Виды информационных угроз и методы борьбы

Летняя Школа Parallels, 2011

Коротаев А. Е.

Здравствуйте

Введение

- Безопасности никогда не бывает мало: степень надёжности системы определяется надёжностью самого незащищённого компонента (мультфильм студии Pilot – Golden Gate).
- Невозможно писать защищённые приложение, не имея представляя о том, какие угрозы нас подстерегают. Поэтому мы говорим о безопасности.
- Невозможно за одну лекцию выучить всё про информационную безопасность! Моя цель: научить вас задумываться о безопасности пользователей ваших приложений.

Виды угроз

- **Q:** Кто может назвать основные виды угроз?

- Модель **STRIDE** включает 6 видов угроз:
  - Подмена личности (**S**poofing identity)
  - Фальсификация данных (**T**ampering with data)
  - Отказ от авторства (**R**epudiation)
  - Раскрытие информации (**I**nformation disclosure)
  - Отказ в обслуживании (**D**enial of service)
  - Повышение привилегий (**E**levation of privilege)

Подмена личности

Что такое подмена личности

- Одна из сторон представляется не тем, кем она является на самом деле:
  - злоумышленник указывает неверное имя пользователя (аналогия: стук в дверь);
  - фальсифицированный сервер отвечает на запросы клиента (отравление кэша DNS, атака на маршрутизатор, Man in the Middle)
  - более изощрённые методы подмены субъекта
- Атака на rsh-сервер подменой TCP-субъекта (атака Морриса старшего)
  - Успешно применён К. Митником для взлома системы Тсутому Шимомуры
  - http://bugtraq.ru/library/books/attack1/chapter4/c45.html?k=9

Аутентификация

- Система должна всегда устанавливать подлинную личность оппонента
- **Аутентификация** - установление личности по некоторым индивидуализирующим признакам

Индивидуализирующие признаки

- **Q:** Давайте вспомним примеры таких признаков в физическом мире.
- Физический мир – естественные признаки:
  - внешний вид (фотография)
  - отпечатки пальцев
  - снимок радужной оболочки глаза
  - …
- Информационные системы – алгоритмы аутентификации на базе секрета, обладать которым может только данный индивид:
  - Пароль
  - Цифровой сертификат **PGP** или **X.509** (*может быть представлен в физическом исполнении*)

Выбор метода аутентификации

- Важно выбрать надёжный механизм аутентификации.

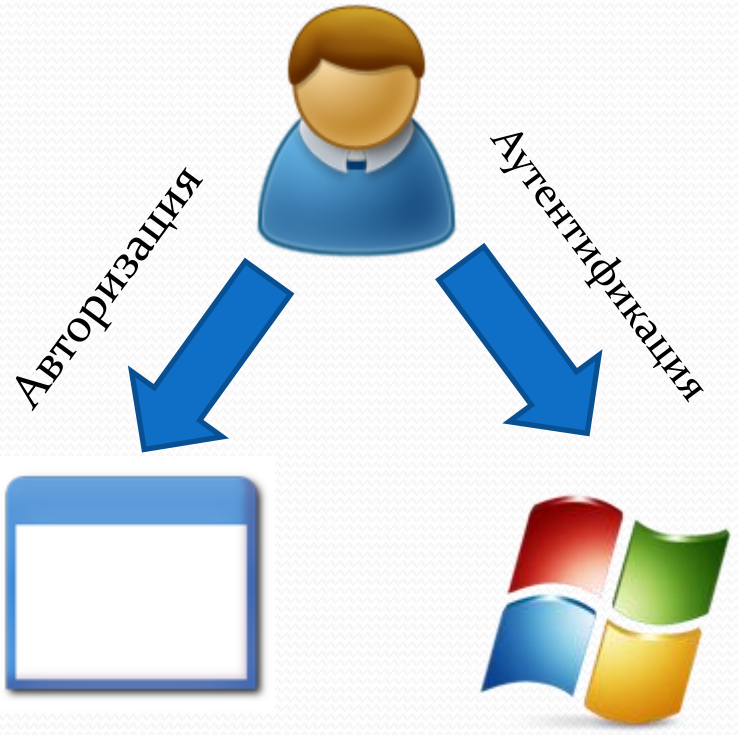
- **Запомните!** Любая кустарная аутентификация, написанная вами, заведомо хуже любой самой плохой, но реализуемой ОС.

