечении защиты конфиденциальной информации (например, персональных данных, данных банковских карт, медицинской информации и т.п.).

Носители информации должны быть четко промаркированы и учтены, их целостность должна проверяться, данные с них должны уничтожаться надежным способом, если они больше не нужны. После значительных обновлений компьютерного парка компании, когда многие рабочие станции и серверы заменяются новыми, широко распространенной ошибкой является неправильная утилизация выведенного из эксплуатации оборудования. Старые компьютеры вместе с жесткими дисками просто выносятся через заднюю дверь вместе со всеми данными, на защиту которых компания потратила столько времени и средств. Это приводит к риску компрометации критичной для компании информации, записанной на носителях информации в этих старых компьютерах. Это нарушает правовые, договорные и этические обязательства компании. Нельзя забывать о необходимости надежного уничтожения информации с носителей, выводимых из эксплуатации.

Очистка носителя от записанной на нем ранее информации, называется «**вымарыванием**» (sanitization). В военных и правительственных классифицированных системах это означает, что информация стирается таким образом, чтобы ее нельзя было восстановить с помощью стандартных средств операционной системы и доступного программного обеспечения, предназначенного для проведения криминалистической экспертизы / восстановления данных. Если носитель информации будет и дальше использоваться в той же среде, для тех же целей, людьми, имеющими аналогичный уровень допуска, достаточно просто стереть информацию с носителя (или форматировать его) перед выдачей другому пользователю.

Очистка (purging) – это полное уничтожение информации, не позволяющее восстановить данные даже при исследовании носителя информации в специализированной лаборатории. Очистка требуется в тех случаях, когда носитель информации покинет пределы контролируемой зоны, в которой разрешен доступ к содержащейся на нем информации (например, он будет использоваться для других целей, другим подразделением компании).

Вымарывание или очистка информации с носителей информации могут проводиться несколькими способами: **обнуление** (zeroization) (перезапись специальным шаблоном, после которой ранее записанные на носителе данные практически невозможно было извлечь), **размагничивание** (degaussing) (магнитное скремблирование шаблонов на ленте или диске, которые замещают собой хранившуюся на них информацию), и **уничтожение** (destruction) (измельчение, дробление, сжигание). Простое удаление файлов с носителя информации средствами операционной системы, в действительности не стирает данные, а только удаляет указатели на его области, где эти данные хранились. При этом сами данные все еще находятся на носителе информации. Удаленные таким образом файлы можно восстановить специальными утилитами (например, если они были удалены случайно). Но даже простая перезапись на носитель информации новых данных или его форматирование не могут

исключить возможность восстановления ранее записанной информации. Именно поэтому для стирания критичной информации требуется обнуление или использование безопасных алгоритмов перезаписи. Если по какой-либо причине часть критичных данных не может быть удалена с носителя информации, сам носитель информации должен быть уничтожен физически.

Не все методы стирания/очистки применимы для любых носителей информации (например, оптические носители не подвержены размагничиванию). Вероятность того, что восстановлением информации будет заниматься достаточно мотивированный и способный противник, нельзя недооценивать или игнорировать. Для наиболее ценных данных коммерческих компаний, классифицированных данных военных организаций, а также для данных, вопросы защиты которых регулируются законодательством, необходимо соблюдать установленные требования по удалению информации.

Остаточные данные (data remanence) – это остаточное физическое представление информации, которая была записана на носитель информации, а затем удалена каким-либо способом. Этой остаточной информации может быть достаточно для того, чтобы реконструировать удаленные данные и восстановить их в читаемом виде. Это может представлять угрозу безопасности компании, которая считает, что она надежно удалила конфиденциальную информацию из своего носителя. Если носитель информации передается для использования другому лицу (*повторное использование объекта*), это лицо может получить несанкционированный доступ к критичным данным.

Если носитель информации не содержит конфиденциальной или критичной информации, простой перезаписи или удаления файлов может быть достаточно (более подробно эти вопросы рассмотрены в Домене 02).

Руководящим принципом при принятии решения о выборе необходимого метода стирания данных, является обеспечение того, чтобы стоимость восстановления данных потенциальным врагом превышала ценность этих данных. Информация, позволяющая «попить» компанию (или целую страну) имеет настолько высокую ценность, что оправданным является полное уничтожение носителей информации, хотя при этом компания будет тратить деньги на сам процесс уничтожения носителя, и будет нести убытки из-за невозможности его повторного использования. Для менее критичной информации, в большинстве случаев достаточно выполнения нескольких перезаписей данных. Каждая компания должна определить ценность своих данных, а затем выбрать наиболее оптимальный метод их уничтожения/удаления.

Мы рассмотрели методы безопасной очистки и вымарывания данных с электронных носителей информации. Но информация может быть сохранена на неэлектронных видах носителей информации, таких как бумага или микрофильмы, она также нуждается в безопасном уничтожении. Разгребание мусора (dumpster diving) – это поиск в мусоре компании или конкретного человека ценных сведений, которые были просто выброшены, а не уничтожены безопасным способом (например, путем сжигания или измельчения).

Управление носителями информации (в библиотеке или без нее) выполняет следующие задачи:

- **Отслеживание** (ведение журнала аудита), кто владеет каждым носителем информации в любой момент времени. При этом создается журнал того же вида, как и при журналировании событий программным обеспечением, что позволяет при расследовании определить, где определенная информация находилась в любой момент времени, кто получал ее, на основании чего (для особо критичной информации). Это дает возможность при проведении расследования более точно определить место и время нарушения и сосредоточить внимание на небольшой группе лиц, потенциально имеющих отношение к этому нарушению (или имеющих

информацию о нем).

- **Реализация эффективного управления доступом** с целью ограничения доступа к носителям информации и их предоставления только тем людям, которым разрешил доступ к соответствующему носителю информации владелец этого носителя (или владелец записанной на него информации), а также с целью соблюдения мер безопасности, основанных на классификации носителей информации (или самой записанной на них информации). Некоторые носители информации могут требовать «специального обращения», что может быть вызвано их физическими особенностями или характером записанной на них информации. Все сотрудники компании, которым разрешен доступ к носителям информации, должны быть обучены правилам использования соответствующих носителей информации. Примером специального обращения, может быть использование носителей классифицированной информации, которые могут покидать помещение библиотеки (или иного места, в котором они обычно хранятся) только под физической охраной, даже если они не покидают пределы здания. Управление доступом включает в себя *физические* (замки на дверях, запираемые шкафы или сейфы), *технические* (управление доступом, авторизация) и *административные* (регламентирующие документы, инструкции) меры и средства. Кроме того, нужно учитывать возможное изменение формата данных и их носителя, например, при печати электронных данных на бумаге. Данные по-прежнему нуждаются в защите, независимо от того, в каком формате и на каком носителе они находятся. Процедуры обеспечения безопасности данных должны включать в себя соответствующие правила, позволяющие сохранить безопасность данных на необходимом уровне. Например, при отправке конфиденциальной информации по почте, содержащие эту информацию документы должны быть упакованы в светонепроницаемый, защищенный от бесконтрольного вскрытия конверт.
- **Отслеживание количества и мест хранения резервных копий** (как на внутренних, так и на внешних площадках). Это необходимо для обеспечения своевременного уничтожения информации, достигшей конца своего жизненного цикла, для оценки мест хранения и доступности информации при проведении аудита, а также чтобы найти резервные копии информации, если первичный источник информации утрачен или поврежден.
- **Документирование истории изменений статуса информации, записанной на носителях информации.** Например, если определенная версия приложения, хранящегося в библиотеке, была признана устаревшей, должно быть зафиксировано, что это устаревшая версия приложения и она более не используется (за исключением случаев, когда по тем или иным причинам потребуется именно эта версия). Даже если носитель информации или его содержимое больше не нужны, соответствующая запись в журнале об этом носителе, времени и способе уничтожения информации с него, могут быть полезны для демонстрации должной осмотрительности.
- **Создание условий, исключающих негативное воздействие на носители информации условий окружающей среды.** Любой из видов носителей информации может быть подвержен негативному влиянию определенных факторов окружающей среды. Например, любой носитель информации может быть уничтожен или поврежден при пожаре, большинство носителей информации подвержены негативному влиянию жидкостей, дыма и пыли. Магнитные носители восприимчивы к сильным магнитным полям. Магнитные и оптические носители информации чувствительны к колебаниям температуры и влажности. Библиотека носителей информации или любое другое помещение, в котором хранятся носители информации, должны быть физически построены таким образом, чтобы внутри них поддерживались условия, безопасные для хранимых в них носителей информации.

Условия окружающей среды в таких помещениях должны непрерывно контролироваться, чтобы они не выходили за пределы допустимого диапазона соответствующих параметров. Библиотеки носителей информации особенно полезны при необходимости хранения большого объема информации в определенных климатических условиях. Это позволит ограничить количество помещений, в которых требуется обеспечить строго определенные условия окружающей среды, что приведет к снижению стоимости необходимого климатического оборудования и упростит управление носителями информации, которые будут храниться в одном (или нескольких) физических местах. При этом удельная стоимость хранения одного носителя информации будет минимальна.

- **Обеспечение целостности носителей информации**, путем выполнения проверок их работоспособности, учитывающих тип носителя и соответствие условий окружающей среды установленным требованиям. Кроме того, должны контролироваться сроки использования носителей информации и при достижении срока, после которого информация на носителе может быть повреждена, ценная информация с таких носителей должна переноситься на новые носители информации. У любого типа носителей информации есть установленный срок эксплуатации (при соблюдении определенных условий хранения), по завершении которого нельзя рассчитывать на надежное хранение информации на этом носителе. Например, срок хранения информации на обычных, серийно выпускаемых компакт-дисках, которые хранятся с соблюдением установленных требований к окружающей среде, составляет не менее десяти лет, тогда как дешевый компакт-диск, стоящий на полке в домашнем офисе, может стать ненадежным уже через год. Для всех типов носителей информации, используемых в компании, должен быть установлен и документирован ожидаемый срок эксплуатации (желательно устанавливая его с запасом). Если записываемая на носитель информация должна храниться дольше, чем срок службы этого носителя, по достижении срока эксплуатации носитель должен быть выведен из эксплуатации, а информация должна быть перезаписана на новый носитель информации (не обязательно того же типа). Также следует принимать во внимание доступность оборудования, необходимого для чтения носителя информации в течение всего срока хранения информации на нем. Даже если носитель информации может надежно хранить данные десятилетиями, то если через десять лет у компании не останется ни одного работающего устройства, которое может прочитать информацию с такого носителя, то его надежность уже не будет иметь никакого значения. Помимо этого, для сохранения целостности содержимого носителя информации (если записанная на нем информация является крайне ценной или у компании есть обязательства по сохранению этой информации), содержимое носителя информации должно быть подписано криптографической подписью, правильность которой должна проверяться на регулярной основе.
- **Инвентаризация носителей информации должна проводиться на регулярной основе**, чтобы как можно раньше обнаружить факты утраты или подмены отдельных носителей информации. Это может снизить размер ущерба посредством предотвращения аналогичных проблем в отношении других носителей информации. Инвентаризация является необходимой частью жизненного цикла управления носителями информации, которая помогает проверить достаточность применяемых механизмов защиты.
- **Выполнение мероприятий по безопасной утилизации**. Утилизация производится после того, как информация теряет свою ценность и требуется уничтожить информацию (или сам носитель). Безопасная утилизация информации (носителя информации) может существенно увеличить расходы на управление носителями информации. Однако учитывая, что не вся информация должна безопасно

уничтожаться в конце ее жизненного цикла, можно сократить долгосрочные операционные расходы компании. Знание о том, что определенная информация должна быть утилизирована безопасным образом, снижает вероятность того, что носитель информации может быть просто выброшен в корзину, а потом найден кем-то, кто опубликует его содержимое и поставит компанию в неловкое положение или будет шантажировать ее за допущенное нарушение безопасности и неправильное уничтожение информации. Библиотекарь, который обеспечивает хранение в библиотеке носителей информации, отвечает за определение времени жизни и уничтожения этой информации. При принятии таких решений, компания должна учитывать полезное для бизнеса время жизни информации, правовые и нормативные ограничения и требования. Если законодательство требует хранить определенную информацию дольше, чем она нужна компании, утилизация такой информации может включать в себя перемещение информации в архив и удаление ее из библиотеки, для которой обеспечивается непрерывная доступность информации, что требует повышенных затрат по сравнению с хранением информации в архиве, в котором информация хранится надежно, но для ее извлечения требуется больше времени.

- **Внутренняя и внешняя маркировка** каждого носителя информации в библиотеке должна включать:
 - Дату создания
 - Срок хранения
 - Уровень классификации
 - ФИО записавшего информацию на носитель
 - Дату уничтожения
 - Название и версию

Все эти задачи, перечисленные выше, вместе реализуют полный жизненный цикл носителей информации, и являются необходимой частью полного жизненного цикла информации, хранящейся на них.

Какие обязанности возлагаются на библиотекаря?

- Маркировка
- Ведение журналов
- Проверка целостности
- Защита от несанкционированного физического доступа
- Защита от воздействий окружающей среды
- Передача
- Утилизация

7. Утечки данных

Утечка персональных данных может привести к большим финансовым потерям. Исследование Ponemon Institute, проведенное в 2008 году, показало, что в среднем утечка одной записи обходится компаниям в 197 долларов США. Это включает в себя расходы на проведение расследования, штрафы, снижение доходов, информирование пострадавших и возмещение им прямого ущерба. В дополнение к финансовым потерям, может быть нанесен значительный ущерб репутации компании. Люди, чья информация была похищена, могут столкнуться с «кражей личности».

Наиболее распространенной причиной утечек информации является недостаток дисциплины среди сотрудников. В подавляющем большинстве случаев, утечки информации были

вызваны халатностью сотрудников (источник: исследование InfoWatch за 2009 год).



Чаще всего халатное обращение с данными выражается в неправильном использовании и удалении информации. Например, сотрудник компании может скопировать данные с защищенной корпоративной системы на свой домашний компьютер (гораздо менее безопасный), чтобы поработать дома. Другой проблемой может быть кража ноутбука, на котором была записана в открытом (незашифрованном) виде критичная информация. Такие кражи очень часто происходят в аэропортах, гостиницах, такси. Еще одной проблемой, которая может привести к утечке информации, является использование технологий, которые могут быть неприемлемыми в отдельных случаях, или простая небрежность. Примером может быть запись конфиденциальной информации на носитель, предназначенный для хранения открытой информации, который не будет уничтожен при выведении из эксплуатации.

Конечно, легко винить сотрудников за ненадлежащее использование информации, которое привело к нарушению ее безопасности, однако в этом виноваты не только они. Сотрудники выполняют свою работу, и их понимание этой работы почти полностью основано на том, что работодатель сообщил им о ней. То, что говорит один сотрудник другому, не является эффективным ограничением и не может заменить «должностную инструкцию». Должна существовать обратная связь, которую сотрудник получает ежедневно, а ежегодно должен производиться анализ работы каждого сотрудника. Если компания не включает вопросы безопасности в свои процедуры коммуникаций с сотрудниками, их обучения, аттестации, а также в процедуры расчета заработной платы и премий, сотрудники не будут считать, что безопасность является частью их работы.

Увеличившаяся сложность среды и появление новых типов носителей информации, требуют уделения большего внимания вопросам обучения и повышения осведомленности персонала, без чего невозможна надежная защита информационных активов компании.

Также нужно учитывать, что никакие политики и тренинги не остановят наиболее «продвинутых» сотрудников от использования новейших устройств и технологий, которые пока не были интегрированы в среду компании и для которых, возможно, пока нет средств эффективной защиты в корпоративной среде (исключением тут могут быть правительственные и военные организации). Компании должны как можно раньше узнавать о новых технологиях, и интересоваться, как сотрудники используют их (хотят использовать) в корпоративной среде. Простое «нет!» не остановит сотрудника от использования коммуникатора, USB-накопителя, либо отправки по электронной почте корпоративных данных на свои личные адреса, чтобы поработать с ними вне офиса. Компании должны предусматривать технические средства безопасности, позволяющие выявлять и/или предотвращать такие действия. Таковыми техническими средствами может быть, например, блокировка портов ввода-вывода компьютера или установка специальной системы защиты от утечек информации (DLP), которая не позволяет сотруднику переписать конфиденциальные данные компании на личный носитель информации или отправить их себе по электронной почте.

Крупнейшие утечки персональных данных (по количеству скомпрометированных записей) за 2009 год приведены ниже (по данным исследования InfoWatch).

Число записей	Страна	Краткое описание инцидента
76 млн.	США	Работники Национального архивного агентства (NARA) отправили в ремонт диск с БД ветеранов (пенсионеров), не удалив данные с диска.
62 млн.	Великобритания	Девять работников британского Департамента труда и пенсий (DWP) уличены в попытках неправомерного доступа к БД департамента, где хранятся персональные данные 62 миллионов человек (включая 12 млн. детей) – почти всех жителей страны.
32,6 млн.	США	Скомпрометированы (через SQL Injection) более 32 миллионов учетных записей в результате атаки на сайт RockYou.com, который предоставляет различные услуги социальным сетям, в том числе, Facebook и MySpace.
7,5 млн.	Германия	Найдена уязвимость в социальной сети StayFriends GmbH (www.stayfriends.de), которая позволяла получить доступ к персональным данным всех участников
6 млн.	Великобритания	Фирмы (в частности, Castrol) использовали для своих рекламных кампаний данные автовладельцев, которые были получены сомнительным способом из государственной БД регистрации автотранспорта.
2,5 млн.	Великобритания	Злоумышленникам удалось получить доступ к БД национальной медицинской службы (NHS), где хранятся данные на пациентов.
1,5 млн.	Япония	Служащий несанкционированно скачал из служебной БД и унёс данные клиентов; часть из них он успел продать.
807 тыс.	США	Утраченная архивная лента содержала данные о подозреваемых лицах за 12 лет, включая номера соцстрахования.

8. Доступность сети и ресурсов

Одним из основополагающих сервисов в триаде услуг безопасности, является *доступность* (двумя другими сервисами являются *конфиденциальность* и *целостность*). Важность доступности сети и ресурсов часто недооценивают, пока она не будет нарушена.

Администраторы и инженеры должны выполнять эффективное резервное копирование информации и обеспечивать наличие избыточных систем. Это позволит сохранить возможность выполнения компанией наиболее критичных из своих функций, даже если возникнет авария или сбой.

Сеть должна надлежащим образом поддерживаться, что позволит обеспечить ее доступность всегда, когда она будет необходима. Например, правильно должен быть выбран тип кабелей в соответствии с требованиями используемой среды и технологий, длина непрерывных сегментов кабеля не должна превышать рекомендуемые значения. Старые кабели должны быть заменены более новыми, должна выполняться периодическая проверка возможных обрывов и неисправностей кабелей.

Большинство сетей используют технологию Ethernet, которая очень устойчива к сбоям. Token Ring также проектировался с учетом обеспечения отказоустойчивости, но он хорошо работает только тогда, когда все компьютеры, подключенные к этой среде, настроены и работают правильно. Если хотя бы одна сетевая карта настроена на использование другой скорости, отличающейся от скорости работы сетевых карт других компьютеров, подключенных к среде Token Ring, это может привести к нарушению работы всего кольца. Если две системы имеют одинаковые MAC-адреса, будет нарушена работа всей сети. Такие вопросы необходимо учитывать при поддержке существующей сети компании.