  Для того, чтобы довести **NTLM** до ума Microsoft понадобилось более 10 лет и 4 релиза серверных ОС. Вы считаете себя умнее? - “Не верю” © Станиславский

- Воспользуйтесь механизмами аутентификации, предоставляемыми ОС.

Аутентификация средствами ОС

В большинстве случаев нет необходимости вести базу пользователей и самостоятельно аутентифицировать их – предоставьте это ОС. Сопоставьте ресурсы вашего приложения с пользователями ОС и возложите на неё функции проверки подлинности пользователей. После удачной аутентификации пользователя в ОС, вашему приложению останется всего лишь принять решение о том, доступ к каким ресурсам предоставить.



Аутентификация и ОС Windows

- Большинство технологий удалённого (и локального) доступа, реализованных в ОС **Windows**, предоставляют функции аутентификации, интегрированные с ОС и / или доменом **Active Directory**

- Single sign-on
  - В большинстве случаев явная аутентификация вообще не требуется. Каждый пользователь уже один раз прошёл через это процедуру: при входе в ОС. Грамотно написанное приложение воспользуется учётной записью интерактивного пользователя, для автоматического входа во все необходимые сервисы.

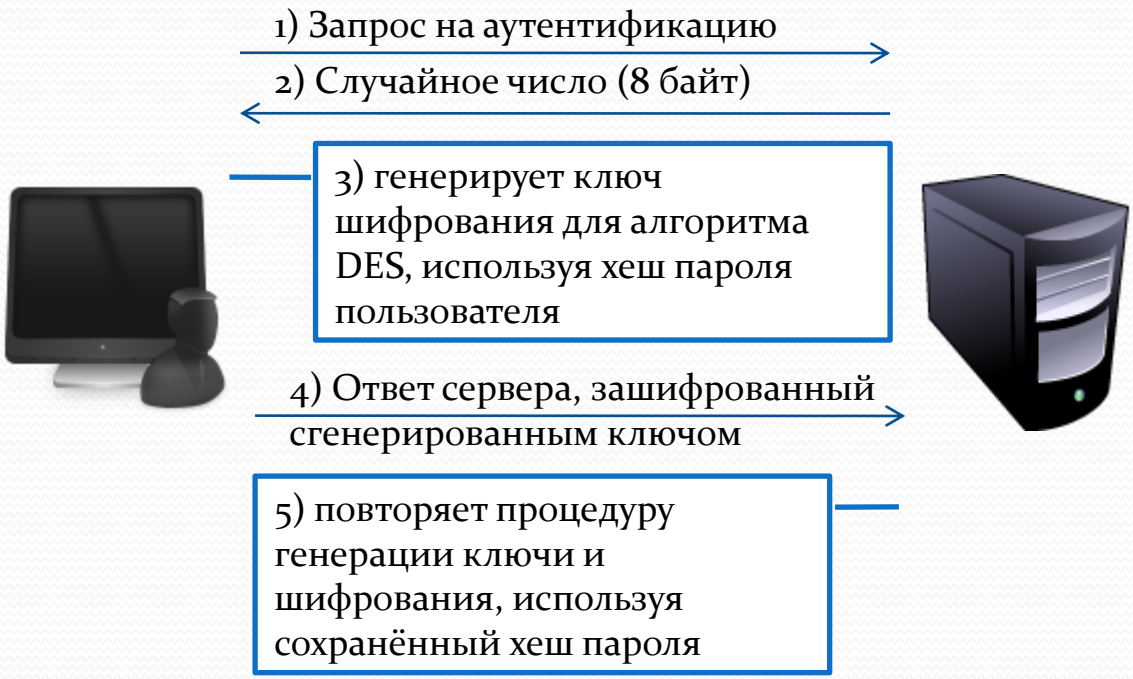
Ошибки кустарной аутентификации

- Иногда (очень редко) возникает необходимость реализации собственного протокола аутентификации (вспомним Windows 95)
- Типичные ошибки реализации собственного протокола аутентификации:
  - пользователь аутентифицируется по паролю, образец которого сохранён в вашей базе
  - для выполнения аутентификации пароль передаётся на сервер
- **Запомните!** Никогда не храните и не передавайте пароли в открытом виде, даже по защищённым каналам!

NTLM – аутентификация без передачи и хранения пароля

Рассмотрим на примере протокола **NTLM**, как может быть устроен надёжный протокол аутентификации

http://mrnone.blogspot.com/2011/05/blog-post.html



Минус:

- можно перехватить ответ клиента и, взломав ключ, получить хеш, поэтому в последствии был заменён на протокол **NTLM v2**.

Плюсы:

- пароль не хранится на сервере в открытом виде
- для аутентификации не нужно передавать ни пароль ни его хеш в открытом виде

Взаимная аутентификация

- Иногда бывает нужно аутентифицировать не только клиента, но и сервер
- Протоколы, позволяющие проверить подлинность сервера:
  - **Kerberos** – сложный промышленный протокол, позволяющий аутентифицировать и клиента, и сервер (опционально). Используется по умолчанию ОС **Windows** для аутентификации в домене **Active Directory**.
  - **SSL/TLS** – защищённый протокол передачи данных, позволяющий проверить подлинность сторон с помощью сертификатов X.509 (*традиционно используется для проверки подлинности сервера, но может быть использован и для контроля подлинности клиента*)

Цифровой сертификат и PKI

- Цифровой сертификат – попытка переноса в цифровой мира такого атрибута, как удостоверение личности
- PKI – **P**ublic **K**ey **I**nfrastructure
- Вся инфраструктура цифровых сертификатов основана на доверии к центру выдачи сертификатов:
  - Локальный центр в пределах сети предприятия
  - Глобальный центр (comodo.com, versign.com)
- Считается, что если вы доверяете этому центру, то вы доверяете подлинности любого сертификата, выданного этим центром.

Сертификат – паспорт цифрового мира

Паспорт

- УВД Центрального р-на г. Н
- Печать УВД
- ФИО: И. И. Иванович
- Фотография и подпись

Сертификат

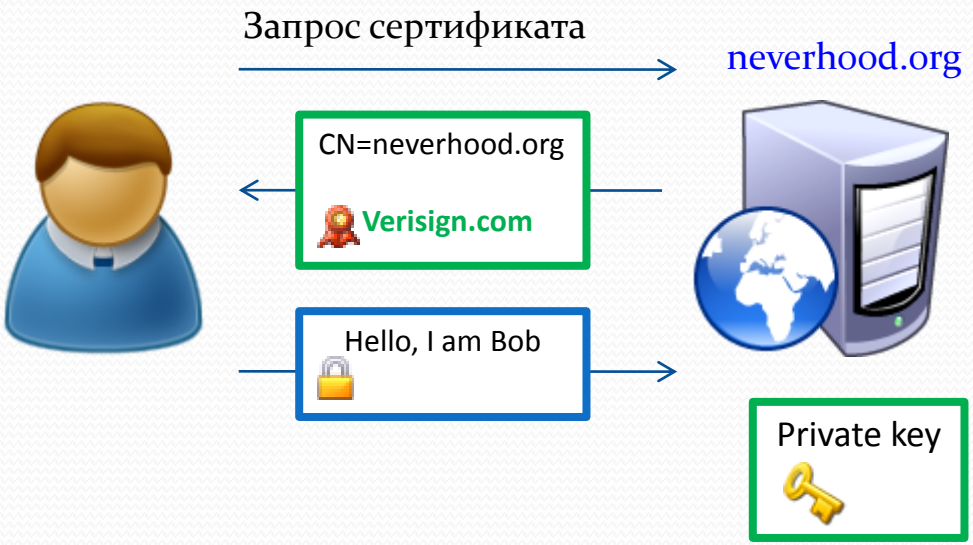
- Verisign.com
- Цифровая подпись CA
- Имя домена: neverhood.org
- Пара ключей RSA