Аналогично выбору варианта уничтожения данных, при выборе решения для резервного копирования (и других решений, обеспечивающих доступность) необходимо сбалансировать уровень критичности восстановления информации в заданные сроки со стоимостью необходимого для этого решения (включая стоимость сопровождения такого решения).

- **Избыточное оборудование, готовое для «горячей замены»**, сохраняет высокую доступность информации, обеспечивая наличие нескольких копий информации («зеркалирование») или достаточное количество дополнительной (избыточной) информации, позволяющей восстановить исходную информацию в случае ее частичной утраты (четность, коррекция ошибок). «Горячая замена» позволяет администратору заменять вышедший из строя компонент не останавливая работу системы и не прерывая доступность информации – при этом, обычно, несколько снижается производительность, но отключения системы не происходит.
- **Отказоустойчивые технологии** поддерживают доступность информации не только при сбоях отдельных накопителей информации, но даже при сбоях целых систем. Обеспечение отказоустойчивости является одним из самых дорогих среди возможных решений, оно оправдано только для самой критичной информации. В любых технологиях рано или поздно происходят различные сбои. Высокая стоимость отказоустойчивых решений может быть оправдана для компаний, которые понесут непоправимые потери (или многомиллионные убытки) от любого незапланированного, даже кратковременного простоя.
- **Соглашения об уровне сервиса** (SLA – Service level agreements) помогает поставщикам услуг, которыми могут быть как внутренние ИТ-службы, так и аутсорсинговые компании, определить оптимальный вариант технологий обеспечения доступности. На основании этого решения может быть определена стоимость услуг или бюджет ИТ. Заключение SLA с бизнесом не менее полезно и для самого бизнеса. Некоторые компании провели такого рода самоанализ, помогающий бизнесу понять реальную ценность его информации, но многие этого не сделали, и они вынуждены проходить через это упражнение каждый раз в процессе формирования бюджета своей внутренней ИТ-службы или внешней аутсорсинговой компании.
- **Целостные операционные процедуры** также необходимы для поддержания доступности. Наиболее надежное оборудование с максимальной избыточностью и отказоустойчивостью, предназначенное для восстановления систем в кратчайшие сроки, может оказаться пустой тратой денег, если операционные процедуры, обучение и непрерывное улучшение не являются частью операционной среды: одно случайное нажатие не на ту кнопку администратором может остановить работу самой надежной системы.

8.1. Среднее время безотказной работы

Среднее время безотказной работы (MTBF – Mean Time Between Failures) – это оценочная «продолжительности жизни» оборудования, которая рассчитывается производителем оборудования или третьей стороной. Значение MTBF нужно для того, чтобы приблизительно знать, когда устройство необходимо будет заменить. Это значение используется в качестве ориентира для оценки среднего времени работы компонентов системы до момента их выхода из строя. Значение MTBF рассчитывается на основе исторических данных, либо научных оценок производителей.

Компании, контролирующие реальные значения MTBF для используемых в своей среде устройств, могут выявить типы устройств, которые выходят из строя чаще, чем среднее время, обещанное их производителями, и принять соответствующие меры – например, заранее связаться с производителем и по гарантии заменить их на новые устройства до того момента, как они начнут массово выходить из строя.

Что это означает на самом деле? Значение MTBF может быть обманчиво. Оставим в стороне вопросы о том, являются ли правдоподобными предсказания производителей в отношении времени безотказной работы их устройств. Рассмотрим настольные компьютеры, на которых установлен один жесткий диск, для которого производитель определил MTBF равным 30000 часов. Разделим это время на количество часов в году: $30000 / 8760 =$ чуть более трех лет. Это

говорит о том, что жесткий диск этой модели отработает около трех лет, после чего он выйдет из строя (в среднем). Оставим в стороне вопросы воздействия окружающей среды офиса, в котором находится компьютер – температура, влажность, удары и пролитый кофе. Установим в компьютер второй такой же жесткий диск. Теперь вероятность отказа удвоилась – теперь существует два шанса, что в течение трехлетнего периода возникнет неисправность жесткого диска в этом компьютере. Если экстраполировать этот подход к дата-центру компании, в котором работает тысяча жестких дисков, становится ясно, что бюджет на замену жестких дисков нужно выделять на каждый год, а также обеспечивать хранение избыточных копий данных для защиты важной информации.

8.2. Среднее время восстановления

Среднее время восстановления (MTTR – Mean Time To Repair) – это время, которое потребуется для возврата отремонтированного устройства обратно в работу. Для жестких дисков, работающих в дисковом массиве, в котором используется избыточное количество дисков, значением MTTR будет являться промежуток времени между моментом выхода жесткого диска из строя до момента, когда кто-то заметит это и заменит отказавший диск, а массив закончит перезапись информации на новый диск. Для жестких дисков в настольных компьютерах, для которых избыточность обычно не обеспечивается, значением MTTR является промежуток времени между моментом, когда пользователь, изрыгая проклятия, звонит в службу технической поддержки (Help Desk), и моментом, когда жесткий диск в компьютере заменен, на него установлена операционная система и необходимое программное обеспечение, а также восстановлены с резервных копий данные пользователя. В этом случае MTTR может измеряться днями. В случае незапланированной перезагрузки компьютера, MTTR будет равен времени, от момента сбоя системы до момента, когда она перезагрузилась, проверила состояние файловой системы, пользователь перезапустил все нужные ему приложения, они проверили целостность своих данных и возобновили обработку транзакций. Для качественного оборудования, на котором работает хорошо управляемая операционная система и программное обеспечение, это может быть всего несколько минут. Для обычных потребительских систем, не имеющих высокопроизводительных журналирующих файловых систем и баз данных, это могут быть часы и даже дни, если не сработают автоматические процедуры восстановления или «отката» и придется восстанавливать систему вручную.

- MTTR может относиться к ремонту компонентов или устройств или их замене, а также может быть связан с SLA с поставщиком.
- Если MTTR является слишком высоким для критичных устройств, следует использовать избыточность.

Значения MTBF и MTTR, указываемые производителями, могут быть полезны при выборе новых систем и оценке затрат на них. Системы, для которых допустимы и не нанесут существенного вреда кратковременные перерывы в работе, могут быть построены из недорогих компонентов с низким ожидаемым значением MTBF и высоким MTTR. Высокое значение MTBF часто сопровождается более высокими ценами. Для систем, перерывы в работе которых недопустимы, необходимы резервные компоненты. Если для системы даже кратковременные простои ведут к значительному ущербу и небольшие периоды отключения, необходимые для замены вышедших из строя компонентов, оказываются недопустимыми, такие системы требуют обеспечения отказоустойчивости.

8.3. Единая точка отказа

Единая точка отказа (single point of failure) создает много потенциальных рисков для сети, поскольку выход из строя одного устройства приводит к негативному влиянию на целый сегмент или даже всю сеть. Устройствами, которые могут быть единой точкой отказа, являются межсетевые экраны, маршрутизаторы, серверы доступа к сети (network access server), каналы T1, коммутаторы, мосты, концентраторы, серверы аутентификации (этот

список можно продолжать и далее). Лучшей защитой от этого недостатка является надлежащее техническое обслуживание, регулярное резервное копирование, обеспечение избыточности и отказоустойчивости.

Между маршрутизаторами должно быть организовано несколько маршрутов, должны применяться динамические протоколы маршрутизации. При этом в случае отказа одного из маршрутизаторов, все оставшиеся маршрутизаторы будут проинформированы об изменении маршрута. Для соединений WAN, должна быть настроена функция "преодоления отказа" (failover), чтобы сохранить доступность ISDN даже в случае отказа WAN-маршрутизатора. Рисунок 10-1 иллюстрирует типовую среду для организации электронной коммерции, которая содержит избыточные устройства.

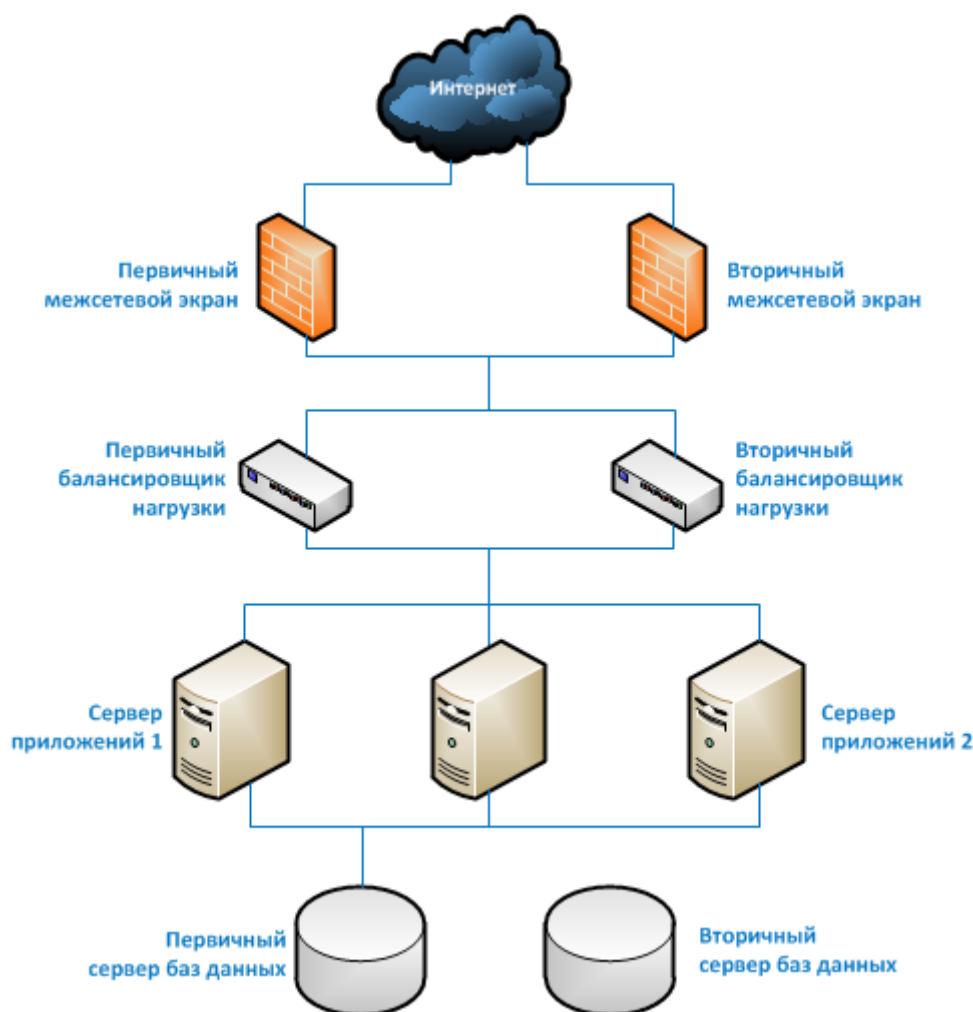


Рисунок 10-1. Каждое критичное устройство должно иметь избыточное дублирующее устройство для обеспечения гарантий доступности

RAID-массив (redundant array of independent disks - избыточный массив независимых дисков) обеспечивает отказоустойчивость дискового хранилища и может улучшить производительность системы. Избыточность и скорость обеспечивается разделением данных и их записью на несколько дисков, что позволяет нескольким дискам работать одновременно, увеличивая скорость чтения/записи информации. В RAID-массивах осуществляется контроль целостности записанных на них данных, он называется контролем *четности* (parity). Если один диск выходит из строя (нарушается целостность информации на нем), другие диски могут продолжить совместную работу и восстановить потерянные данные.

Для информации, которая должна быть постоянно доступна (т.е. для которой MTTR должен быть практически нулевым) и для которой неприемлемо существенное уменьшение

производительности, должно быть организовано «зеркалирование» (RAID 1) или «дуплексирование». В обоих этих режимах операция записи выполняется одновременно (или почти одновременно) на несколько физических дисков. Различие между «зеркалированием» и «дуплексированием» состоит в том, что при «зеркалировании» два (или более) диска, на которые записываются данные, могут быть подключены к одному и тому же контроллеру, оставляя при этом в хранилище единую точку отказа, вызванную неисправностью самого контроллера; при «дуплексировании» используются два (или более) отдельных контроллеров. «Зеркалирование» и «дуплексирование» могут выполняться на нескольких устройствах хранения данных, физически удаленных друг от друга, что обеспечивает дополнительную степень отказоустойчивости.

Одно из преимуществ «зеркалирования» / «дуплексирования» заключается в том, что большинство операций чтения могут выполняться с помощью любой из имеющихся копий, потенциально позволяя увеличивать скорость чтения кратно количеству используемых в RAID-массиве устройств. RAID-массивы будут рассмотрены далее в этом разделе. Также, в этом разделе будут рассмотрены некоторые другие технологии, которые могут использоваться для предотвращения потери производительности или простоя систем, вызванных единой точкой отказа.

Устройства хранения с прямым доступом

Устройство хранения с прямым доступом (DASD – Direct Access Storage Device) – это общий термин для устройств хранения с магнитными дисками, которые исторически использовались в средах мэйнфреймов и миникомпьютеров. RAID-массив является одной из разновидностей DASD. Ключевое различие между устройствами с прямым доступом и последовательным доступом (SASD – Sequential Access Storage Device) заключается в том, что в устройстве с прямым доступом любой блок информации может быть быстро найден и прочитан, тогда как при последовательном доступе нужно будет пройти весь путь между текущей позицией и позицией, в который находится необходимый блок. Примером устройства хранения с последовательным доступом является стример. Однако некоторые ленточные накопители обладают минимальным количеством встроенных функций прямого доступа. Такие накопители работают с многодорожечными лентами и хранят в определенных областях на ленте и в кэше ленточного накопителя информацию о том, где начинаются основные разделы данных на ленте. Это позволяет ленточному накопителю быстрее находить начало дорожки и то место, с которого нужно начинать выполнение запрошенной операции. Это позволяет существенно сократить время перехода к нужным точкам и значительно повышает скорость работы таких ленточных накопителей. Но даже с учетом такого увеличения скорости, разница в производительности между устройствами последовательного и прямого доступа просто огромна.

RAID-массивы

Избыточный массив независимых дисков (RAID – Redundant array of independent disks) представляет собой технологию, используемую для обеспечения избыточности и/или повышения производительности. RAID-массив – это логический массив, который объединяет несколько физических дисков. При записи на RAID-массив информации, данные сохраняются на всех дисках, входящих в его состав. При этом для приложений и других устройств RAID-массив представляется в виде единого запоминающего устройства.

При записи данных на входящие в состав RAID-массива диски, используется техника *чередования* (striping). Эта техника позволяет разделить данные для их записи на несколько дисков. При этом скорость записи не снижается, а скорость чтения резко возрастает, т.к. данные загружаются одновременно с нескольких дисков.

Различные уровни RAID определяют схему хранения данных на физических дисках, входящих в состав RAID-массива. Некоторые уровни обеспечивают только повышение

производительности, тогда как другие – обеспечивают и производительность, и отказоустойчивость. Если RAID-массив обеспечивает отказоустойчивость, применяется четность. Если диск выходит из строя, именно четность дает основные указания, которые позволяют RAID-массиву восстановить потерянные данные на новый жесткий диск. Четность используется для перезаписи информации на новый диск, обеспечивая таким образом восстановление всей информации. Большинство RAID-систем допускают «горячую» замену (hot-swapping) дисков – это означает, что диски в такой системе можно заменять непосредственно в процессе ее работы. При замене диска или добавлении нового диска, данные четности используются для восстановления данных на новый, только что добавленный, диск.

ПРИМЕЧАНИЕ. RAID уровня 15 на самом деле является комбинацией уровней 1 и 5, а RAID 10 представляет собой комбинацию уровней 1 и 0.

Наиболее распространенными уровнями RAID, используемыми в наше время, являются уровни 1, 3 и 5. В Таблице 10-2 описаны все возможные уровни RAID.

Уровень RAID	Описание работы	Название
0	Дисковый массив из двух или более дисков без использования избыточности и четности. Информация разбивается на блоки данных, которые поочередно записываются на диски, входящие в состав массива. Если один диск выходит из строя, весь массив становится неработоспособным. Применяется только для повышения производительности.	Чередование (striping)
1	Зеркалирование дисков. Одни и те же блоки данных записываются на все диски, входящие в состав массива. Если один диск выходит из строя, массив продолжает работать, данные загружаются с других дисков массива.	Зеркалирование (mirroring)
2	Данные разбиваются на слова, причем размер слова соответствует количеству дисков для записи данных. Для каждого слова вычисляется код коррекции ошибок, который записывается на диски, выделенные для хранения контрольной информации. Их число равно количеству бит в слове контрольной суммы. Возможно использование в массиве до 39 дисков, при этом 32 из них будут использоваться для хранения данных, а 7 – для хранения контрольной информации. Эта схема хранения данных не используется в промышленных средах, поскольку она плохо справляется с большим количеством запросов.	Четность с использованием кода Хэмминга (Hamming code parity)
3	Данные разбиваются на блоки размером 1 байт и распределяются по всем дискам, входящим в состав массива, за исключением одного, который используется для хранения блоков четности. При выходе из строя одного диска, хранившиеся на нем данные могут быть восстановлены с помощью диска четности.	Четность на уровне байтов
4	RAID 4 аналогичен RAID 3, но данные в нем разбиваются на блоки, а не на байты.	Четность на уровне блоков
5	Блоки данных и контрольные суммы циклически записываются на все диски массива. Под контрольными суммами подразумевается результат операции XOR(исключающее или). Этот уровень обеспечивает отказоустойчивость, т.к. при его использовании нет единой точки отказа.	Четность с чередованием (interleave parity)
6	RAID 6 – похож на RAID 5, но имеет более высокую степень отказоустойчивости, т.к. под контрольные суммы в нем выделяется емкость двух дисков, а сами контрольные суммы рассчитываются по разным алгоритмам и записываются на все диски. Обеспечивает работоспособность даже после одновременного выхода из строя двух дисков (защита от кратного отказа).	Вторичные данные четности (или двойная четность)
10	Данные записываются одновременно с зеркалированием и чередованием. Архитектура этого уровня представляет собой массив типа RAID 0, сегментами которого вместо отдельных дисков являются массивы RAID 1. RAID 10 объединяет в себе высокую отказоустойчивость (в т.ч. защиту от кратного отказа) и производительность.	Чередование и зеркалирование

Таблица 10-2. Различные уровни RAID

ПРИМЕЧАНИЕ. Самым часто используемым уровнем RAID является уровень 5.

Ссылки по теме:

- The RAID Tutorial from the University of Massachusetts
- “A Case for Redundant Arrays of Inexpensive Disks (RAID),” by David A. Patterson, Garth Gibson, and Randy H. Katz

Массив с неактивными дисками

Относительной новинкой, выходящей на арену хранилищ среднего уровня (в сотни терабайт), является **массив с неактивными дисками** (MAID – Massive Array of Inactive Disks). MAID применяется в нише, в которой требуются хранилища данных, объемом до нескольких сотен терабайт, выполняющих в основном операции записи. Меньшие требования к хранилищу, как правило, не оправдывают повышенную стоимость и более сложную эксплуатацию MAID. А средние и большие хранилища, в которых постоянно используется значительная часть данных, не позволят получить реальных выгод от MAID, поскольку производительность MAID при таком использовании быстро снижается по мере увеличения потребности в активных накопителях выше уровня, который может предложить MAID. При максимально высоких требованиях к хранилищу, используемому в основном для записи, самым экономичным решением остаются стримеры, благодаря минимальной стоимости единицы объема хранящейся на ленте информации, а также снижению процента носителей информации (от общего их количества), которые в данный момент должны быть в режиме онлайн.

В дисковых массивах MAID отключается питание всех неактивных дисков, работает только дисковый контроллер. Когда приложение запрашивает данные, контроллер включает соответствующий диск (диски), передает данные, а затем отключает диск (диски). Если диски используются редко, потребление энергии значительно сокращается, а срок службы дисков может возрасти.

Избыточный массив независимых лент

Избыточный массив независимых лент (RAIT – redundant array of independent tapes) похож на RAID, но в нем используются ленточные накопители вместо жестких дисков. Ленточное хранилище – это самый дешевый вариант для очень больших объемов данных, но очень медленный по сравнению с дисковым хранилищем. RAIT может быть подходящим решением для очень больших хранилищ, ориентированных на запись информации, для которых MAID оказывается не экономичен, и где желательна более высокая производительность, чем для обычных ленточных хранилищ, либо требуется более высокая надежность, чем может обеспечить ленточное хранилище.

Как и при использовании RAID 1, в RAIT данные параллельно записываются на несколько ленточных накопителей, с использованием или без использования избыточной ленты четности. Это обеспечивает высокую емкость при низкой стоимости, типичной для ленточных хранилищ, с более высокой скоростью передачи данных, чем для обычной ленты. Кроме того, RAIT может обеспечить целостность данных (опционально).

Сети хранения данных

Сеть хранения данных (SAN – Storage Area Network) состоит из большого количества устройств хранения данных, связанных между собой высокоскоростной внутренней сетью и специальными коммутаторами, ориентированными на хранилища. Это создает структуру (fabric), которая позволяет пользователям подключиться и взаимодействовать в прозрачном режиме. Чтобы сделать запрос к файлу, пользователю не нужно знать, на каком сервере или ленточном накопителе он находится – программное обеспечение SAN найдет нужный файл и предоставит его пользователю.

Во многих инфраструктурах все данные «разбросаны» по сети и нахождение необходимой информации может оказаться сложной задачей. К тому же могут возникнуть сложности при настройке системы резервного копирования для организации копирования всех необходимых данных.

SAN обеспечивает избыточность, отказоустойчивость, надежность, резервирование и позволяет пользователям и администраторам взаимодействовать с SAN как с одной виртуальной сущностью. Поскольку сеть SAN (по которой передаются данные внутри SAN) отделена от обычных сетей передачи данных компании, ее производительность, надежность

и гибкость не подвержена воздействию от других систем в основной сети компании.

В средних или небольших компаниях SAN обычно не используются. SAN предназначены для компаний, которые должны обрабатывать терабайты данных, и имеют достаточно денег для покупки такой техники. Поставщики хранилищ в настоящее время переживают период расцвета не только потому, что компании делают свой бизнес цифровым и должны где-то хранить данные, но также и потому, что требования законодательства обязывают компании хранить некоторые данные в течение определенного срока (в большинстве случаев, измеряемого годами). Представьте себе, что потребуется для хранения всего почтового трафика вашей компании в течение семи лет... А это только один из видов данных, которые должны быть сохранены.

ПРИМЕЧАНИЕ. Ленточные накопители, оптические запоминающие устройства и дисковые массивы могут быть подключены к SAN и работать совместно с ней.

Кластеризация

Кластеризация (clustering) – это отказоустойчивая серверная технология, которая похожа на использование избыточных серверов, за исключением того, что каждый сервер, входящий в кластер, принимает участие в обработке поступивших запросов. Кластер серверов (server cluster) – это группа серверов, которые выглядят для пользователя как один логический сервер и могут управляться как единая логическая система. Кластеризация повышает доступность и масштабируемость. Она группирует различающиеся по своим физическим характеристикам системы, что повышает устойчивость к сбоям и улучшает производительность. Кластеры используют интеллектуальные модули для балансировки трафика. Пользователи, использующие кластер, даже не догадываются, что в разные моменты времени их запросы могут выполнять разные системы. Для пользователей все серверы, входящие в состав кластера, выглядят одним единым сервером. Кластеры могут также называться фермами серверов (server farm).

Если одна из систем в кластере выходит из строя, работа кластера продолжается, т.к. остальные системы просто берут на себя возросшую нагрузку, хотя это может привести к снижению производительности кластера. Использование кластеров более привлекательно, чем наличие второго (резервного) сервера, который простаивает, ожидая своего часа – выхода из строя основного сервера. Наличие простаивающего длительное время резервного сервера может являться слишком расточительным для компании. Решением этой проблемы будет использование кластеризации, при которой все системы используются для обработки запросов и ни одна из них не простаивает, ожидая, что что-то сломается. Кластеризация является логическим продолжением избыточных серверов.

Кластеризация дает гораздо больше, чем просто повышение доступности. Она также обеспечивает балансировку нагрузки (каждая система получает только часть от общего числа запросов, полученных кластером), избыточность и отказоустойчивость (кластер продолжает работать, если одна из его систем выходит из строя).

Grid-вычисления

Grid-вычисления (grid computing) являются другим вариантом распределенных вычислений с балансировкой нагрузки. Эта технология похожа на кластеры, но она реализуется слабо связанными между собой системами, которые могут произвольным образом присоединяться и покидать распределенную систему (grid). Большинство компьютеров не используют все свои вычислительные ресурсы полностью, они имеют значительные запасы процессорных ресурсов, которые большую часть времени в течение дня не используются. Это достаточно расточительно, поэтому умные люди придумали способ использования всех этих дополнительных вычислительных мощностей. Так же, как энергосистема обеспечивает потребителей электроэнергией по мере необходимости (если, конечно, вы не забываете оплачивать счета), компьютеры могут добровольно предоставлять свои дополнительные

вычислительные ресурсы различным группам для различных проектов. Первым проектом, использующим grid-вычисления, был проект SETI (поиск внеземного разума), в рамках которого обычным пользователям сети Интернет предлагалось установить на свой компьютер программу, которая задействует свободные ресурсы этого компьютера для участия в сканировании Вселенной в поисках инопланетного разума, пытающегося говорить с нами.

Хотя это может быть похоже на работу кластера, центральный контроллер которого управляет распределением ресурсов и пользователей по узлам кластера, а также управляет самими узлами в кластере (в том же доверенном домене), при grid-вычислениях узлы не доверяют друг другу и не имеют централизованного управления.

Приложения, которые имеют техническую возможность работы с использованием grid-вычислений, могут воспользоваться экономическими преимуществами этой технологии – большой и дешевой вычислительной мощностью распределенной системы. Однако не следует использовать распределенные вычисления для обработки секретной информации, поскольку данные, загруженные на каждый из участвующих в вычислениях компьютер, не могут быть гарантировано защищены от владельца этого компьютера. Кроме того, поскольку различные члены распределенной системы имеют различный объем доступных ресурсов и не доверяют друг другу, grid-вычисления не подходят для приложений, которым требуется постоянное взаимодействие и скоординированное планирование рабочей нагрузки между отдельными модулями. Иными словами, критичные данные не должны обрабатываться с помощью систем grid-вычислений, кроме того, эта технология не подходит для выполнения приложений, чувствительных ко времени.

Наиболее подходящим вариантом для использования grid-вычислений являются такие проекты, как финансовое моделирование, моделирование погоды и землетрясений. Каждая из этих задач моделирования имеет невероятное количество входных данных и переменных, которые должны непрерывно обрабатываться. Этот подход также используется для попыток взлома различных алгоритмов, для генерации «радужных» таблиц (Rainbow Table).

ПРИМЕЧАНИЕ. «Радужные» таблицы содержат хэши всех возможных паролей. Это позволяет злоумышленникам взламывать пароли гораздо быстрее, чем при выполнении атаки по словарю или полного перебора возможных паролей.

8.4. Резервное копирование

Программное обеспечение резервного копирования и резервные аппаратные устройства являются двумя основными компонентами обеспечения доступности сети (эти вопросы подробно рассматривались в Доменах 04 и 07, поэтому здесь мы обсудим их очень кратко). Вы должны иметь возможность восстановить данные, если жесткий диск выйдет из строя, произойдет авария или чрезвычайная ситуация, либо будут повреждены отдельные системы.

Должна быть разработана политика, которая определяет, что подлежит резервированию, как часто и каким образом должно выполняться резервное копирование данных. Если важная информация хранится на рабочих станциях пользователей, Департамент эксплуатации должен разработать методику резервного копирования, которая будет включать в резервные копии содержимое определенных каталогов на рабочих станциях пользователей. Другим вариантом является разработка в компании политики, которая требует от пользователей хранить критичные данные только в предназначенных для этих целей сетевых папках, которые включаются в резервные копии. Резервное копирование может выполняться один или два раза в неделю, каждый день или раз в три часа. Компания самостоятельно определяет для себя наиболее оптимальные параметры этой процедуры. Чем чаще выполняется резервное копирование, тем больше требуется ресурсов для его выполнения и хранения созданных копий, поэтому необходимо соблюдать баланс между расходами на выполнение резервного копирования и рисками потенциальной потери данных.

Компания может решить, что выполнение автоматического резервного копирования с использованием специализированного программного обеспечения является более экономичным и эффективным вариантом, по сравнению с затратами времени специалистов ИТ на выполнение этой задачи. При этом должно контролироваться содержимое автоматически создаваемых резервных копий, чтобы убедиться, что копирование нужной информации прошло успешно. Это гораздо лучше, чем после пожара в серверной выяснить, что система автоматического резервного копирования была настроена на копирование только временных файлов.

Иерархическое управление носителями

Иерархическое управление носителями (HSM - Hierarchical Storage Management) обеспечивает непрерывное выполнение резервного копирования в режиме реального времени. Эта технология сочетает в себе использование жестких дисков совместно с более дешевыми и медленными оптическими и ленточными накопителями. Система HSM динамически управляет хранением и восстановлением файлов, которые копируются на различающиеся по скорости и стоимости носители информации. Часто используемые данные хранятся на более быстрых носителях информации, а редко используемые – на медленных (в т.ч. последовательных (near-line)) устройствах, как показано на Рисунке 10-2. Хранилище может включать в себя такие носители информации, как оптические диски, магнитные диски и ленты. Вся эта функциональность, включая выбор носителя информации, работает в фоновом режиме без необходимости участия пользователя.

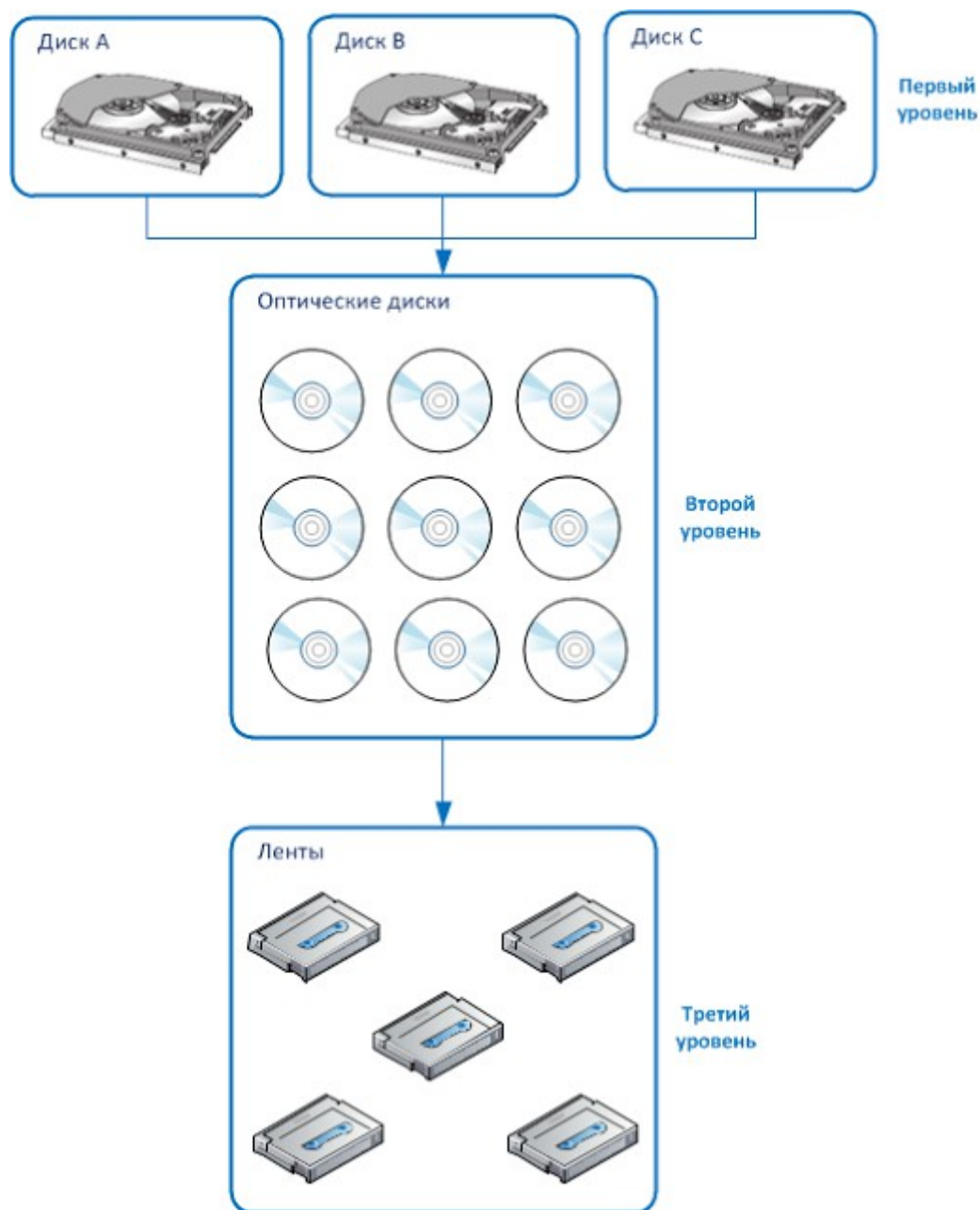


Рисунок 10-2. HSM обеспечивает экономичный и эффективный способ хранения данных

HSM работает в соответствии с настройками, основываясь на компромиссе между затратами на хранение и доступностью информации. Она переносит содержимое редко используемых файлов на более медленные и более дешевые устройства хранения, оставляя «заглушку», которая выглядит для пользователя как обычный файл, содержащий полные данные перенесенного файла. Когда пользователь или приложение обращается к заглушке, HSM использует записанную в ней информацию, чтобы найти реальное местоположение нужных данных и извлечь их прозрачно для пользователя.

Эта технология была создана для экономии денег и времени. Хранить все данные на жестком диске может быть слишком дорого. Если большую часть данных хранить на лентах, слишком много времени будет занимать их восстановление в случае необходимости. Поэтому HSM обеспечивает сбалансированный подход, оперативно предоставляя пользователю нужные ему данные. При этом администратору не нужно искать ленты или оптические диски, на которых они сохранены.

Технологии полного, инкрементального и дифференциального резервного копирования данных были рассмотрены в Домене 07, там же было рассмотрено необходимое для этих

целей программное и аппаратное обеспечение. В резервные копии должны включаться операционные системы и приложения, а также конфигурационные файлы. В большинстве компаний все системы подключены к сети, а в сетевых устройствах также могут происходить сбои и потери данных. Для сетевого устройства потеря данных обычно означает утрату конфигурационных файлов (что приведет к тому, что сетевое устройство не сможет даже загрузиться), либо замену на конфигурацию «по умолчанию» (что позволит сетевому устройству загрузиться, но выполнять свои функции оно все равно не сможет). Поэтому резервное копирование конфигурационных файлов сетевых и других устройств (например, телефонных систем), используемых в среде, также необходимо.

ПРИМЕЧАНИЕ. Для сохранения конфигурационных настроек сетевых устройств часто используются серверы TFTP. Однако TFTP – небезопасный протокол, а некоторые сетевые настройки содержат критичную информацию, для которой должна быть обеспечена конфиденциальность. Кроме того, возможны скоординированные атаки на сетевые устройства, загружающие свои конфигурационные файлы с сервера TFTP. Для этого атакующий вызывает сбой сетевого устройства, предварительно взломав сервер TFTP и внося изменения в конфигурационные файлы. При этом сетевое устройство перезагрузится и получит вредоносные настройки. Поэтому следует искать альтернативы TFTP.

8.5. Планирование действий на случай непредвиденных ситуаций

Когда случается инцидент, требуется больше, чем просто знать, как восстановить данные из резервной копии. Необходимы подробные процедуры, описывающие действия по сохранению доступности критичных систем и гарантирующие продолжение их функционирования и обработки данных. Кризисное управление (contingency management) определяет, что нужно делать во время и после инцидента. Должны быть документированы и доступны всему персоналу Департамента эксплуатации действия, которые необходимо предпринять для реагирования на чрезвычайную ситуацию, поддержки непрерывности выполнения операций, учитывающие возможность возникновения крупных аварий. Такие документы должны храниться, как минимум, в трех местах: оригинал – на основной площадке, копия – также на основной площадке, но в защищенном несгораемом сейфе, а еще одна копия – на территориально удаленной площадке.

Планам действий в непредвиденных ситуациях (contingency plans) не следует доверять, пока они не были проверены. Компания должна проводить учения для того, чтобы сотрудники в полной мере осознали свои обязанности и поняли, как их выполнять. Кроме того, нужно решить вопрос о порядке поддержания этих планов в актуальном состоянии. По мере внесения изменений в сетевую среду компании, должны актуализироваться и планы по ее спасению в случае аварии.

Хотя в безопасности часто считаются синонимами термины «*планирование действий в непредвиденных ситуациях*» (contingency planning) и «*планирование непрерывности бизнеса*» (business continuity planning), очень важно понимать реальную разницу между ними. ВСП рассматривает вопрос – как сохранить компанию в бизнесе после катастрофы. Здесь речь идет о выживании компании и обеспечении возможности продолжения выполнения критически важных функций даже после катастрофы. Планы действий в непредвиденных ситуациях рассматривают вопросы по борьбе с менее значительными инцидентами, которые не квалифицируются как катастрофы (например, отключение электроэнергии, сбой сервера, отключение связи с сетью Интернет, сбой программного обеспечения и т.п.). Важно, чтобы компания была готова как к крупным, так и к менее значительным проблемами, которые могут произойти рано или поздно.

9. Мейнфреймы

Основная часть рассмотренного ранее материала была посвящена системам низкого (настольные компьютеры, ноутбуки, рабочие станции) и среднего (серверы) уровня. Однако **мейнфреймы** (mainframe) по-прежнему используются, и, по всей видимости, будут

использоваться еще некоторое время. Различия между мейнфреймом и мощным сервером, подключенным к SAN, сокращаются, но определенные атрибуты пока продолжают оставаться отличием мейнфрейма.