Ассиметричное шифрование в PKI

- Ассиметричные криптосистемы (RSA, DSA, эллиптические кривые):
  - 2 ключа – *публичный* и *приватный*
  - сообщение зашифрованное приватным ключом можно расшифровать только публичным и наоборот
- Цифровые сертификаты:
  - 2 части сертификата – публичная и приватный ключ
  - публичный сертификат предоставляется по требованию, приватный ключ держится в секрете
  - ключ из публичного сертификата используется клиентом для шифрования сообщения – это гарантирует, что только истинный владелец сертификата, обладающий приватным ключом, сможет его прочитать

Схема работы PKI

Клиент запрашивает публичный сертификат сервера. Полученный сертификат должен быть выписан центром сертификатов, которому доверяет клиент, на имя того сервера, к которому он обратился. В этом случае можно воспользоваться ключом из сертификата для шифрования сообщений. Если сервер является тем, за кого себя выдаёт, то он сможет прочитать сообщения, используя свой приватный ключ. Если сервер фальшивый, скопировавший и вернувший настоящий публичный сертификат, то он не сможет прочитать сообщение, поскольку не владеет приватным ключом.



Раскрытие информации

Раскрытие информации

- Доступ к информации получает не тот, кому она предназначалась.

- **Q:** Какой самый верный способ сохранить секрет?
- **A:** Не знать его! Что знают двое – знает и свинья.
- Принцип отказа от хранения секретов:
  - Если есть возможность отказаться от хранения секрета – откажитесь.
- Примеры:
  - Хранение номеров кредитных карточек – зло. Пользователь всегда может ввести номер по запросу.
  - Незачем хранить пароль, достаточно хеша.

Передача секрета

- Старайтесь избегать передачи секрета по сети – обработайте его не стороне клиента. Пример (**Q:** Кто вспомнит?):
  - аутентификация без передачи пароля
- Если возникает необходимость в передаче секрета, используйте защищённые (шифрованные) протоколы:
  - SSL/TLS
  - DCOM
  - IPSec …

Ключ – это тоже секрет!

- При передачи секрета по зашифрованному каналу, возникает ещё один секрет, который нужно надёжно передать – ключ шифрования канала.
- Способы надёжной передачи ключа по открытому каналу:
  - Алгоритм Диффи-Хельмана
  - PKI и ассиметричные криптосистемы

Алгоритм Диффи-Хельмана

Первая схема безопасного обмена ключами по открытому каналу (1976 год).

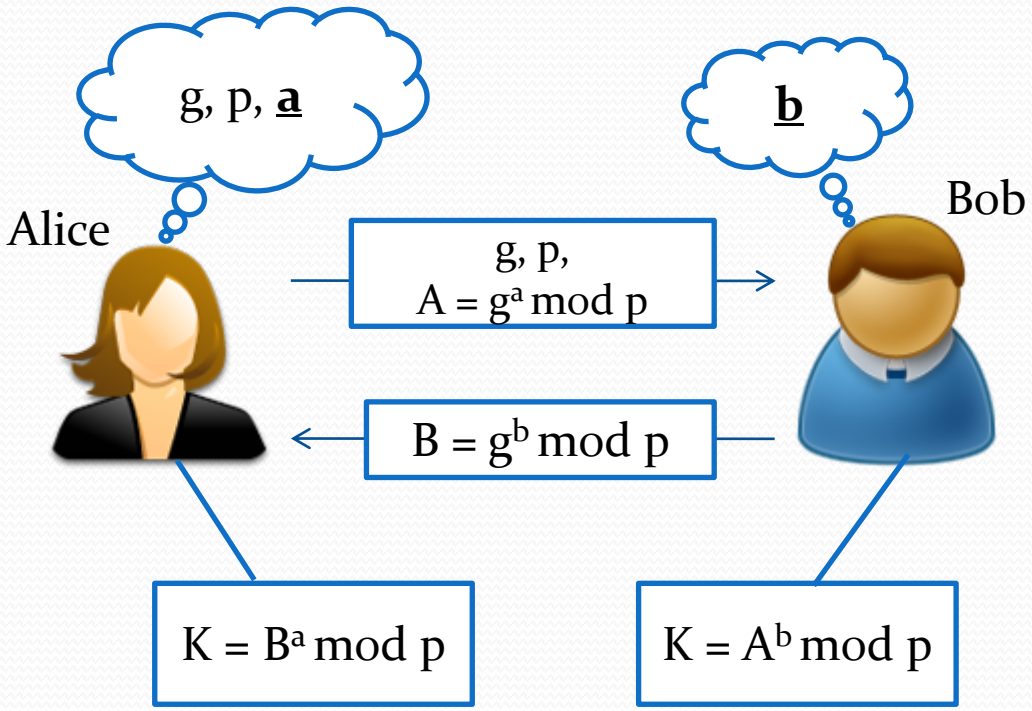
p – достаточно большое простое число (порядка 10^{300});

g – первообразный корень по модулю p;

a и **b** – достаточно большие натуральные числа (порядка 10^{100}), сохраняющиеся в секрете

K – общий секретный ключ, вычисленный по открытым ключам.

Человек, прослушивающий, но не модифицирующий канал, не в состоянии вычислить $K = g^{ab} mod\ p$ по перехваченным $g^a \bmod p$ и $g^b \bmod p$ – проблема дискретного логарифмирования.



$$K = B^a \bmod p = (g^b \bmod p)^a \bmod p = g^{ab} \bmod p = (g^a \bmod p)^b \bmod p = A^b \bmod p$$

Проблема алгоритма Диффи-Хельмана

- Алгоритм не спасает от атаки Man in the Middle:
  - Mallory вклинивается между Alice и Bob'ом и, модифицируя сообщения, передаёт свои версии публичных ключей
  - Возникает 2 "*секретных*" ключа, известных Mallory: K_{AM} – для канала между Alice и Mallory и K_{MB} – для канала между Mallory и Bob'ом
  - Канал компрометирован: Mallory получает сообщения от Alice, расшифровывает ключом K_{AM} и передаёт Bob'у, зашифровав его ключом K_{MB}.
- PKI и асимметричные криптосистемы решили эту проблему.

RSA – пример асимметричного шифра

- Асимметричные криптосистемы или криптосистемы с открытым ключом
- В основе необратимые функции:
  - y = f(x) вычисляется легок для известного x
  - нет эффективно-вычислимой функции g(y) = x, которую можно вычислить за разумное время на современном оборудовании
- Рональд Райвест (**R**ivest), Ади Шамир (**S**hamir), Леонард Адлеман (**A**delman) – 1977.
- Первый алгоритм асимметричного шифрования
- В основе задача умножения и разложения на простые сомножители, являющаяся вычислительно однонаправленной.

Генерация ключей RSA

- Выбираются **p** и **q** – различные простые числа одного размера (например 1024 бит)
- **n = pq** его размер в 2 раза больше размера p и q (этот размер и есть размер ключа RSA)
- $\phi(n) = (p - 1)(q - 1)$ – функция Эйлера
- Выбирается **e** – открытая экспонента: $1 < e < \phi(n)$ и взаимно простое с $\phi(n)$
- Вычисляется секретная экспонента

  $d = e^{-1} \bmod \phi(n)$

  число мультипликативно обратное к **e** по модулю $\phi(n)$. Для вычисления обычно используется алгоритм Эвклида.
- **(e, n)** – открытый ключ RSA
- **(d, n)** – закрытый ключ RSA

Шифрование алгоритмом RSA

- Alice передаёт свой открытый ключ **(e, n)** Bob'y
- Bob берёт текст **m**, вычисляет шифротекст

  $\mathbf{c} = m^e \pmod{n}$

  и передаёт Alice по открытому каналу
- Alice берёт полученный шифротекст c и вычисляет исходный текст, используя приватный ключ:

  $m = c^d \pmod{n}$.
- Уравнения для вычисления c и m являются взаимно обратными преобразованиями над множеством простых чисел.
- Восстановить **m** за разумное время по имеющимся **c**, **e** и **n** при достаточно большом **n** невозможно.