Мейнфреймы обеспечивают высокую надежность и доступность, причем реализовано это не за счет их аппаратной архитектуры, а за счет очень консервативных (и, следовательно, очень дорогих) инженерных решений. Разработчики мейнфреймов тратят огромные деньги на проведение исследований и сложной разработки, чтобы сделать системы нижнего уровня максимально быстрыми и с максимальным количеством функций обеспечения надежности, что всегда является недостатком низкоуровневых систем. Побочным эффектом этого являются высокие инвестиции в качество программного обеспечения (в т.ч. в операционную систему и приложения), чтобы данные, полученные в результате обработки в мейнфрейме, имели тенденцию к большей точности, чем при обработке обычным программным обеспечением на стандартных серверах.

Учитывая повышенную надежность, мейнфреймы лучше подходят для обработки критически важных данных, которые должны быть всегда доступны.

Существует ключевое различие в аппаратном обеспечении между мейнфреймами и даже самыми мощными из систем среднего уровня: мейнфрейм аппаратно спроектирован, в первую очередь, для массового ввода/вывода. Мощность процессоров увеличилась в миллион раз за последние 25 лет, но возможности ввода/вывода улучшились на порядки меньше. Это позволяет мейнфреймам, особенно сегодня, когда они получили огромные преимущества за счет современных процессоров, одновременно запускать огромное количество процессов, без их простоя в ожидании завершения операций ввода/вывода. Даже самые мощные современные компьютеры могут одновременно выполнять только небольшую часть требовательных к данным процессов, чтобы «бутылочное горлышко» в виде операций ввода/вывода не приводило к простоям компьютера. Это делает мейнфреймы не-процессо-специфичными (какими могут быть сверхбыстрые дешевые современные компьютеры), поэтому они являются отличными платформами, обрабатывающими большие объемы данных. Мейнфреймы имеют огромную процессорную мощность, скрытую в их процессорах для фронтальных интерфейсов (которые поддерживают взаимодействие с пользователями, не отвлекая на это центральный процессор), процессорах ввода/вывода (которые перемещают данные и работают с дисками и ленточными накопителями, не загружая центральный процессор), а также сетевых процессорах (которые эффективно перемещают данные из сети и в сеть, также без загрузки центрального процессора). Вся эта «скрытая» вычислительная мощь требует использования соответствующего оборудования, что обуславливает высокую стоимость мейнфреймов.

Еще одним преимуществом в обеспечении надежности мейнфреймов является то, что они не требуют большого объема работ для своей поддержки. По сравнению с регулярным выпуском патчей, которые нужно ставить на большое количество систем нижнего уровня, регулярные патчи для мейнфреймов выпускаются значительно реже, а количество исправлений в каждом выпуске намного меньше.

Другим классическим отличием между мейнфреймом и системой среднего уровня или персональным компьютером является пользовательский интерфейс. В наше время мейнфреймы чаще выполняют пакетную обработку, а не работу в интерактивном режиме. В отдельных случаях (хотя эта практика сокращается) они принимают запросы от пользователей (через терминалы мейнфрейма) в виде заданий через Remote Job Entry (RJE).

Более принципиальным различием между мейнфреймами и другими типами систем, предлагающим интересные возможности, является то, что начальная загрузка мейнфрейма (IPL) может настроена на различные системы при каждой загрузке. Это позволяет обеспечить обратную совместимость новых процессоров со старыми операционными системами и дает компаниям возможность сохранить в течение многих лет эффективность

произведенных в программное обеспечение инвестиций. Для обычного компьютера или системы среднего уровня очень старое программное обеспечение не может быть использовано. Мейнфреймы первыми начали широкомасштабно использовать виртуализацию, позволившую одному физическому мейнфрейму (который может состоять из нескольких банков памяти, накопителей информации и процессоров, которые в наше время могут даже добавляться динамически) представляться в виде нескольких независимых компьютеров, со строгим разделением среди своих нескольких операционных сред, совместным использованием общих ресурсов физического мейнфрейма, в соответствии с настройками, установленными системным администратором.

Суперкомпьютеры можно рассматривать как мейнфреймы особого класса. В их архитектуре много общего. Но если мейнфреймы предназначены для выполнения очень больших объемов общей обработки, суперкомпьютеры оптимизированы для выполнения чрезвычайно сложной централизованной обработки (что также требует огромных возможностей ввода/вывода, реализованных в архитектуре мейнфрейма). Несколько процессоров мейнфреймов распределяют между собой нагрузку от выполнения очень большого числа обычных процессов. Суперкомпьютеры же выполняют большое число очень высоко распараллеленных копий конкретного приложения, работающего в режиме реального времени, либо очень небольшое число чрезвычайно сложных научных алгоритмов использующих огромные объемы данных одновременно.

10. Безопасность электронной почты

Интернет был изначально разработан в основном для взаимодействия и совместного использования информации в правительственных учреждениях и университетах, но сегодня он необходим коммерческим компаниям для повышения своей производительности и получения прибыли. Миллионы людей также зависят от этого окна в мир, поскольку оно предоставляет быстрое и эффективное средство для взаимодействия.

Электронная почта стала важной и неотъемлемой частью жизни людей. Она используется для взаимодействия с семьей и друзьями, партнерами по бизнесу и клиентами, коллегами и руководством, для организации интернет-магазинов и работы правительственных учреждений. При обычном повседневном использовании электронной почты, как правило, не рассматриваются вопросы аутентификации, безопасности и целостности передаваемых сообщений. Большинство пользователей осведомлены о вредоносных программах, которые могут вкладываться в сообщения электронной почты, а также о том, что сообщения электронной почты могут быть изменены в процессе передачи.

Сообщения электронной почты могут быть легко подделаны (*спуфинг*), например, с целью изменения поля, содержащего сведения об отправителе. Все что нужно для этого злоумышленнику, это изменить стандартные настройки своего почтового клиента и перезапустить приложение. Например, злоумышленник может изменить имя в поле отправителя сообщения на имя сетевого администратора и направить сообщение секретарю генерального директора, с указанием сменить свой пароль на «12345678», мотивируя это проблемами с серверами. Открыв это сообщение, секретарь увидит, что оно пришло от администратора, и, вероятно, выполнит просьбу, ничего не заподозрив.

Такие действия очень широко используются в наше время, это одна из популярных тактик социальной инженерии. Другой вариант называется фишингом. Фишинг – это отправка поддельного сообщения, которое выглядит очень похоже на сообщения, отправляемые пользователю доверенным отправителем, с которым у пользователя есть какие-либо отношения (например, банком). В сообщении пользователю говорят о необходимости перейти по ссылке, содержащейся в сообщении, которая якобы ведет на страницу регистрации. Злоумышленник может указать в сообщении, что это необходимо, например, для выполнения действий по обслуживанию учетной записи пользователя. Однако вместо этого, ссылка направляет пользователя на поддельный веб-сайт, который очень похож на

сайт его банка. Этот поддельный сайт принадлежит злоумышленнику. Когда пользователь введет на нем свои учетные данные, злоумышленник получит все необходимые реквизиты для доступа к счету пользователя на реальном сайте банка.

Для защиты от такой (и аналогичных) атаки необходимо обеспечить надлежащую аутентификацию отправителя, которая позволит убедиться, что сообщение на самом деле пришло из указанного в нем отправителя. Компаниям, которые рассматривают безопасность в качестве одной из своих приоритетных задач, следует внедрить систему защиты электронной почты на основе цифровой подписи сообщений (например, PGP – Pretty Good Privacy) или использовать инфраструктуру открытых ключей (PKI). Дополнительно можно рассмотреть возможность использования шифрующего протокола, что поможет защититься от несанкционированного перехвата сетевого трафика и передаваемых с его помощью сообщений.

Сообщение электронной почты, которое передается из Москвы во Владивосток проходит через множество сетевых устройств, установленных между отправителем и получателем. При этом существует несколько потенциальных точек перехвата трафика, в которых злоумышленник может перехватывать, просматривать, изменять или удалять сообщения в процессе их передачи, как показано на Рисунке 10-3.

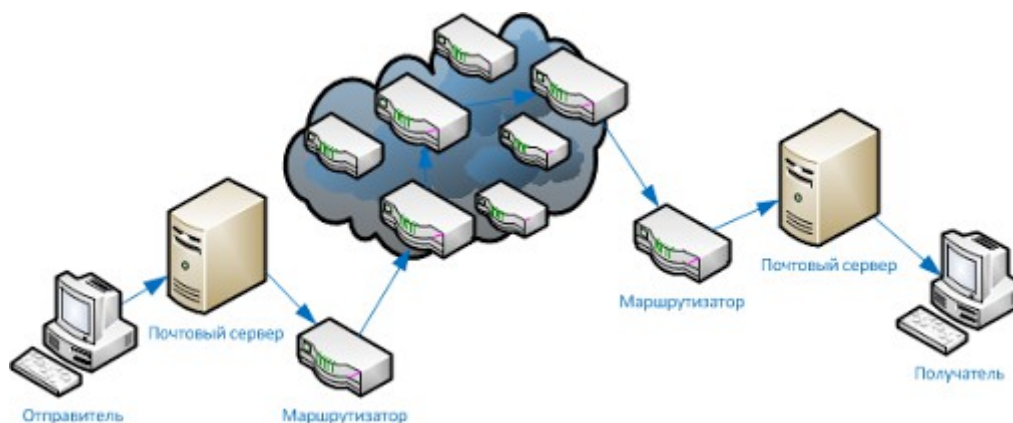


Рисунок 10-3. Сообщение электронной почты может быть перехвачено на множестве различных точек

Если пользователь принимает решение воспользоваться определенным механизмом безопасности, чтобы защитить свои сообщения от перехвата, модификации и подделки, он должен договориться с получателем своих сообщений об использовании одинакового механизма безопасности (например, средства шифрования и электронной подписи). Если для защиты будет применяться криптография с открытыми ключами, пользователи должны иметь возможность для безопасного обмена ключами шифрования. Это относится к использованию PGP, цифровых подписей, продуктов, использующих стандарт S/MIME. Эти вопросы подробно рассматривались в Домене 06. Если администратор или специалист по безопасности, хочет быть уверен, что все сообщения, передаваемые между двумя точками, шифруются, и не хочет зависеть в этом вопросе от действий самих пользователей (которые могут что-то забыть или сделать неправильно), он может воспользоваться решениями на основе VPN (это также обсуждалось в Домене 06).

10.1. Как работает электронная почта

Пользователь использует клиентскую программу электронной почты (почтовый клиент), которая позволяет создавать, изменять, отправлять, принимать и пересылать сообщения. Почтовый клиент может предоставлять и другие функции, такие как личная адресная книга, возможность добавления вложений в сообщения, установку флагов, отзыв сообщений, а также сохранение сообщений в различных папках.

Сообщения электронной почты не имеют смысла, если их нельзя отправить. Для отправки сообщений используется протокол **SMTP** (Simple Mail Transfer Protocol - Простой протокол передачи почты), который работает в качестве агента передачи сообщений - он передает сообщения с компьютера отправителя на почтовый сервер при нажатии пользователем кнопки «Отправить». SMTP также определяет стандарт обмена сообщениями, именно он определяет привычную большинству людей форму адреса электронной почты: something@somewhere.com.

Зачастую сообщению приходится путешествовать по сети Интернет, проходя через различные почтовые сервера, пока оно не поступит на почтовый сервер получателя. SMTP – это протокол, который передает это сообщение, он работает поверх протокола TCP. TCP используется в качестве транспортного протокола, поскольку он является надежным протоколом, обеспечивающим доставку почты до места назначения.

Чтобы использовать протокол SMTP, почтовый клиент пользователя должен быть совместим с ним и правильно настроен. Почтовый клиент предоставляет пользователю интерфейс для создания и редактирования сообщений, которые затем передаются посредством SMTP. Если привести аналогию с отправкой бумажного письма через почтовое отделение, почтовый клиент является пишущей машинкой, на которой человек пишет письмо; SMTP является почтовым курьером, который забирает почту из почтового ящика и доставляет ее в почтовое отделение; а почтовым отделением является сервер электронной почты. Почтовый сервер должен понимать, куда направляется сообщение, чтобы правильно маршрутизировать его.

Почтовый сервер часто называют сервером SMTP. Наиболее распространенным программным обеспечением SMTP-сервера является Unix-программа Sendmail, фактически являющаяся приложением почтового сервера. Unix использует Sendmail для хранения, сопровождения и маршрутизации сообщений электронной почты. В среде Microsoft обычно для этих целей используется Microsoft Exchange, а в среде Novell – программное обеспечение GroupWise. SMTP работает в тесном сотрудничестве с двумя протоколами почтового сервера: POP и IMAP, которые будут рассмотрены далее.

POP

POP (Post Office Protocol - Протокол почтового отделения) – это протокол интернет-сервера электронной почты, поддерживающий входящие и исходящие сообщения. Почтовый сервер использует POP для хранения и передачи сообщений электронной почты, а SMTP – для передачи сообщений между почтовыми серверами.

Небольшая компания может иметь один POP-сервер, который поддерживает все почтовые ящики всех сотрудников. Более крупные компании могут использовать несколько POP-серверов, например, по одному для каждого департамента компании. В сети Интернет также используются POP-серверы, которые позволяют обмениваться сообщениями людям со всего мира. Это достаточно полезная система, которая хранит сообщения на почтовом сервере, пока пользователь не загрузит свои сообщения на свой компьютер, вместо того, чтобы просто пытаться отправить сообщение непосредственно на компьютер получателя, который может быть в этот момент недоступен или отключен от сети.

Сервер электронной почты может использовать различные схемы аутентификации для предоставления пользователю доступа к его почтовому ящику. Обычно это реализуется с помощью имен и паролей пользователей.

IMAP

IMAP (Internet Message Access Protocol - Протокол доступа к электронной почте Интернета) является еще одним интернет-протоколом, который позволяет пользователям получать доступ к почте на почтовом сервере. IMAP, помимо реализации функций, предоставляемых протоколом POP, дает дополнительные возможности и функции. При использовании POP,

когда пользователь обращается к почтовому серверу, чтобы узнать, не пришли ли ему новые сообщения, все полученные сообщения автоматически загружаются на его компьютер, после чего они обычно удаляются с сервера (в зависимости от настроек). POP может быть неудобен мобильным пользователям, поскольку сообщения автоматически передаются на пользовательский компьютер (или устройство), а они могут не иметь достаточного пространства для хранения всех сообщений. В особенности это относится к мобильным устройствам, которые могут быть использованы для доступа к электронной почте. Также это может быть неудобно для людей, проверяющих свою почту на чужих компьютерах, поскольку их почта будет загружена на эти компьютеры.

Если пользователь использует IMAP вместо POP, он может скачать все сообщения или оставить их на почтовом сервере в своей папке сообщений, именуемой почтовым ящиком (mailbox). Пользователь может управлять сообщениями в своем почтовом ящике на почтовом сервере также, как если бы эти сообщения находились на его локальном компьютере: он может создавать и удалять сообщения, осуществлять поиск сообщений, устанавливать и отключать различные флаги. Это дает пользователю больше свободы и позволяет хранить сообщения в централизованном хранилище, пока он специально не включит функцию загрузки всех своих сообщений с почтового сервера на свой компьютер.

IMAP представляет собой серверный протокол хранения и пересылки почты, который считается преемником POP. IMAP предоставляет больше возможностей и администраторам в части управления и хранения сообщений пользователей.

Ретрансляция сообщений электронной почты

Электронная почта кардинально изменилась со времен мейнфреймов. Тогда для работы системы электронной почты использовались простые протоколы SNA (Systems Network Architecture - Системная сетевая архитектура) и формат ASCII. Сегодня множество различных видов почтовых систем работают в различных операционных системах и предлагают широкий спектр функциональных возможностей. Иногда компаниям требуется использовать в одной сети различные типы почтовых серверов и сервисов, что может быть достаточно сложной задачей, в т.ч. с точки зрения обеспечения безопасности.

У большинства компаний есть внешние почтовые серверы в DMZ, а также один или несколько почтовых серверов в локальной сети. Внешние почтовые серверы имеют прямое подключение к сети Интернет, поэтому они должны размещаться в защищенном пространстве DMZ. Эти серверы должны быть хорошо укреплены, а ретрансляционные механизмы на них должны быть правильно настроены. Почтовые серверы используют **агентов ретрансляции** (relay agent) для пересылки сообщения с одного почтового сервера на другой. Этот агент должен быть правильно настроен, чтобы почтовый сервер компании не использовался посторонними для рассылки спама.

Обычно рассылка спама является незаконной, поэтому отправляющие его люди хотят сохранить свою анонимность. Они ищут почтовые серверы в сети Интернет или в DMZ компаний, на которых механизмы ретрансляции позволяют получить к ним доступ, и используют эти серверы для рассылки спама. Если агент ретрансляции на почтовом сервере открыт, такой сервер может быть использован для приема любых почтовых сообщений и пересылки их нужным получателям. Таким образом, если компания неправильно настроит ретрансляцию почтовых сообщений, ее сервер может быть использован для распространения спама и другой незаконной корреспонденции. Важно, чтобы на почтовых серверах были установлены и настроены средства защиты от спама, которые обеспечат защиту от ретрансляции нежелательной почты. Почтовый сервер компании должен принимать только почту, адресованную домену компании, и не должен пересылать сообщения, предназначенные для других почтовых серверов и доменов.

Многие компании устанавливают на свои почтовые сервера антивирусные средства и

средства контентной фильтрации, пытаясь предотвратить распространение вредоносного кода и нежелательных сообщений. Важно фильтровать не только входящие, но и исходящие сообщения. Это позволит обеспечить защиту от распространения вирусов и спама сотрудниками компании и не позволит им отправлять электронные письма, нарушающие политику компании.

Ссылки по теме:

- “How E-Mail Works,” by Marshall Brain, HowStuffWorks
- IETF S/MIME Mail Security (smime) Charter

10.2. Безопасность факсов

Отправка факсов является сегодня весьма популярным способом передачи информации. Поэтому, как и другие каналы передачи информации, факсимильная связь должна учитываться в политике и программе безопасности компании.

Факсимильные аппараты могут привести к проблемам с безопасностью, если они используются для передачи конфиденциальной или критичной для компании информации. Передаваемая информация сканируется устройством и передается по телефонной линии получателю. Факсимильный аппарат получателя принимает информацию и распечатывает ее. Поскольку многие современные факсимильные аппараты могут работать в автоматическом режиме, полученный факс часто просто лежит в лотке, пока тот, кому он был адресован, не подойдет и не заберет его. Если переданный факс содержит конфиденциальную информацию, такое положение работы может быть не лучшей идеей, поскольку любой проходящий мимо может увидеть его.

Некоторые компании используют **факс-серверы** (fax server), которые являются системами, управляющими входящими и исходящими факсимильными сообщениями. При получении факса факс-сервером, он направляет его тому человеку, которому он был адресован. Передача документа при этом осуществляется, как правило, по внутренней электронной почте компании в электронном, а не печатном, виде.

Факс-сервер позволяет множеству пользователей передавать документы по факсу непосредственно со своих компьютеров, без необходимости «прогона» документов через факсимильный аппарат. Это сокращает количество конфиденциальных бумажных документов, которые затем должны быть надежно уничтожены, и может сэкономить деньги на печать и термобумагу, необходимую для обычных факсимильных аппаратов.

Факс-сервер обычно имеет возможности для распечатки получаемых факсов, но это может создать те же проблемы безопасности, что и автономный факсимильный аппарат. В средах, требующих высокого уровня безопасности, возможность печати полученных факсов должна быть отключена, важные документы должны сохраняться и право доступа к ним должно предоставляться только их получателю (причем только в электронном, а не печатном виде).