RSA и PKI

- RSA само по себе не решает проблему безопасной передачи ключа, для этой цели используется PKI
- Публичный ключ RSA публикуется в составе публичной части сертификата. Приватный ключ RSA является приватным ключом сертификата
- Bob считает, что ключ Alice корректен, если он получен в составе сертификата, выданного центром сертификации, которому он доверяет.

Защиты канал алгоритмом RSA

- Шифрование алгоритмом RSA ресурсоёмко
- Передаваемые данные шифруются симметричным шифром:
  - **AES** (стандарт правительства США)
  - **Camellia** (сертифицирован в Японии)
  - **ГОСТ 28147-89** (стандарт РФ и СНГ)
- RSA используется только на стадии установления соединения для аутентификации сторон и передачи симметричного ключа в зашифрованном виде

Надёжное хранение

- Ограничивайте доступа к данным, даже если они зашифрованы
  - Авторизация – процесс определения списка ресурсов, доступных аутентифицированному клиенту
  - Используйте механизмы разграничения доступа, предоставляемые ОС
- ACL (Access Control List) в Windows позволяют
  - Настроить стандартные права доступа для любых системных объектов
  - Настроить собственные права доступа для любых пользовательских объектов
  - В сочетании с системной аутентификацией предоставляют мощный механизм защиты
- Научитесь использовать криптографию
  - Используйте алгоритмы, проверенные временем, не изобретайте собственные *“надёжные”* криптографические функции!
  - Любой алгоритм основанный на секрете самого алгоритма – это фикция (пример **IMail**); надёжный алгоритм шифрования основан на секрете ключа
  - Применяйте *“соль”*
  - Позаботьтесь о надёжном генераторе случайных чисел

Надёжное хранение (ключи)

- Научитесь работать с ключам
  - Внимательно подходите к выбору длины ключа (бесполезно шифровать 112-битный 3DES ключ 512 битным RSA ключом: 2-ой взламывается быстрее 1-ого)
  - Ключ, сохранённый вместе с данными, даже в секретном месте – это полное отсутствие защиты (типичная ошибка: ключ “*зашитый*” в ПО)
  - Если ключ используется для шифрования большого объёма данных, он может использоваться в течении короткого промежутка времени – кратковременный ключ (защита сессии)
  - Если ключ используется в течении длительного промежутка времени (шифрование и генерация сессионных ключей) или для защиты хранимых данных, он может использовать для шифрования лишь небольшого объёма информации – долговременный ключ
  - Никогда не смешивайте функции долговременных и кратковременных ключей

Фальсификация данных

Угроза известная с детства

- Фальсификация может нанести не меньший, а иногда и больший урон, нежели раскрытие данных.
- Этот вид угрозы знаком всем практически с детства.
- **Q:** Кто вспомнит самый известный пример?

  …Допьяна гонца поят
  И в суму его пустую
  Суют грамоту другую…

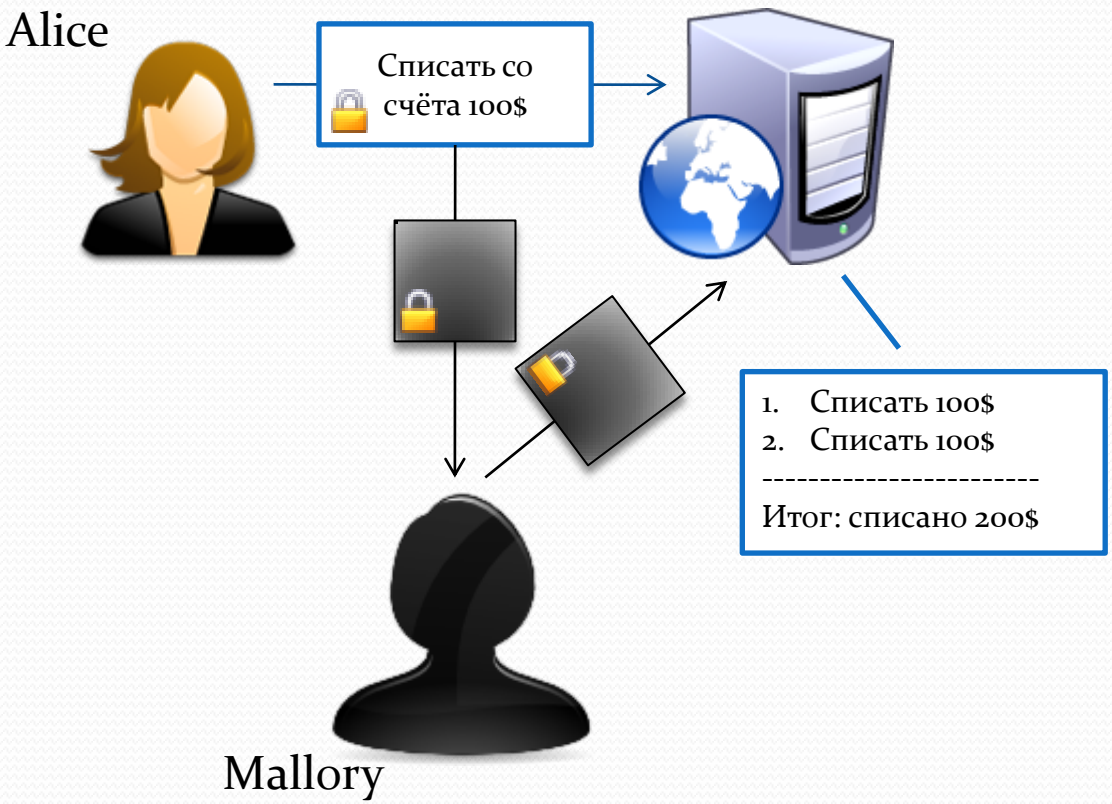
  А.С. Пушкин, Сказка о царе Салтане.

- **Q:** Пример, который уже прозвучал в этой лекции
  - Подмена ключей в алгоритме Диффи-Хельмана

Replay атака

Replay атака – это фальсификация данных, связанная с перехватом и повторным использованием устаревшей информации взамен актуальной.

Особенность: атакующему не нужно взламывать криптозащиту, чтобы нанести существенный урон. Достаточно знать особенности работы протокола, его слабые стороны.



Методы защиты от фальсификаций

- Авторизация – защита от несанкционированной модификации хранимых данных и системных объектов
- Контрольные суммы – защита от случайных модификаций в следствии ошибок передачи (**не является методом защиты от злонамеренных модификаций!**)
- Временные метки и метки сессии – защита от *replay* атак
- Цифровые подписи, MAC коды, защищённые протоколы – защита от прямых злонамеренных изменений передаваемых данных

Хеш функция

- **Криптографическая** хеш-функция – однонаправленная функция, преобразующая исходное сообщение в небольшое по размеру значение – порядка 128 бит (512 для SHA-512), стойкая к коллизиям.
- Хеш не несёт никакой информации о данных, но позволяет их однозначно идентифицировать.
- Примеры: **MD5**, **SHA-2**, **ГОСТ Р 34.11-94**.
- Сам по себе хеш не гарантирует аутентичность сообщения, поскольку также может быть изменён злоумышленником

MAC и Цифровая подпись

- MAC – Message Authentication Code
  - Результат применения хеш-функции к сообщению, дополненному секретом, известным только сторонам диалога
  - Для проверки сообщения необходимо повторить операцию получения MAC и сравнить с полученным значением
  - Аутентичность гарантирует секрет, известный только 2-ум сторонам
- Цифровая подпись
  - PKI и сертификаты
  - Результат шифрования хеша сообщения приватным ключом отправителя
  - Для проверки сообщения необходимо сравнить вычисленный хеш с результатом расшифровки, выполненной публичным ключом из сертификата отправителя
  - Аутентичность гарантируется приватностью ключа отправителя и доверием к центру выдачи сертификата