При использовании факс-серверов обычно доступны широкие возможности журналирования и аудита событий, их следует использовать и осуществлять надлежащий контроль, особенно в компаниях, которые требуют высокого уровня безопасности. Поскольку данные факсов передаются факсимильными аппаратами в открытом виде, некоторые компании могут дополнительно использовать средства для шифрования передаваемых данных. Компания может использовать **факс-шифратор** (fax encryptor) – механизм канального шифрования, который шифрует все данные факса, которые попадают в сетевой кабель или телефонный провод. Компании имеет смысл внедрить факс-шифратор, если она не хочет зависеть от сознательности каждого отдельного пользователя. Используя факс-шифратор, компания может быть уверена, что все данные, выходящие за пределы ее сети, надлежащим образом зашифрованы.

10.3. Методы взлома и атак

Далее будут рассмотрены несколько типов атак. Будет показано, как они связаны между собой, как они могут быть обнаружены и как от них можно защититься.

Большинство инструментов, применяемых хакерами, имеют двойное назначение – они могут быть использованы для добра или зла. Злоумышленник может использовать специальный инструмент, чтобы найти эксплуатируемую уязвимость и воспользоваться ей, а специалист по безопасности может использовать тот же инструмент для поиска и устранения уязвимостей. Когда такой инструмент используется «черными шляпами» (злоумышленники), это называется *хакингом* (hacking). Когда такой инструмент используют «белые шляпы» (специалисты по безопасности), это называется *этичным хакингом* (ethical hacking) или *тестированием на проникновение* (penetration testing).

Инструменты, используемые для проведения атак на сети и системы, со временем стали настолько мощными и в то же время простыми, что даже человек с невысокой квалификацией (иногда таких людей называют *скрипт-кидди* (script kiddie)) способен выполнить чрезвычайно разрушительную атаку. Многие из этих инструментов сегодня имеют графический интерфейс, проводящий пользователя по шагам: идентификация ресурсов, создание карты сети, проведение атаки. Людям больше не нужно детально разбираться стеках протоколов, знать назначение отдельных полей в них, понимать процесс выполнения программ операционными системами и даже знать программирование. Хакеру достаточно просто указать диапазон IP-адресов в соответствующем окне программы и нажать кнопку «Старт». Современное сообщество хакеров использует множество различных инструментов (типов инструментов), некоторые из них способны автоматически генерировать код вирусов и червей, создавать эксплойты для конкретной уязвимости конкретной операционной системы, платформы или даже версии приложения.

Хакерские утилиты широко доступны, их без особого труда может получить любой желающий. В свое время, подобные инструменты были доступны лишь небольшой группе высококвалифицированных специалистов, а сегодня этому посвящены сотни веб-сайтов, рассказывающих людям, как использовать эксплойты, и предоставляющих средства выполнения атак за небольшую плату, либо вообще бесплатно. Простота использования этих средств, привела к значительному росту интереса к хакерству.

Любой администратор безопасности, работающий с межсетевым экраном, или простой пользователь, на компьютере которого работает программный межсетевой экран, знает, насколько активно в Интернете осуществляется сканирование портов. Межсетевые экраны постоянно получают подобные запросы. Некоторые сканеры ищут определенные типы компьютеров (например, системы Unix, веб-серверы или базы данных). Обычно это вызвано тем, что злоумышленник имеет определенный эксплойт или умеет проводить определенные виды атак, которые он хочет осуществить. Такие атаки обычно направлены на конкретные операционные системы или приложения. Сканеры могут искать компьютеры, на которых установлены определенные троянские программы или вирусы, рассчитывая на получение доступа к такой системе, которую они могут затем использовать для проведения DDoS-атак. Такие сканеры сканируют тысячи компьютеров и сетей, как правило, без каких-либо определенных целей. Они просто ищут любую уязвимость, для злоумышленника не имеет большого значения, на какой системе находится найденная уязвимость.

Иногда у злоумышленника есть конкретные цели, в таких случаях он не проводит столь широкого сканирования. После нахождения своей цели, злоумышленник составляет карту ее сети и проводит сканирование портов. Инструменты для создания карты сети отправляют, казалось бы, вполне легальные пакеты множеству различных систем в сети. Эти системы отвечают на полученные пакеты, а инструмент анализирует их ответы, выясняя, какие на них запущены службы, типы операционных систем и, возможно, их местоположение в сети. Может показаться, что из пакетов с такими ответами нельзя собрать много информации, но

используемые для этих целей инструменты очень хорошо откалиброваны для извлечения максимального количества сведений из получаемых ответных пакетов, насколько это возможно. Различные операционные системы имеют отличия в своих стеках протоколов, например, они могут использовать различные поля в кадрах протоколов или заполнять эти поля немного иначе, чем другие операционные системы. Если инструмент отправляет одинаковый запрос двум компьютерам, работающим под управлением различных операционных систем, полученные от них ответы будут, скорее всего, немного различаться. Хотя суть ответов обоих компьютеров вероятно будет одинаковой, но отдельные поля в ответных пакетах могут быть заполнены по-разному, как показано на рисунке 10-4. На основании этих различий в ответах, инструмент может определить тип операционной системы ответивших компьютеров. После этого злоумышленник с использованием инструмента собирает вместе всю полученную информацию и пытается на основе нее определить топологию сети жертвы.

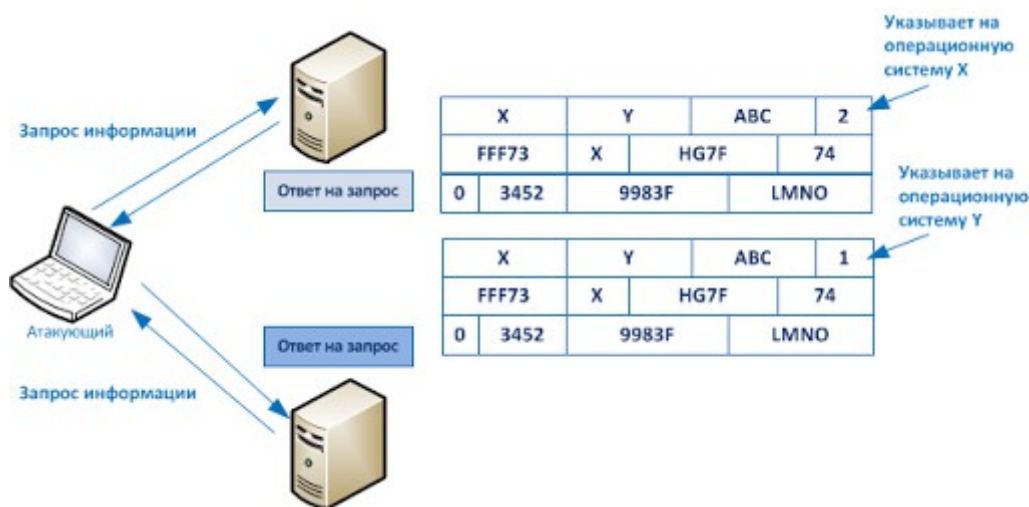


Рисунок 10-4. Существуют специальные программные инструменты, позволяющие определить тип операционной системы, с которой взаимодействует компьютер атакующего

Инструмент создания карты сети (network-mapping tool) может обладать базой данных, содержащей варианты ответов различных операционных систем и приложений в зависимости от их версий. Например, если инструмент отправил на целевой компьютер ICMP-пакет и получил от него в ответ пакет ICMP Ping Sweep в определенном бите которого установлен флаг «не фрагментировать», этот инструмент может определить, что на целевом компьютере установлена операционная система Unix. Если инструмент отправил пакет TCP SYN и в ответ получил FIN/ACK, инструмент может сделать вывод, что целевая система – Windows NT. Эта работа называется **снятием отпечатков операционной системы** (operating system fingerprinting).

Злоумышленнику важно знать, что за системы работают в атакуемой им сети. Отправляя первый пакет, инструмент смотрит, отвечает ли целевой компьютер – его ответ указывает на то, что по указанному адресу компьютер действительно существует, он включен и работает. Обычно, когда компьютер посылает пакет SYN с запросом на создание TCP-соединения с удаленным компьютером, первым шагом установления соединения является процедура "рукопожатия". Однако, хакерские утилиты, которые используют пакеты SYN, не намереваются создавать реальные соединения и обманывают надежды целевой системы, не посылая ответ. Правильный ответ на пакет SYN – это SYN/ACK, но администратор может заблокировать трафик такого типа на межсетевом экране или маршрутизаторе, либо заблокировать соответствующий порт на компьютере. В таком случае злоумышленник может не получить никакого ответа или получить сообщение «порт недоступен». Такой ответ сообщает инструменту создания карты сети, что межсетевой экран компании позволяет трафику этого типа войти в сеть, однако на целевой машине соответствующий порт закрыт.

Все эти сведения используются злоумышленником, чтобы узнать как можно больше об атакуемой среде, что позволит ему выбрать наиболее эффективные методы для проведения атаки.

Таким образом, первым шагом для атакующего является выявление работающих компьютеров, типов установленных на них операционных систем и начало построения топологии сети. Следующим шагом является выполнение **сканирования портов** на целевых компьютерах, позволяющее определить открытые на них порты. Эти порты открывают двери в операционную систему для других компьютеров, процессов, протоколов и... злоумышленников. Если злоумышленник может узнать, какие порты открыты, он может получить достаточно полное представление о том, какие службы работают на просканированных системах. Знание работающих служб может помочь дополнительно уточнить тип целевого компьютера. Например, если целевой компьютер отвечает на пакеты SMTP, отправленные на его порт 25, то злоумышленник может сделать вывод, что этот компьютер может быть сервером электронной почты. Однако необходимо будет выполнить дополнительные проверки, чтобы подтвердить это предположение.

Злоумышленник хочет знать, какая операционная система установлена на целевом компьютере, какие на нем работают приложения и службы. Это позволит ему понять, какую атаку нужно проводить в отношении этого компьютера. Если он находит сервер Unix с запущенным Apache, он будет выполнять совершенно другую атаку, чем для сервера с Windows 2000, на котором работает IIS. Каждая операционная система, приложение и служба имеют свои собственные уязвимости и правильное их определение позволяет злоумышленнику быть более успешным в своей атаке.

Каждая компьютерная система имеет 65535 TCP-портов и 65535 UDP-портов. Первые 1024 портов называют общеизвестными портами (well-known ports). Это означает, что каждый порт с номером до 1025, как правило, связан с общеизвестным и широко используемым протоколом. Например, порт 80 почти всегда связан с протоколом HTTP, порт 21 – с протоколом FTP, а порт 25 – с протоколом SMTP. Эти порты могут быть перенастроены, например, чтобы порт 21 был связан с HTTP, но это делается очень редко.

Утилита сканирования портов используется злоумышленником, чтобы выяснить, какие порты открыты в системе, и понять, какие ему доступны двери. Эта утилита отправляет пакеты на каждый порт в системе и «слушает» ответ. Отсутствие ответа или получение сообщения «порт недоступен» обычно указывает на то, что порт заблокирован и/или соответствующая служба отключена. Если от порта получен предсказуемый ответ, злоумышленник может сделать вывод, что порт открыт и доступен для атаки.

Порт должен быть открыт и доступен для использования. Например, для использования функциональности электронной почты в большинстве случаев требуется использование протокола SMTP (порт 25). Администратору необходимо установить барьер между потенциальными злоумышленниками и этим уязвимым портом и его службой. Конечно, первым шагом является создание сетевого периметра и его защита с помощью межсетевых экранов, прокси-серверов и маршрутизаторов, которые разрешают только определенные (допустимые) подключения к внутренним ресурсам и системам. Но если администратор или специалист по безопасности хочет установить еще один уровень защиты, ему следует внедрить **TCP Wrappers**. Эти программные компоненты (wrappers) отслеживают поступающие на компьютер входящие сетевые пакеты и проверяют, какой из них может, а какой не может получить доступ к службам, связанными с определенными портами. Когда запрос приходит на определенный порт компьютера, его операционная система проверяет, включен ли этот порт. Если порт включен и операционная система видит, что соответствующая служба защищена (wrapped), она проверяет список контроля доступа, чтобы принять решение, может ли этот запрос получить доступ к этой службе. Если человеку (или компьютеру), пытающемуся получить доступ к этой службе, доступ к ней (в

соответствии с ее списком контроля доступа) разрешен, операционная система позволяет создать подключение. В противном случае пакет уничтожается, либо отправителю запроса направляется сообщение об ошибке.

Итак, к настоящему моменту злоумышленник получил представление о том, какие системы работают в атакуемой сети, какие порты открыты на них, какие службы находятся в режиме прослушивания и доступны для атаки. Хотя поиск эксплуатируемых уязвимостей становится все более целенаправленным, нельзя забывать про то невероятное количество уязвимостей, которые может иметь одна сеть. Иногда нападающие специализируются на нескольких определенных атаках, которые они знают достаточно хорошо и могут проводить вручную, либо у них есть готовые скрипты или эксплойты, которые могут выполнить за них почти всю работу. Однако многие атакующие хотят получить как можно более широкий диапазон возможностей для проведения атак, поэтому они используют **средства сканирования уязвимостей**.

Средства сканирования уязвимостей обладают большой базой данных уязвимостей и могут эксплуатировать многие из выявляемых уязвимостей. В операционных системах, веб-серверах, базах данных и приложениях новые уязвимости обнаруживаются каждую неделю. Однако выявление и использование этих уязвимостей вручную является крайне сложной задачей. Сканеры уязвимостей могут упростить эту задачу для специалистов по безопасности и, к сожалению, для злоумышленников. Такие инструменты позволяют подключаться к целевой машине и проверять ее, основываясь на своей базе данных уязвимостей, чтобы найти уязвимости этой машины. Некоторые инструменты могут пойти дальше и позволить атакующему попытаться воспользоваться найденной уязвимостью, чтобы определить, действительно ли система ей подвержена.

Как отмечалось ранее, эти средства имеют двойное назначение. Администраторы и специалисты по безопасности должны использовать такие инструменты для поиска уязвимостей в своей среде. Важно, чтобы они обнаружили уязвимости раньше, чем это сделают злоумышленники, что позволит заранее установить необходимые патчи или внести изменения в настройки для закрытия уязвимостей или минимизации их влияния.

Браузинг

Браузинг (Browsing) – это распространенный способ, используемый злоумышленниками для получения информации, к которой у них нет доступа. Это атака, при выполнении которой злоумышленник ищет критичные данные, не зная их формата (текстовый редактор, электронная таблица, база данных, бумага). Браузинг может выполняться путем поиска данных в чужих файлах, сохраненных на сервере или рабочей станции, разгребания мусора в поисках небрежно выброшенной информации, или анализа информации, записанной на флеш-накопителях. Наиболее передовым и сложным примером браузинга является анализ злоумышленником остаточной информации на носителе и восстановление данных на ее основе. Пользователь может удалить файлы с флэш-накопителя, но, как отмечалось ранее, при этом удаляются только ссылки на файлы, а сами файлы остаются на носителе информации. Опытный взломщик может получить доступ к остаточной информации и восстановить на ее основе удаленные пользователем данные, к которым у него нет разрешения на доступ.

Другим видом атаки браузинга является "подглядывание через плечо", при котором злоумышленник пытается "подсмотреть" информацию на мониторе или нажатия клавиш на клавиатуре.

Снифферы

Сетевой сниффер – это инструмент, который выполняет мониторинг трафика, передающегося по сети. Администраторы и сетевые инженеры часто используют снифферы для диагностики сетевых проблем. Снифферы также называют сетевыми анализаторами и

анализаторами протоколов. При их использовании в качестве диагностического инструмента, они позволяют администратору увидеть типы генерируемого трафика, что может позволить ему приблизиться к источнику проблемы в сети. Если сниффер используется злоумышленником, он может перехватить с его помощью имена пользователей, пароли и другую конфиденциальную информацию, передаваемую по сети.

Сниффер, как правило, является частью более функционального программного обеспечения, работающего на компьютере с сетевой картой, настроенной на работу в режиме прослушивания (promiscuous mode). Обычно сетевая карта обращает внимание только на сетевые пакеты, направленные в ее адрес, однако, если она работает в режиме прослушивания, она принимает весь трафик, проходящий мимо нее по сетевым проводам. После взлома компьютера, злоумышленник часто выполняет установку на него сниффера для поиска интересного трафика. Некоторые снифферы являются специализированными и выполняют автоматический поиск в трафике только определенной информации (например, паролей), игнорируя все остальное.

В локальных сетях Ethernet (используемых в большинстве сред) снифферы работают весьма успешно, поскольку Ethernet является ширококестательной технологией. В таких сетях значительный объем данных постоянно рассылается ширококестательно, поэтому его легко может получить злоумышленник, установивший сниффер в сетевом сегменте. Однако с переходом на коммутируемые среды, эффективность снифферов снижается. Коммутируемые среды разделяют сетевые сегменты на ширококестательные и коллизийные домены. При этом злоумышленник сможет получить нужную ему информацию только в том случае, если он сможет установить сниффер в интересующем его коллизийном и ширококестательном домене среды. Это связано с тем, что коммутатор обычно передает трафик с порта отправителя непосредственно на порт получателя, то есть трафик не рассылается всем подряд. Коммутируемый трафик передается из точки А в точку Б через коммутатор и не распространяется на все компьютеры в сети, как это происходит в некоммутируемых сетях (ширококестательные и коллизийные домены подробно обсуждались в Домене 05).

Для противодействия работе снифферов в сети, по возможности должны использоваться безопасные версии служб и протоколов. Многие службы и протоколы были разработаны с учетом только их функциональности, но не безопасности. Однако после того, как необходимость безопасности стала очевидной, большинство этих служб и протоколов были усовершенствованы для обеспечения надлежащего уровня безопасности. Многие старые протоколы имеют более современную безопасную версию, которая не содержит уязвимостей, имевшихся в первоначальной версии протокола (службы). Одним из таких примеров является **Secure RPC** (S-RPC), в котором используется криптография с открытым ключом на базе алгоритма Диффи-Хеллмана для генерации и обмена общих секретным ключом, применяющимся при шифровании симметричным алгоритмом. Если в среде используется протокол S-RPC, сниффер может по-прежнему перехватить передаваемые данные, но далеко не всегда он сможет расшифровать их.

Большинство протоколов являются уязвимыми, поскольку они не требуют строгой аутентификации (или не требуют вообще никакой аутентификации). Например, r-утилиты, используемые в Unix (rexec, rsh, rlogin и rcp), как известно, имеют множество недостатков. Обычно они выполняют аутентификацию на основе проверки списка IP-адресов, содержащегося в файле .rhosts, а не идентификаторов пользователей и паролей. Следует заменить эти утилиты на те, которые требуют выполнения строгой аутентификации, например, Secure Shell (SSH).

Перехват коммуникационного сеанса

Многие злоумышленники подделывают свои адреса, поэтому в отправленных ими на целевую систему сетевых пакетах, обычно указан IP-адрес отправителя, не принадлежащий злоумышленнику. Это существенно усложняет поиск злоумышленника. Кроме того, это

позволяет злоумышленнику перехватить коммуникационный сеанс (Session Hijacking) между двумя пользователями, не будучи замеченным.