Отказ от авторства

Примеры отказа от авторства

- Сотрудник КБ смог тайно переслать конкурирующей фирме чертежи нового двигателя
- Низко квалификацией кассир списал со счёта клиента $1000 вместо 1000 рублей и свалил всё на своего коллегу
- Злоумышленник взломал систему извне и остался безнаказанным, поскольку смогли обнаружить лишь деструктивные последствия взлома, но не смогли отследить его действия

Пугает не наказание, а его неотвратимость

- Запрет на анонимные операции
  - Явная аутентификация каждого пользователя с помощью уникального логина для совершения любых операций в системе
- Аудит
  - Упрощает расследование инцидента и поиск виноватого, в том числе и при внешних воздействиях (Митник попался потому, что на атакованном компьютере вёлся подробный аудит всей сетевой активности)
  - Вместе с обязательной аутентификацией позволяет однозначно идентифицировать пользователя, совершившего операцию
- Цифровая подпись
  - Добавляет юридическую ответственность, упрощает доказательство в суде

Отказ в обслуживании

Устойчивый сервер – залог успеха

- Запрещайте анонимные операции или лимитируйте их количество
  - Анонимные соединения – потенциальный источник DoS атак
- Позаботьтесь о раннем детектирование некорректных соединений на уровне протокола
  - Прерывайте некорректное соединение как можно быстрее, не дожидаясь завершения бесконечной передачи
  - В этом плане протокол на базе *“чистого”* XML плох. **Q:** Почему?
- Не забывайте про предел мощностей
  - Количество одновременно обрабатываемых запросов должно быть ограничено
- Помните, что принятие запроса и его обработка – разные вещи
  - Принятый корректный запрос может быть помещён в очередь ожидания обработки

Повышение привилегий

Принцип минимальных привилегий

- Печально известная уязвимость службы DCOM / RPC
  - сервис, выполнявшийся с привилегиями системы, имел уязвимость типа “Удалённое исполнение кода”
- Всегда используйте привилегии минимально достаточные для выполнения задачи
  - В своём коде запрашивайте только тот уровень доступа к объектам, который необходим для выполнения операции
  - Приложение должно корректно работать под любой непривилегированной учётной записью
  - Операции, требующие повышенных привилегий, выносите в сервисы, для запуска которых используйте специальные учётные записи, обладающие только необходимыми привилегиями и уровнем доступа
  - Тщательно планируйте ACL объектов в соответствии с этими принципами

Подведём итог

Угроза	Методы защиты
Подмена личности	•Надёжный механизм аутентификации •Защита секретов •Отказ от хранения секретов
Фальсификация данных	•Надёжный механизм авторизации •Использование хешей •MAC-коды •Цифровые подписи •Протоколы с защитой от несанкционированного доступа
Отказ от авторства	•Цифровые подписи •Аудит
Раскрытие информации	•Надёжный механизм авторизации •Протоколы с защитой от несанкционированного доступа •Защита секретов •Отказ от хранения секретов •Шифрование
Отказ в обслуживании	•Надёжный механизм аутентификации •Надёжный механизм авторизации •Фильтрация запросов •Управление числом входящих запросов
Повышение привилегий	•Принцип минимальных привилегий

Вопросы

Что почитать

- М. Ховард, Д. Лебланк. Защищённый код. Microsoft Press / Русская редакция, 2005 г.
- М. Ховард, Д. Лебланк. Защищённый код для Windows Vista. Microsoft Press / Русская редакция, 2008 г.
- М. Ховард, Д. Лебланк, Дж. Вьега. 24 смертных греха компьютерной безопасности. Питер, **2010 г.**
- Б. Шнайер. Прикладная криптография. Триумф, 2002 г.
- Н. Фергюсон, Б. Шнайер. Практическая криптография. Вильямс, 2004 г.
- Дж. Рихтер, Дж. Кларк. Программирование серверных приложений для Windows 2000. Питер, 2001 г.
- Й. Снейдер. Эффективное программирование TCP/IP. Питер, 2002 г.

Что почитать в online

- http://bugtraq.ru
- http://www.securitylab.ru
- http://blogs.technet.com/b/markrussinovich
- http://www.schneier.com