Чтобы перехватить сеанс связи между двумя компьютерами, злоумышленник должен встать между ними, но так, чтобы его не обнаружили. Для этих целей могут использоваться специальные утилиты, например, Juggernaut или HUNT Project. Эти утилиты позволяют злоумышленнику «шпионить» за TCP-соединением и в любой момент перехватить его.

Рассмотрим следующий пример. Кристи и Дэвид взаимодействуют посредством сеанса TCP. Злоумышленнику нужно исключить из сеанса Дэвида и перехитрить Кристи, чтобы она думала, что по-прежнему общается с Дэвидом. Для этого злоумышленник должен использовать утилиту для изменения своего адреса на адрес Дэвида и временно отключить Дэвида от сети, проведя DoS-атаку на его компьютер. После того, как он все это сделает, при дальнейшей отправке сообщений от Кристи Дэвиду, они будут фактически направляться злоумышленнику, который может отвечать Кристи. При этом Кристи будет думать, что она получила ответ от Дэвида. Злоумышленник может пойти и по другому пути, оставив Дэвида в сети. При этом он может перехватывать сообщения, которыми обмениваются Дэвид и Кристи, читать их, и заново формировать сетевые пакеты, изменяя их заголовки, чтобы они не указывали на перехват сеанса.

Если перехват сеанса является серьезной угрозой для сети, администратор может внедрить защищенные протоколы, такие как IPSec или Kerberos, которые требуют выполнения взаимной аутентификации между пользователями или системами. Поскольку у злоумышленника не будет необходимых полномочий для прохождения проверки подлинности от имени пользователя, он не сможет перехватить сеанс.

Loki

Распространенной современной атакой на скрытые каналы является атака Loki. Эта атака использует протокол ICMP для передачи данных, хотя этот протокол не был разработан для использования таким образом, он предназначен лишь для отправки сообщений о текущем статусе и ошибках. Но кто-то разработал специальный инструмент (Loki), который позволяет злоумышленнику записывать данные сразу после заголовка ICMP.

Это позволяет злоумышленнику организовать связь с другой системой посредством скрытого канала. Часто такая атака оказывается весьма успешной, поскольку большинство межсетевых экранов настроены на разрешение входящего и исходящего трафика ICMP. Это скрытый канал, т.к. он использует для связи протокол, который не был разработан для этого. Подробную информацию об атаке Loki можно найти на <http://xforce.iss.net/xforce/xfdb/1452>.

Взлом паролей

В Домене 02 детально рассматривались вопросы управления доступом и методы аутентификации. Хотя существуют различные способы аутентификации, компании чаще всего используют наиболее простой вариант – статические пароли, т.к. это дешево и все хорошо знают, как их использовать. Во многих системах и приложениях процессы аутентификации запрограммированы таким образом, что они поддерживают аутентификацию только с помощью паролей – это гораздо проще реализовать и сопровождать в дальнейшем, соответственно, это значительно удешевляет разработку по сравнению с другими вариантами, такими как смарт-карты и биометрия.

Однако статические пароли не очень сложно взломать, если злоумышленник воспользуется правильными инструментами. Примером такого инструмента является утилита John The Ripper, которая сочетает в себе множество различных техник взлома паролей. Также существуют и другие мощные инструменты (например, L0phtcrack), которые могут использоваться для выполнения атак по словарю и брутфорс-атак на перехваченный пароль или файл с паролями.

Основной контрмерой для противодействия взлому паролей является строгая парольная политика. Эта политика должна требовать использования паролей, длиной не менее восьми символов, включающих заглавные и строчные буквы, а также не менее двух специальных символов (*, \$, @). Если пароль является длинным, сложно угадываемым и содержит специальные символы, взломщику потребуется гораздо больше времени на его взлом. Чем дольше продолжается процесс взлома, тем выше вероятность того, что злоумышленник переключится на более легкую жертву. Существуют программные приложения и дополнения, которые позволяют проверить соблюдение пользователями требований по выбору паролей, установленных в политике безопасности компании.

Бэкдоры

В Домене 03 мы уже рассматривали бэкдоры и тот ущерб, который они могут нанести. Было описано, как бэкдоры вставляются в код программы, чтобы разработчик смог получить к ней доступ в более позднее время в обход обычной процедуры аутентификации и авторизации. Теперь мы посмотрим, каким образом и почему нападающие устанавливают бэкдоры на компьютеры жертв.

Бэкдор – это программа, которая устанавливается злоумышленником, чтобы он мог снова подключиться к взломанному компьютеру позднее, не вводя учетные данные и не проходя процедуру авторизации. Это нарушает работу системы управления доступом. Программа-бэкдор после установки начинает прослушивать определенный порт, настроенный злоумышленником. Когда злоумышленник отправит запрос на подключение к этому порту, программа предоставит ему доступ.

Злоумышленник может взломать компьютер и установить на него программу-бэкдор, либо поместить код бэкдора внутрь вируса или троянской программы, которые установят бэкдор, когда произойдет заранее определенное событие. В большинстве случаев, злоумышленник устанавливает бэкдор, чтобы иметь возможность позднее подключиться к взломанному компьютеру и управлять им, заставляя выполнять задачи, указанные злоумышленником. Наиболее известными инструментами, представляющими функциональность бэкдора, являются Back Orifice, NetBus и SubSeven.

Сегодня большинство антивирусных приложений и IDS имеют сигнатуры таких инструментов в своих базах данных, и отслеживают известные модели их поведения. Использование IDS уровня узла может быть одним из лучших способов обнаружить бэкдор. Для этого IDS должна быть настроена на выявление подозрительной активности при использовании сетевых портов, т.к. бэкдор будет прослушивать определенные порты, ожидая подключений злоумышленника. Кроме этого, администратор может выполнять сканирование файлов на компьютерах в поисках известных исполняемых файлов бэкдоров (или связанных с ними), проверять списки автоматически загружаемых программ, отыскивая в них подозрительные исполняемые файлы, а также проверять контрольные суммы системных и автоматически загружаемых файлов, чтобы убедиться, что они не были изменены. Такие действия обычно не делаются вручную, т.к. это требует слишком много времени. Обычно они выполняются автоматически с помощью антивирусного программного обеспечения и систем IDS. Но злоумышленники умны, они постоянно разрабатывают новые инструменты и дорабатывают старые инструменты, чтобы они не обнаруживались средствами безопасности. Эта игра в «кошки-мышки» продолжается на протяжении многих лет между хакерами и сообществом безопасности.

Ниже приведено краткое описание некоторых типов атак, с которыми вы должны быть знакомы:

- **DoS-атака** (Denial-of-Service attack). Злоумышленник посылает множество запросов на компьютер жертвы, пока они в конечном счете не выведут систему из строя, вызвав ее зависание, перезагрузку компьютера или другим способом не позволят выполнять

свои обычные задачи.

- **Атака «человек посередине» (Man-in-the-middle attack).** Нарушитель внедряется в сеанс связи между двумя компьютерами, после чего он может перехватывать и читать сообщения, которыми обмениваются стороны. С атакой этого типа можно бороться с помощью цифровых подписей и технологий взаимной аутентификации.
- **Бомбардировка электронной почтой (Mail bombing).** Это атака применяется для вывода из строя почтовых серверов и клиентов электронной почты, путем направления им большого количества сообщений электронной почты. Для противодействия этой атаке может использоваться фильтрация сообщений электронной почты и правильная настройка функций почтового ретранслятора на почтовых серверах.
- **Wardialing.** Это брутфорс-атака, выполняемая с помощью специальной программы, которая систематически набирает большое количество телефонных номеров в поисках модемов. Модемы могут предоставить легкий доступ в сеть компании. Контрмерами для противодействия этой атаке является запрет на публикацию телефонных номеров модемов в открытых источниках, а также выполнение строгого контроля доступа к модемам и модемным пулам.
- **Пинг смерти (Ping of death).** Это разновидность DoS-атаки, при которой жертве направляются пакеты ICMP увеличенного размера. Системы, уязвимые для такой атаки, не знают, что делать с такими ICMP-пакетами, это может привести к их зависанию или перезагрузке. Контрмерами против такой атаки является установка необходимых патчей на систему, внедрение средств для обнаружения таких пакетов.
- **Поддельный экран входа в систему.** Для выполнения этой атаки, злоумышленник создает поддельный экран входа в систему и устанавливает его на систему жертвы. Когда пользователь пытается войти в систему, ему показывается этот фальшивый экран, запрашивающий его реквизиты для аутентификации в системе. После того, как он вводит эти данные, программа, отображающая поддельный экран, завершается и пользователь видит настоящий экран регистрации в системе. При этом пользователь может подумать, что он просто ошибся при вводе пароля и, ничего не подозревая, пытается повторно пройти аутентификацию. Для обнаружения этого, может использоваться IDS уровня узла.
- **Teardrop.** При выполнении этой атаки, злоумышленник посылает жертве неправильно фрагментированные пакеты. При этом система жертвы не может правильно собрать пакет, что может привести к ее зависанию. Контрмерой против этой атаки является установка патчей на систему, и использование средств фильтрации трафика для выявления таких пакетов.
- **Анализ трафика.** Это метод раскрытия информации, реализуемый путем наблюдения за изменениями потоков трафика в сети. Например, интенсивный трафик между компьютерами отдела кадров и компьютерами руководства может свидетельствовать о предстоящем сокращении. Для борьбы с этой атакой могут применяться ложные потоки трафика, которые передаются по сети, чтобы скрыть реальные изменения и затруднить их обнаружение.
- **Slamming и Cramming.** Slamming – это изменение провайдера пользователя без его ведома. Cramming – это реализация загрузки ложной информации при любых запросах пользователя. Единственной контрмерой против таких атак является надлежащий мониторинг загружаемой информации.

В этом разделе были рассмотрены только некоторые из возможных атак. Одной из наиболее полезных вещей, которые сетевой администратор, сетевой инженер или специалист по

безопасности могут сделать для противодействия атакам, это следование лучшим практикам, выполнение мониторинга и регулярного контроля. Конечно, это хорошо, если администратор устанавливает обновления безопасности систем сразу после того, как он прочитает об опасной уязвимости, которая стала популярной у злоумышленников, но более важно, чтобы он своевременно узнавал о новых уязвимостях, регулярно сканировал свою сеть, помнил основные симптомы воздействия наиболее распространенных атак, вирусов, троянских программ и бэкдоров. Эти привычки позволят навести порядок и помогут поддерживать безопасность сети, делая жизнь администратора менее хаотичной. Одним из лучших способов обеспечения уверенности в том, что компьютер должным образом защищен, является самостоятельное выявление и устранение проблем, не дожидаясь, пока их найдут хакеры. Для этого выполняется тестирование уязвимостей.

11. Тестирование уязвимостей

Проведение ручного или автоматизированного (а лучше их сочетания) тестирования уязвимостей (vulnerability testing), требует наличия у компании сотрудников (либо заключения договора с консультантами), обладающих большим опытом в безопасности, а также высоким уровнем доверия. Даже самый лучший автоматизированный инструмент сканирования уязвимостей выдает результаты, которые могут быть неправильно интерпретированы (ложное срабатывание), либо выявленные уязвимости могут не иметь значения для вашей среды или быть компенсированы за счет различных защитных мер. С другой стороны, в сети может быть найдено две отдельных уязвимости, которые сами по себе несущественны, но вместе взятые они имеют критическое значение. И, конечно же, автоматизированный инструмент может пропустить отдельные уязвимости, например, уязвимости в малоизвестном элементе, имеющем большое значение для вашей среды.

ПРИМЕЧАНИЕ. Перед проведением тестирования уязвимостей, необходимо получить письменное согласие от руководства компании! Это защитит от судебного преследования тестировщика, выполняющего эту работу, и исключит любое недопонимание, т.к. все требования будут предоставлены в письменном виде и в них будет указано, что разрешается делать тестировщику, а что – нет.

Цели проведения такой оценки заключаются в следующем:

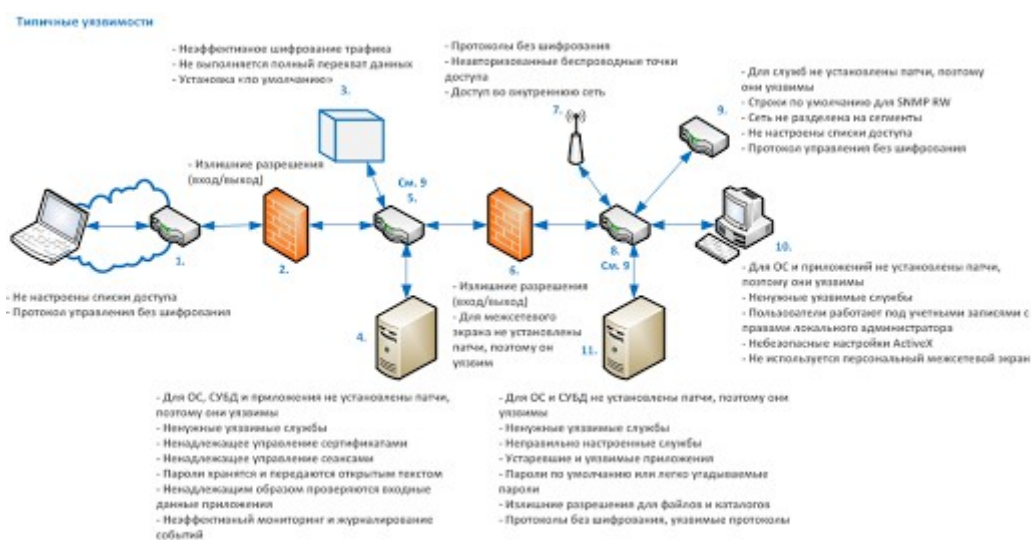
- Оценить истинное состояние безопасности среды.
- Выявить максимально возможное количество уязвимостей, оценить и приоритезировать каждую из них.
- Проверить, как системы реагируют на определенные действия и атаки, чтобы узнать не только о наличии известных уязвимостей (устаревшая версия службы, учетная запись без пароля), а также о возможности несанкционированного использования отдельных элементов среды (SQL-инъекции, переполнение буфера, использование архитектурных недостатков системы (например, в атаках социальной инженерии)).

Перед принятием решения о границах тестирования, тестировщик должен объяснить возможные последствия тестирования. Уязвимые системы могут быть выведены из строя некоторыми из тестов, проведение тестирования может негативно отразиться на производительности систем из-за дополнительной загрузки при их тестировании.

Кроме того, руководство должно понимать, что результаты тестирования – это только «моментальный снимок». Поскольку в среде постоянно происходят изменения, в любой момент могут появиться новые уязвимости. Руководство также должно понимать, что возможны различные варианты оценки, каждый из которых позволяет выявить различные виды уязвимостей в среде, но каждый из них имеет свои ограничения. Такими вариантами тестирования могут быть следующие:

- **Тестирование персонала**, которое включает в себя:

- анализ задач, выполняемых сотрудником, что позволит выявить уязвимости (недостатки) в сложившихся практиках и процедурах, которым должны следовать сотрудники;
 - демонстрацию атак социальной инженерии;
 - проверку результатов обучения пользователей по противодействию различным атакам;
 - анализ политик и процедур, которые должны соблюдать сотрудники, что позволит убедиться, что все риски, которые не могут быть сокращены за счет физических и технических мер, учтены с помощью административных мер.
- **Физическое тестирование**, которое включает в себя анализ механизмов защиты здания и периметра. Например, действительно ли двери закрываются автоматически? Раздается ли сигнал тревоги, если дверь остается открытой слишком долго? Работают ли внутренние защитные системы серверных комнат, коммутационных шкафов, помещений, в которых установлены критичные системы и/или активы? Правильно ли работают считыватели смарт-карт и действительно ли доступ предоставлен только уполномоченному персоналу? Является ли угрозой «разгребание мусора» (другими словами, выполняется ли надежное уничтожение носителей конфиденциальной информации при их выводе из эксплуатации, в т.ч. уничтожение бумажных документов)? Как обстоят дела с защитой от искусственных, природных и технических угроз? Работает ли система пожаротушения? Она безопасна для людей и оборудования в здании? Установлена ли критичная электроника выше фальшполов, чтобы она смогла пережить небольшое затопление? И так далее.
 - **Тестирование сети и систем** – возможно именно об этом думает большинство людей, когда речь заходит о тестировании уязвимостей в рамках информационной безопасности. Автоматизированный сканер уязвимостей находит известные уязвимости в системах, а в некоторых случаях (если руководство подписало разрешение на активное воздействие и приняло риски возможного возникновения перебоев в работе) пытается эксплуатировать выявленные уязвимости.



Поскольку оценка безопасности является «мгновенным снимком» состояния среды, такая оценка должна проводиться на регулярной основе. Низкоприоритетные или лучше защищенные части среды, либо менее рискованные части, могут сканироваться один или два раза в год. Высокоприоритетные, наиболее ценные системы (например, серверы системы электронной коммерции, промежуточное программное обеспечение, обеспечивающее их работу) следует сканировать почти непрерывно.

При использовании преимущественно автоматизированного поиска уязвимостей, следует использовать несколько различных автоматизированных сканеров или разные сканеры при последующих проверках. Ни один сканер не знает и не может найти все известные уязвимости. Производители различных инструментов сканирования обновляют базы данных уязвимостей своих продуктов с разной скоростью и могут добавлять отдельные уязвимости в различном порядке. Обязательно проводите обновление базы данных уязвимостей для каждого используемого сканера перед началом тестирования. Помимо автоматизированных инструментов, время от времени следует проводить тестирование уязвимостей вручную с участием экспертов, а также уточнять интерпретацию результатов автоматизированного тестирования. Также как и автоматизированные сканеры, ни один эксперт не сможет найти все возможные уязвимости.

11.1. Тестирование на проникновение

Тестирование на проникновение (penetration testing) представляет собой процесс моделирования атак на сети и системы по просьбе их владельца – руководителя высшего звена. При тестировании на проникновение тестировщик использует набор процедур и инструментов, предназначенных для тестирования и попыток обхода защитных мер системы. Целью тестирования на проникновение является оценка уровня сопротивления компании атаке и выявление любых недостатков в ее среде. Компании нужно независимо оценить эффективность своих мер безопасности, а не просто довериться обещаниям поставщиков. Хорошая компьютерная безопасность основывается на реальных фактах, а не на одном только представлении, как все должно работать.

Тестирование на проникновение имитирует те же методы, которые используют реальные злоумышленники. Нужно учитывать, что злоумышленники могут быть очень умными, творческими людьми, весьма изобретательными в своих подходах, поэтому тестирование на проникновение должно также использовать новейшие методы взлома наряду с прочной методологией проведения такого тестирования. В процессе тестирования нужно проанализировать каждый компьютер в среде, как показано на Рисунке 10-5, поскольку не стоит рассчитывать, что злоумышленник просканирует только один или два компьютера и, не найдя в них уязвимостей, выберет другую компанию.

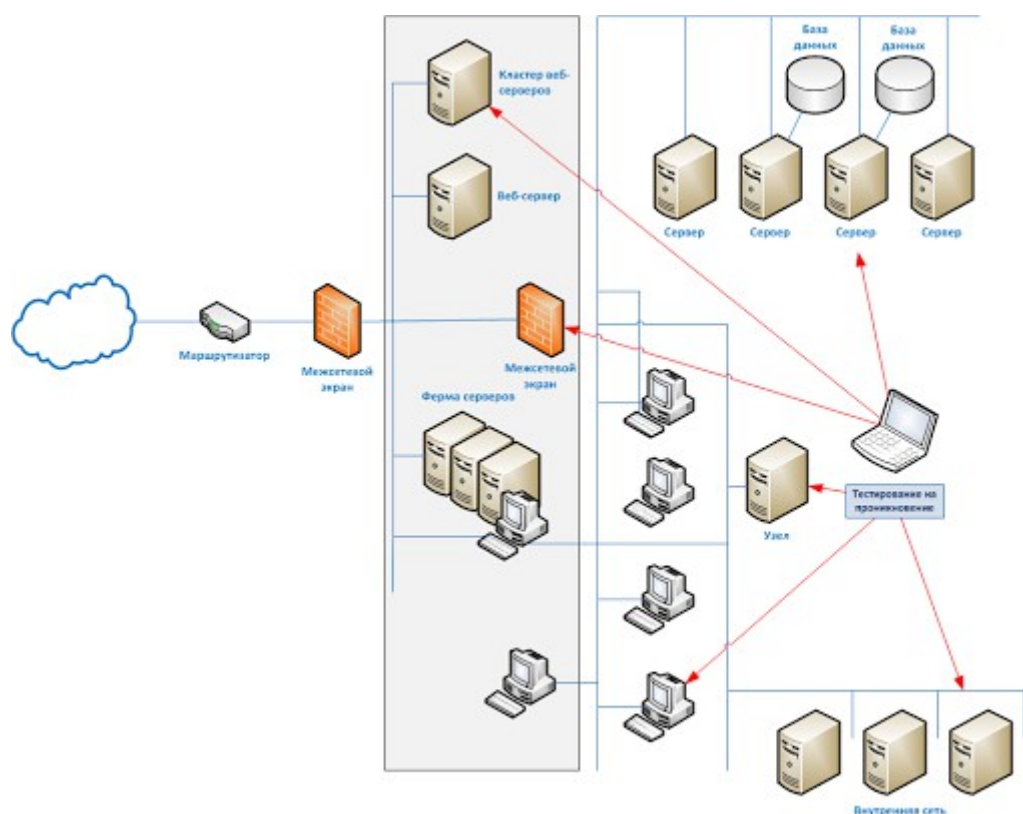


Рисунок 10-5. Тестирование на проникновение проводится, чтобы доказать, что злоумышленник действительно может взломать систему

Возможности сканеров уязвимостей. Сканеры уязвимостей предоставляют следующие возможности:

- Выявление активных систем в сети
- Выявление активных уязвимых служб (портов) на найденных системах
- Выявление работающих на них приложений и анализ баннеров
- Определение установленных на них операционных систем
- Выявление уязвимостей, связанных с обнаруженными операционными системами и приложениями
- Выявление неправильных настроек
- Тестирование на соответствие с политиками использования приложений и политикам безопасности
- Подготовка основы для проведения тестирования на проникновение

Выбор варианта тестирования на проникновение зависит от компании, ее целей в отношении безопасности и целей ее руководства. Некоторые крупные компании регулярно выполняют тестирование на проникновение в свою среду, используя различные виды инструментов, либо применяя сканирующие устройства, которые непрерывно анализируют сеть компании, автоматически выявляя в ней новые уязвимости. Другие компании обращаются к поставщикам соответствующих услуг для выявления уязвимостей и проведения тестирования на проникновение, чтобы получить более объективное мнение о защищенности своей среды.

При тестировании на проникновение могут быть проверены веб-серверы, DNS-серверы, настройки маршрутизаторов, проанализированы уязвимости рабочих станций, проверена возможность доступа к критичной информации, проверены системы удаленного доступа, открытые порты, свойства доступных служб и все остальное, чем может воспользоваться реальный злоумышленник, чтобы получить несанкционированный доступ к защищаемым информационным активам компании. Некоторые тесты могут оказывать негативное влияние на работу систем, выводить их из строя. Сроки проведения тестирования должны быть заранее согласованы. В процессе тестирования не должно оказываться существенного влияния на производительность работы компании, а персонал компании должен быть готов при необходимости оперативно восстановить работу систем.

По результатам тестирования на проникновение должен быть оформлен отчет, описывающий выявленные уязвимости и степень их критичности, а также рекомендации по их исправлению. Этот отчет должен быть предоставлен руководству компании. На основании отчета руководство должно определить, с чем в действительности связаны обнаруженные уязвимости, и насколько эффективны реализованные контрмеры. Крайне важно, чтобы руководители высшего уровня хорошо понимали риски, связанные с проведением тестирования на проникновение - соответствующая информация должна быть предоставлена им перед тем, как они дадут свое разрешение на проведение тестирования. Это связано с тем, что в отдельных случаях используемые инструменты и методики тестирования на проникновение могут вывести из строя системы или приложения. Целью тестирования на проникновение является поиск уязвимостей, оценка реальной эффективности используемых компанией мер и средств безопасности, анализ реагирования систем и персонала безопасности на подозрительную активность, анализ выдаваемых системами предупреждений (которые могут и не появиться).

Специалисты по безопасности перед проведением тестирования на проникновение должны получить официальный документ (письмо) от руководства компании, в котором указаны, в частности, разрешенные границы тестирования. Этот документ должен быть доступен всем членам команды, участвующим в процессе проведения тестирования. Этот документ часто называют «пропуском на выход из тюрьмы» (Get Out of Jail Free Card). Кроме того, участникам тестирования должна быть доступна контактная информация ключевого

персонала компании, и «дерево вызовов» на случай, если что-то пойдет не по плану, и потребуется восстановить систему.

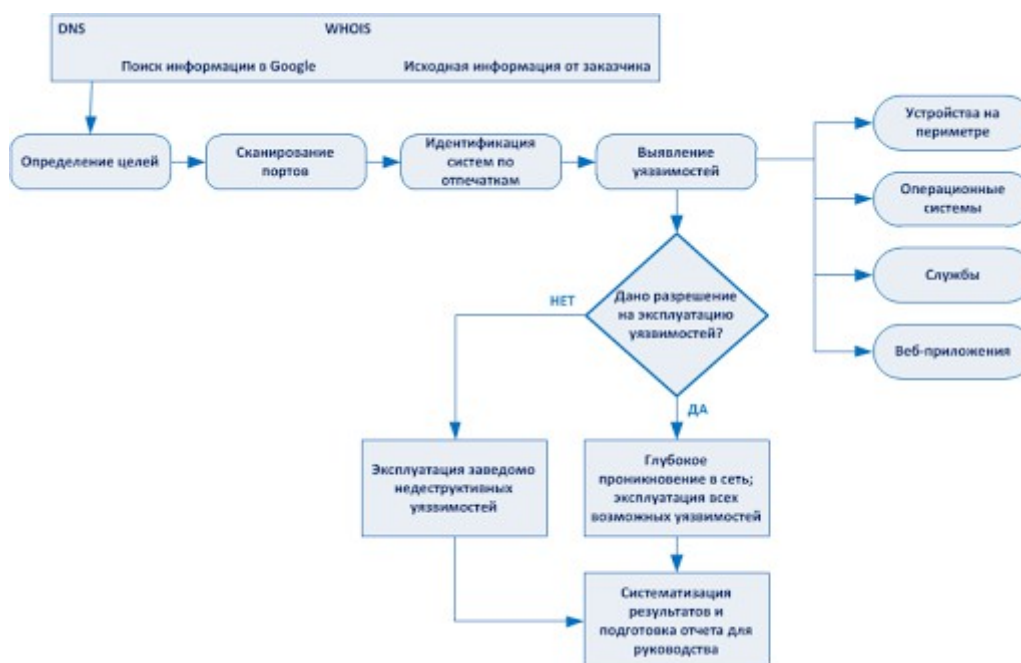
ПРИМЕЧАНИЕ. «Пропуск на выход из тюрьмы» – это документ, который вы можете предъявить любому, кто сочтет, что вы осуществляете незаконную деятельность, когда на самом деле вы проводите разрешенное тестирование. Нередко случались ситуации, когда эксперт (или группа экспертов) проводили тест на проникновение, а к ним подходили ничего не знающие об этом охранники, которые считали, что он оказался в неправильном месте в неправильное время.

В процессе выполнения тестирования на проникновение, команда проходит через пять этапов:

1. **Исследование.** Сбор информации о цели.
2. **Перечисление (enumeration).** Проведение сканирования портов, использование техник и инструментов для идентификации обнаруженных систем и ресурсов.
3. **Выявление уязвимостей (vulnerability mapping).** Выявление уязвимостей в идентифицированных системах и ресурсах.
4. **Эксплуатация.** Попытки получить несанкционированный доступ с помощью выявленных уязвимостей.
5. **Отчет руководству.** Предоставление руководству документированных результатов тестирования и предложений по устранению выявленных недостатков (контрмерам).

Команде тестирования на проникновение до начала тестирования может быть предоставлен различный объем информации о цели, в отношении которой осуществляется проникновение.

- **Нулевая информация.** Команда не имеет никакой информации о цели и должна начинать с нуля.
- **Частичная информация.** Команда имеет некоторые сведения о цели.
- **Полная информация.** Команда имеет полную и детальную информацию о цели.



Типы тестирования. Тестирование уязвимостей позволяет выявить широкий круг уязвимостей в среде. Обычно оно выполняется с помощью инструментов сканирования. В отличие от этого, при тестировании на проникновение специалисты по безопасности эксплуатируют одну или несколько уязвимостей, чтобы продемонстрировать заказчику (или руководству компании), что хакер реально может получить доступ к корпоративным ресурсам.

Тестирование безопасности среды может выполняться в различных формах, в зависимости

от объема знаний, которые тестировщику разрешается иметь о среде перед началом тестирования, а также от степени разрешенной информированности персонала компании о проводимом тестировании перед его началом.

Тестирование может проводиться извне (из удаленного места) или изнутри (т.е. тестировщик находится в сети компании). Следует проводить оба варианта тестирования, чтобы понять как внешние, так и внутренние угрозы.

Тесты могут проводиться на основе слепого метода, двойного слепого метода, или быть целенаправленными. При тестировании **слепым методом** (blind test), эксперты знают только общедоступные данные перед началом тестирования. Персонал компании, занимающийся обслуживанием сети, и сотрудники безопасности знают о проводимом тестировании.

Двойной слепой метод тестирования (double-blind test) (скрытая оценка) похож на слепой метод тестирования, однако персонал компании не ставится в известность о проводимом тестировании (в т.ч. сотрудники безопасности). Это позволяет проверить не только уровень безопасности сети, но и реакцию сотрудников, реальное выполнение ими функций мониторинга журналов регистрации событий, знание процедур реагирования на инциденты и эскалации. Такой метод тестирования является наиболее реалистичной демонстрацией вероятности успеха или провала вероятной атаки.

Для **целенаправленного тестирования** (targeted test) могут привлекаться внешние консультанты и внутренний персонал. При этом осуществляется тестирование, ориентированное на конкретные области, представляющие интерес для компании. Например, перед внедрением нового приложения компания может решить проверить наличие в нем уязвимостей до того, как оно будет установлено в промышленную среду. Другим примером является тестирование, ориентированное на конкретные системы, участвующие, например, в выполнении операций дистанционного банковского обслуживания. Остальные системы при этом в тестировании не участвуют.

Важно, чтобы команда начинала работать, имея только права обычного пользователя, что позволит более реалистично имитировать атаки. Команда должна использовать различные инструменты и методы атак, рассматривать все возможные уязвимости – так, как это будут делать настоящие злоумышленники.

В следующих разделах мы рассмотрим некоторые действия, обычно выполняемые в процессе тестирования на проникновение.

11.2. Сканирование телефонных номеров

Как было сказано ранее, инструменты для выполнения сканирования телефонных номеров (wardialing) позволяют злоумышленникам (или администраторам) автоматически набирать большие диапазоны телефонных номеров для поиска доступных модемов. Существует ряд бесплатных и коммерческих инструментов, которые могут набирать все телефонные номера в указанном диапазоне (например, все номера от 212-55-00 до 212-55-99) и отмечать номера телефонов, с которых ответили модемы. Обычно диапазон номеров выбирается таким образом, чтобы в него входили только номера, принадлежащие компании (по возможности). Такие инструменты могут быть достаточно «умными», звоня лишь ночью, когда большинство телефонов не контролируется. Также они могут выбирать телефонные номера в случайном порядке, чтобы снизить вероятность того, что сотрудники поднимут тревогу, услышав звонки один за другим по всем телефонам. Сканирование телефонных номеров может быть проведено достаточно быстро с использованием дешевого оборудования. Разработанные для этих целей инструменты могут не только выявить модем, но и определить тип ответившей системы на основе ее отклика (аналогично тому, как это делают сканеры уязвимостей), а также попытаться автоматически произвести попытки подключения к сети, и, в случае удачи, предоставить атакующему консоль взломанной системы, готовой выполнять его команды. Следует отметить, что некоторые офисные телефонные системы

(или специальные телефонные диагностические средства) могут автоматически обнаруживать модемные линии и сообщать о них соответствующим сотрудникам компании.

Самотестирование. Некоторые из тактик, используемых злоумышленниками при проведении сканирования телефонных номеров, могут быть полезны и для системного администратора – например, проведение сканирования ночью, чтобы не мешать работе сотрудников. Однако нужно при этом помнить, что обзвон телефонных номеров ночью может не позволить обнаружить модемы, подключенные к системам, которые пользователи выключают в конце рабочего дня. Инструменты wardialing могут быть настроены на пропуск определенных номеров или диапазонов номеров, чтобы они не набирали номера, в отношении которых системный администратор точно знает, что они являются голосовыми, например, номер службы технической поддержки. Такие же исключения могут быть настроены на офисных телефонных станциях, чтобы они не блокировали номера, с которых должны работать разрешенные модемы.

Любые факты обнаружения в процессе сканирования несанкционированно установленных модемов должны быть расследованы. Использование этих модемов должно быть либо официально разрешено (при этом должны быть предприняты все необходимые меры безопасности), либо эти модемы должны быть удалены, а установившие их сотрудники должны пройти дополнительное обучение или получить дисциплинированное взыскание.

11.3. Другие виды уязвимостей

Как было отмечено ранее, сканеры уязвимостей находят потенциальные уязвимости. Для определения того, могут ли они действительно быть использованы и нанести ущерб среде, должно проводиться тестирование на проникновение.

Чаще всего, эксплуатируемыми уязвимостями являются:

- **Недостатки на уровне ядра (kernel flaws).** Это проблемы, которые возникают ниже уровня пользовательского интерфейса, они находятся глубоко внутри операционной системы. Любой недостаток в ядре, который может быть использован злоумышленником, при его эксплуатации может предоставить злоумышленнику самый полный уровень контроля над системой.
 - **Контрмеры.** Необходимо максимально быстро устанавливать на системы патчи, выпускаемые производителями, после проведения их надлежащего тестирования.
- **Переполнение буфера.** Использование разработчиками небезопасных практик программирования, а также различные ошибки в библиотеках, позволяют программе получать больше входных данных, чем эта программа выделила места для их хранения. Это приводит к тому, что полученные данные при записи в буфер выходят за его пределы и перезаписывают область данных или исполняемого кода программы в памяти, что позволяет злоумышленнику внедрить в программу вредоносный программный код и заставить процессор выполнить его. Это дает злоумышленнику возможность получить такой же уровень доступа, который есть у программы, на которую произведена атака. Если программа была запущена от имени административной или системной учетной записи, это может означать полный доступ злоумышленника к системе.
 - **Контрмеры.** Необходимо использовать безопасные практики программирования, применять автоматизированные сканеры исходных кодов, расширенные библиотеки программирования, а также языки программирования со строгой типизацией, которые не позволяют переполнить выделенные буферы, снижая эту чрезвычайно распространенную уязвимость.
- **Символические ссылки (symbolic links).** Хотя возможности доступа злоумышленника к просмотру или изменению содержания критичных системных файлов и данных могут быть заблокированы, если программа использует символическую ссылку (файл-заглушка, который перенаправляет

приложение/пользователя в другое место, где хранится реальный файл), а злоумышленник имеет возможность внесения изменений в содержимое этой символической ссылки, он может получить несанкционированный доступ к системе (символические ссылки используются в системах Linux и Unix). Это может позволить злоумышленнику повредить важные данные и/или получить привилегированный доступ к системе. Примером может быть использование символической ссылки для удаления файла паролей или замены в нем отдельных строк, задающих пустой или заведомо известный злоумышленнику пароль для нужной ему учетной записи (например, root).

- **Контрмеры.** Программы и особенно скрипты должны быть написаны так, чтобы гарантировать невозможность изменения полного пути к файлу.
- **Атаки на файловый дескриптор.** Дескрипторы файлов – это числовые значения, которые многие операционные системы используют для представления открытых файлов в процессе работы с ними. Некоторые номера дескрипторов файлов являются универсальными, то есть одними и теми же для любых программ. Если программа использует дескрипторы файлов небезопасным образом, злоумышленник может передать программе неожиданные входные данные или сделать так, чтобы результаты были записаны в неправильное место с привилегиями запущенной программы.
 - **Контрмеры.** Использование безопасных практик программирования, автоматизированных сканеров исходных кодов, проведение тестирования безопасности приложений. Все это способы для сокращения такого рода уязвимостей.
- **Состояние гонки (race conditions).** Это состояние вызывается ошибкой в архитектуре многозадачной системы, при которой работа системы зависит от того, в каком порядке выполняются части ее кода. Примером может быть открытие временных файлов при работе в привилегированном режиме без обеспечения невозможности перезаписи этих файлов неуполномоченным пользователем или процессом, либо создание экземпляра функции динамически загружаемой библиотеки без обеспечения безопасности указателя, содержащего путь к файлу динамической библиотеки. Все это может позволить злоумышленнику нарушить работу программы (работающей с повышенными привилегиями) или заставить ее выполнять команды злоумышленника.
 - **Контрмеры.** Использование безопасных практик программирования, автоматизированных сканеров исходных кодов, проведение тестирования безопасности приложений – все это может помочь сократить уязвимости такого рода.
- **Разрешения для файлов и каталогов.** Многие из описанных выше атак основаны на использовании неправильно настроенных разрешений для доступа к файлам и каталогам, т.е. ошибках в управлении доступом к отдельным частям системы, от которых зависят другие части системы, требующие большей безопасности. Кроме того, ошибки системного администратора могут привести к установке небезопасных разрешений для доступа к критическим файлам, таким, как файл с базой данных учетных записей и паролей пользователей или конфигурационный файл, в котором указываются стандартные пути поиска исполняемых файлов и библиотек. Злоумышленник может воспользоваться этой возможностью, чтобы добавить своего пользователя или указать недоверенный каталог, в котором будет расположена вредоносная библиотека.
 - **Контрмеры.** Выполнение проверки целостности файлов, в процессе которой должны также проверяться действующие разрешения для файлов и каталогов. Это позволит своевременно выявить несанкционированно измененные файлы

или недопустимые разрешения доступа.

Существует множество различных видов уязвимостей, мы рассмотрели только некоторые из них, о которых вы должны знать для успешной сдачи экзамена.

11.4. Что дальше?

После завершения тестирования, интерпретации его результатов и расстановки приоритетов мероприятий по устранению выявленных недостатков, руководство получит подробный список способов, с помощью которых компания может быть успешно атакована. Это является входными данными для корректировки и реализации следующего этапа стратегии повышения безопасности. У любой компании существуют ограничения по бюджету и человеческим ресурсам, поэтому ни одна компания не может полностью устранить все свои риски – она может их только снизить до приемлемого уровня. Обеспечение баланса между рисками, риск-аппетитом компании и затратами на возможное снижение рисков, позволит компании принять наиболее правильные решения по расходу своих ограниченных ресурсов. Необходимо организовать контроль, который позволит убедиться, что риски снижены до приемлемого уровня, а расходы на защитные средства надлежащим образом отслеживаются и сравниваются с первоначальной сметой. Каждый случай, когда реальная стоимость значительно отличается от ожидаемой (как в положительную, так и в отрицательную сторону), должен детально анализироваться, а процесс внедрения должен приостанавливаться на время проведения анализа. Вполне возможно, что повысившаяся стоимость контрмер потребует пересмотра решения по их внедрению, и компания выберет принятие соответствующих рисков, либо их снижение альтернативным способом.

Когда, наконец все риски действительно снижены до приемлемого уровня, всем станет немного легче. За исключением, может быть, инженеров безопасности, в задачи которых входит обслуживание всех внедренных защитных механизмов, ведение мониторинга, анализ сообщений об обнаружении новых уязвимостей, работа со службами по раннему предупреждению об угрозах, предлагаемыми некоторыми поставщиками. Среда рисков постоянно меняется. Ведение мониторинга может помочь компании узнать о наличии в своей среде недавно выявленных уязвимостей до того, как будет проведено следующее плановое тестирование. Они могут быть слишком критичными, чтобы ждать так долго. Поэтому должна быть предусмотрена возможность выполнения еще одного, меньшего цикла принятия и корректировки решений по снижению рисков информационной безопасности.

В Таблице 10-3 приведен пример графика проведения тестирования, который должны совместно разработать Департамент эксплуатации и Департамент безопасности, и в дальнейшем следовать ему.

Тип тестирования	Периодичность	Результаты
Сканирование сети	Постоянно, либо ежеквартально	<ul style="list-style-type: none"> • Прохождение по сетевой структуре, выявление активных узлов в сети и установленного на них программного обеспечения • Выявление узлов, несанкционированно подключенных к сети • Идентификация открытых портов • Выявление фактов работы неразрешенных служб
Сканирование телефонных номеров	Ежегодно	<ul style="list-style-type: none"> • Выявление несанкционированно установленных модемов и предотвращение несанкционированного доступа в защищаемую сеть
Сканирование беспроводных сетей	Постоянно, либо еженедельно	<ul style="list-style-type: none"> • Выявление несанкционированно установленных беспроводных точек доступа и предотвращение несанкционированного доступа в защищаемую сеть
Выявление вирусов	Еженедельно или при необходимости	<ul style="list-style-type: none"> • Выявление и удаление вирусов до того, как они будут успешно установлены в систему
Анализ журналов регистрации событий	Ежедневно для критических систем	<ul style="list-style-type: none"> • Проверка соответствия работы систем действующей политике
Взлом паролей	Постоянно или с той же периодичностью, что и срок действия паролей, указанный в политике	<ul style="list-style-type: none"> • Проверка, что политика действительно обеспечивает использование надежных паролей, которые сложно взломать • Проверка, что пользователи действительно используют пароли, соответствующие политике безопасности компании
Сканирование уязвимостей	Ежеквартально, или каждые два месяца (для наиболее критических систем), либо по мере обновления базы данных уязвимостей	<ul style="list-style-type: none"> • Прохождение по сетевой структуре, выявление активных узлов в сети и установленного на них программного обеспечения • Определение целевого набора компьютеров, на которых нужно сосредоточиться при проведении анализа уязвимостей • Выявление потенциальных уязвимостей на системах из целевого набора • Проверка установки обновлений, патчей и актуальности версий операционной системы и основного программного обеспечения
Тестирование на проникновение	Ежегодно	<ul style="list-style-type: none"> • - Определение, насколько сеть компании уязвима для вторжения злоумышленника, а также уровня последствий, к которым может привести такое вторжение • - Проверка реагирования персонала ИТ на инциденты безопасности, а также уровня знания ими и реализации политики безопасности компании и требований по безопасности систем
Проверка целостности	Ежемесячно и в случае подозрительных событий	<ul style="list-style-type: none"> • Выявление несанкционированных изменений файлов

Таблица 10-3. Пример графика проведения тестирования для Департаментов эксплуатации и безопасности

В этом Домене и в предыдущих Доменах мы рассмотрели некоторые методы обеспечения гарантий, которые позволят убедиться, что Департамент эксплуатации правильно и эффективно выполняет возложенные на него обязанности.

Ссылки по теме:

- NIST Security Self-Assessment Guide for Information Technology Systems, by Marianne Swanson, NIST Special Publication 800-26 (Nov. 2001)
- Computer Security course, Module 16, “Vulnerability Analysis,” Polytechnic University (Nov. 2003)
- “DDoS: A Look Back from 2003,” presentation by Dave Dittrich, University of Washington
- “Password Cracking, Sniffing, and Man-in-the-Middle,” Prof. Henry Owen, Georgia Tech University

12. Резюме

Операционная безопасность включает в себя поддержку внедренных решений, отслеживание изменений, надлежащее сопровождение систем, непрерывное соблюдение необходимых стандартов, а также следование практикам и задачам безопасности. Компания не получит существенных преимуществ от разработки строгой парольной политики, если по прошествии нескольких месяцев окажется, что никто ее не соблюдает, а пользователи используют любые пароли, какие хотят. Это похоже на принятие решения о переходе к

здоровому образу жизни. Если вы неделю посещали спортзал, а потом до конца года ели пончики, то нельзя ожидать, что вы сможете обрести или сохранить хорошую физическую форму. Безопасность требует ежедневного соблюдения дисциплины, установленного режима, соблюдения принципа должной заботы.

Тест

Вопросы экзамена CISSP являются концептуальными, поэтому они сформулированы соответствующим образом. Задачей кандидата является выбор наилучшего из всех представленных вариантов ответа. Среди вариантов ответа может не быть идеального ответа на поставленный вопрос - кандидат должен выбрать лучший ответ из имеющихся вариантов.

1. Что из приведенного ниже лучше всего описывает операционную безопасность?

- ☐ A. Постоянное отслеживание возможных действий хакеров и выявление уязвимостей
- ☐ B. Реализация управления доступом и физической безопасности
- ☐ C. Выполнение шагов, позволяющих убедиться, что для среды и всего, что находится внутри нее, сохраняется определенный уровень защиты
- ☐ D. Выполнение стратегического планирования для разработки безопасной среды и последующего ее правильной реализации

2. Что из перечисленного ниже определяет причины важности операционной безопасности?

- ☐ A. Среда постоянно изменяется и существует вероятность снижения уровня ее безопасности
- ☐ B. Она помогает сохранять функциональность и продуктивность среды
- ☐ C. Она обеспечивает защиту от несанкционированного доступа в к информационным ресурсам
- ☐ D. Она постоянно повышает уровень защищенности компании

3. В чем заключается разница между должной заботой (due care) и должной осмотрительностью (due diligence)?

- ☐ A. Должная забота – это постоянная работа, направленная на то, чтобы все в компании было правильно, а должная осмотрительность – это постоянная работа, направленная на сохранение соответствия требованиям регуляторов
- ☐ B. Должная забота и должная осмотрительность являются противоположностью концепции «разумного человека» (prudent person)
- ☐ C. Эти два термина означают одно и то же
- ☐ D. Должная осмотрительность затрагивает вопросы анализа рисков, тогда как должная забота обеспечивает выполнение необходимых шагов для снижения этих рисков

4. Почему работодателю следует убеждаться, что сотрудники берут положенный им отпуск?

- ☐ A. Это его обязанность, установленная законом
- ☐ B. Это является частью принципа должной осмотрительности
- ☐ C. Это способ выявления мошенничества
- ☐ D. Чтобы сотрудники не переутомлялись

5. Что из перечисленного ниже лучше всего описывает разделение обязанностей (separation of duties) и ротацию обязанностей (job rotation)?

- ☐ A. Разделение обязанностей обеспечивает наличие в компании более одного сотрудника знающего, как выполнять работу определенной должности, а ротация обязанностей исключает возможность единоличного выполнения одним человеком высокорискованных задач
- ☐ B. Разделение обязанностей исключает возможность единоличного выполнения одним человеком высокорискованных задач, а ротация обязанностей обеспечивает наличие в компании более одного сотрудника знающего, как выполнять работу определенной должности
- ☐ C. Это одно и то же, просто по-разному называется
- ☐ D. Это административные меры, которые помогают реализовать управление доступом и защиту ресурсов компании

6. Если программисту запрещен доступ на обновление и изменение кода в промышленной среде, примером чего это является?

- ☐ A. Ротации обязанностей
- ☐ B. Должной осмотрительности
- ☐ C. Разделения обязанностей
- ☐ D. Контроля входных данных

7. Почему так важно контролировать и отслеживать входные и выходные данные?

- ☐ A. Некорректные значения могут вызвать ошибки при обработке данных, в некоторых случаях они могут свидетельствовать о мошенничестве
- ☐ B. Некорректные значения могут быть следствием ошибки программиста, они могут привести к нарушению принципа должной заботы
- ☐ C. Некорректные значения могут быть вызваны брутфорс-атакой
- ☐ D. Некорректные значения не являются проблемой безопасности

8. В чем отличие между принципами наименьших привилегий и «необходимо знать»?

- ☐ А. Пользователю должны быть предоставлены наименьшие привилегии, ограничивающие то, что ему «необходимо знать»
- ☐ В. Пользователю необходим определенный уровень допуска для использования ресурсов – это принцип «необходимо знать», а принцип наименьших привилегий дает ему полный доступ к этим ресурсам
- ☐ С. Для доступа к определенным ресурсам пользователю необходим атрибут «необходимо знать», а принцип наименьших привилегий должен быть реализован для того, чтобы этот пользователь мог получить доступ только к ресурсам, которые ему «необходимо знать»
- ☐ D. Эти два термина обозначают одно и то же

9. Что из перечисленного ниже не требует обновления документации?

- ☐ А. Обновление антивирусных сигнатур
- ☐ В. Изменение настроек сервера
- ☐ С. Изменение политики безопасности
- ☐ D. Установка патча на систему в промышленной эксплуатации

10. Если критичные данные, записанные на компакт-диске (CD-ROM), больше не нужны, какой вариант уничтожения данных будет правильным?

- ☐ А. Размагничивание (degaussing)
- ☐ В. Стирание (erasing)
- ☐ С. Очистка (purging)
- ☐ D. Физическое уничтожение (physical destruction)

11. Что является основной проблемой безопасности при использовании SSL для шифрования передаваемых сообщений?

- ☐ А. Различие версий SSL на различных системах в различных сетевых сегментах
- ☐ В. Пользователь может получить сообщение, зашифрованное прикладной программой, несовместимой с SSL
- ☐ С. Перехват передаваемых сообщений
- ☐ D. Сети, в которые передаются сообщения, не контролируются компанией

12. Зачем нужен протокол SMTP?

- ☐ А. Он позволяет пользователям расшифровывать полученные с сервера сообщения
- ☐ В. Он позволяет пользователю просматривать и изменять почтовые сообщения на сервере
- ☐ С. Для передачи почтовых сообщений с компьютера пользователя на почтовый сервер
- ☐ D. Для шифрования почтовых сообщений перед их отправкой

13. Если компании сообщили, что ее почтовый сервер используется для распространения спама, что может являться наиболее вероятной проблемой?

- ☐ А. Внутренний почтовый сервер был взломан внутренним злоумышленником
- ☐ В. Почтовый сервер в DMZ содержит записи внутренних и внешних ресурсов
- ☐ С. На почтовом сервере неправильно настроена ретрансляция почтовых сообщений
- ☐ D. На почтовом сервере включено использование SMTP

14. Что из перечисленного ниже не является причиной использования факс-серверов многими компаниями?

- ☐ А. Они позволяют сэкономить деньги, поскольку компании не нужны отдельные факсимильные аппараты и бумага для факсов
- ☐ В. Они позволяют безопасно отправлять факсы в отличие от обычных факсимильных аппаратов, в которых принятые факсы распечатываются и лежат в лотке, ожидая, пока кто-нибудь заберет их
- ☐ С. Факсы могут автоматически направляться сотрудникам в электронном виде по внутренней электронной почте
- ☐ D. Они повышают потребности в обеспечении безопасности других коммуникационных механизмов

15. Если компания хочет защитить данные факсов на этапе их передачи, какой из приведенных ниже механизмов ей следует выбрать?

- ☐ А. PGP и MIME
- ☐ В. PEM и TLS
- ☐ С. Канальное шифрование и факс-шифратор
- ☐ D. Канальное шифрование и MIME

16. Зачем нужны TCP wrappers?

- ☐ А. Обеспечение мониторинга запросов на определенные порты и контроля доступа к критичным файлам
- ☐ В. Обеспечение мониторинга запросов к определенным службам и контроля доступа к файлам с паролями
- ☐ С. Обеспечение мониторинга запросов к определенным службам и контроля доступа к ним
- ☐ D. Обеспечение мониторинга запросов к системным файлам и ограничение возможностей для их изменения

17. Как работает сетевой sniffер?

- ☐ A. Он исследует системы в сетевом сегменте
- ☐ B. Он прослушивает ARP-запросы и ICMP-пакеты
- ☐ C. Он требует установки и настройки дополнительной сетевой карты
- ☐ D. Он переводит сетевую карту в режим прослушивания

18. Что из перечисленного ниже не является атакой, направленной на функционирование?

- ☐ A. Брутфорс-атака
- ☐ B. DoS-атака
- ☐ C. Переполнение буфера
- ☐ D. ICMP Sting

19. Зачем нужно сохранять идентификаторы пользователей при записи информации о событиях в журналы регистрации событий?

- ☐ A. Это упрощает поиск файлов, которые были атакованы
- ☐ B. Это обеспечивает персональную ответственность
- ☐ C. Это требуется для выявления DoS-атак
- ☐ D. Это помогает при выполнении корректирующих мер

20. Какая из указанных ниже защитных мер требует разделять обязанности людей, работающих совместно для выполнения критичной задачи?

- ☐ A. Принцип наименьших привилегий
- ☐ B. Скрытие данных
- ☐ C. Двойной контроль
- ☐ D. Административные меры

21. Что из перечисленного ниже не является задачей по управлению использованием носителей информации?

- ☐ A. Компрессия и декомпрессия хранимой информации
- ☐ B. Стирание данных по окончании срока их хранения
- ☐ C. Хранение резервной информации в защищенном месте
- ☐ D. Контроль доступа к носителям информации и журналирование действий с ними

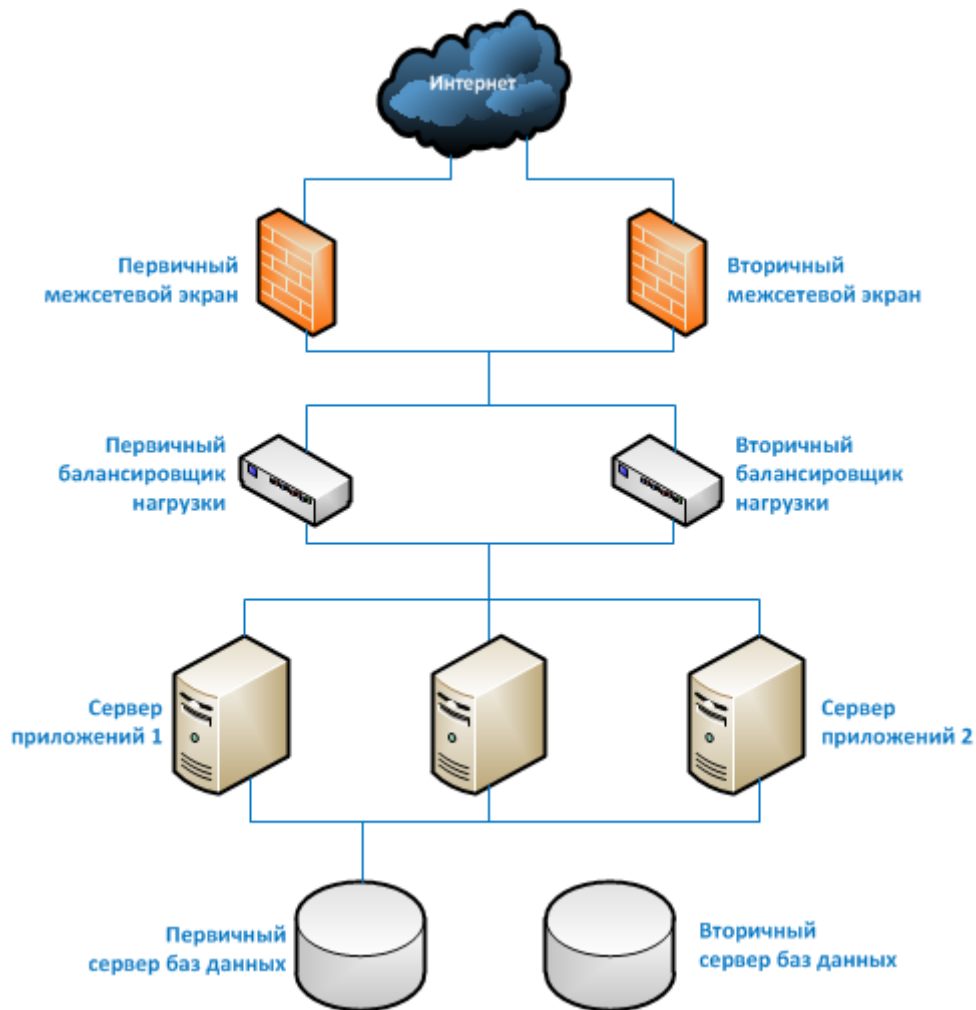
22. Каким образом использование уровней отсечения (clipping levels) помогает отслеживать нарушения?

- ☐ A. Это базовый уровень обычных ошибок пользователей, любые превышения установленного порога таких ошибок должны фиксироваться и анализироваться, чтобы понять, почему это произошло
- ☐ B. Это позволяет администратору видеть снижение количества нарушений по отдельным типам ошибок после внесения каких-либо изменений
- ☐ C. Это позволяет администратору настроить журналирование событий таким образом, чтобы в журналы записывались только события, имеющие отношение к безопасности
- ☐ D. Это позволяет администратору настроить журналирование событий таким образом, чтобы в журналы записывались только события нарушения прав доступа и DoS-атаки

23. Для каких целей организуется библиотека носителей информации?

- ☐ A. Архивное хранение
- ☐ B. Анализ уровней отсечения (clipping levels)
- ☐ C. Защита информационных ресурсов
- ☐ D. Управление изменениями

24. Что иллюстрирует следующее изображение?



- ☐ A. Иерархическое управление носителями (HSM)
- ☐ B. Сеть хранения данных (SAN)
- ☐ C. Обеспечение избыточности сети
- ☐ D. Единая точка отказа

25. Какая из перечисленных ниже защитных мер не позволит сотруднику Департамента эксплуатации получить доступ к данным, на использование которых у него нет полномочий (за исключением сговора с сотрудником другого подразделения)?

- ☐ A. Ограничение локального доступа персонала Департамента эксплуатации
- ☐ B. Ведение журналирования событий
- ☐ C. Выполнение ротации обязанностей
- ☐ D. Ограничение контроля действий руководства компании