Реферат на тему:

Сетевой протокол для безопасной передачи данных SSH.

Кевролетин В.В. гр. С8403а

2011г

SSH (Secure Shell) — сетевой протокол прикладного уровня, разработанный для безопасного удаленного доступа через открытые каналы связи. Безопасность обеспечивается за счет взаимной аутентификацию участников соединения(сервера и клиента), шифрования передаваемых данных и применения средств поддержания их целостности.

Помимо своей основной задачи ssh предоставляет так же вспомогательный фуккцию агрегированной(передача в нескольких потоков) передачи трафика других протоколов.

Учитывая высокие требования к безопасности и производительности SSH использует шифрование с открытым ключём во время установления соединеня(для аутентефикации) и блочные или потоковые шифры на этапе обмена данными.

Протокол SSH разрабатывается и стандартизируется Инженерным советом Интернет (Internet Engineering Task Force, IETF). Поэтому является открытым стандартом, как и все стандарты, разрабатываемые данной организацией.

Протоколы SSH и их функции

SSH разделён на 3 основные части[1] устроенные в виде стека протоколов и работающие обычно, поверх сетей TCP/IP

1. Протокол аутентификации пользователя [2]
2. Протокол соединения - установление сессии, удаленное исполнение команд, передача данных других протоколов. [3]
3. Транспортный протокол - обеспечивает шифрование, аутентефикацию сервера и поддержание целостности пересылаемых данных. Опционально может реализовывать компрессию.[4]

Описание протокола

Установление сессии обмена данными между клиентом и сервером проходит в несколько этапов.

1.Обмен идентификационными сообщениями

После установления сессии, инициированной клиентом, участники клиент и сервер обмениваются идентификационными сообщениями . Пока что пакеты идут открытыми.- цель обмена сообщениями выяснить версию поддерживаемого протокола.

2. Согласование алгоритмов

В протокол SSH заложены возможность использования не определённых стандартом алгоритмов, там что перед установлением сессии согласовываются:

1. алгоритм обмена ключами
2. алгоритм создания публичного ключа хоста сервера
3. алгоритмы шифрования для сервера и клиента
4. MAC алгоритмы сервера и клиента (объяснение ниже)
5. алгоритмы сжатия для сервераи клиента
6. национальный язык, используемый сервером и клиентом в сообщениях

[5]

3. Обмен ключами

Обмен ключами влияет на то, как будут генерироваться ключи одноразовых сессий и как будет проходить аутентефикация сервера. Обмен проходит по алгоритму утвержденному на предыдущем шаге.

Стандартом определён к реализации только один метод: алгоритм Диффи — Хеллмана

Этот алгоритм позволяет 2м сторонам обменяться общим секретом через прослушиваемый, но защищенный от подмены данных канал связи. В основе алгоритма лежит проблема дискретного логарифмирования. Краткое описание алгоритма:

1. Генерируются 2 случайных чила, известные обеим сторонам: p — больше простое число, g — любое число < p.
2. Клиент генерирует случайное число x, вычисляет `e = g^x (mod p)` и отсылает `e` серверу.
3. Сервер получает `e` и вычисляет `K = e^y (mod p)`
4. Сервер генерирует случайное число y, вычисляет `f = g^y (mod p)`и отсылает `f` клиенту.
5. Клиент получает `f` и вычисляет `K = f^x (mod p)`

Нетрудно проверить подстановкой, что клиент и сервер, действительно получили одно и то же число K. Злоумышленники, перехватившему только числа e и f необходимо решить сложную задачу дискретного логарифмирования, чтобы найти x или y.

4. Обмен данными

Хотя, стандарт позволяет полностью отказаться от защиты при обмене информации, рекомендовано использовать шифрованием и механизмом поддержания целостности.

Стандарт рекомендует к реализации множество алгортмов шифрования, среди которыех

- AES (Rijndael) в режиме SDCTR с 128-битным ключем
- AES с 192-битным ключем
- AES с 256-битным ключем
- Тройной 3DES в режиме SDCTR
- Blowfish в режиме SDCTR
- Twofish в режиме SDCTR

Режим SDCTR (stateful-decryption counter mode) превращает блочный шифр в поточный. Последовательность целых чисел(начиная с чила, определяемого на этапе обмена ключами) шифруется блочным шифром. Полученный шифротекст рассматривается как поток бит, который побитово складывается по модулю 2 с потоком входных данных. Таким образом режим SDCTR ничто иное, как частный случай известного режима CTR(counter).

Метод шифрования 3DES — метод, согласно которому к каждому блоку открытого текста 3 раза последовательно применятся алгоритм DES: блок шифруется с одним ключём, расшифровывается с другим и снова шифруется с третьим ключём. Длина ключа DES 8 байт, так что длина ключа для 3DES 24байта.

Для обеспечения целостности данных в каждый пакет помещается специальное сообщение(MAC), вычисленное как функция от закрытого ключа, порядкового номера покета и содержимого пакета.

Стандарт рекомендует несколько алгоритмов, среди которых обязателен к реализации:

- HMAC-SHA1

Алгоритмы публичных ключей

- Raw DSS Key (обязателен к реализации)
- Raw RSA Key
- OpenPGP certificates (RSA key)
- OpenPGP certificates (DSS key)

Атаки

Опишем распространённые атаки, которые могут быть придприняты на ssh и уровень защищенности протокола от них.

1 Раскрытие шифров

В качестве шифра спецификацией рекомендуетсяч ряд надёжных по современным меркам алгоритмов. Тем не менее все шифры лишь рекомендованы и возможно использование своих шифров. Так что безопасность передачи данных зависит от выбора алгоритма шифрования во время установления сессии. Стоит отметить, что протокол рекомендует менять ключ после передачи 1 Гигабайта данных[6], что должно усложнить вскрытие ключа шифрования.

2 Перехват с участием человека (Man-in-the-middle)

Во время передачи данных атака неэффективна в следствии использования шифрования и кода аутентификации сообщения (MAC).

Протокол уязвим во время обмена ключами, так как не включает в себя предположений о распределение открыты хостов, и не описывает способ обмена ключами. Т.е. протокол не позволяет проверить подлинность сервиса.

3. Использование перехваченных данных (Replay)

Код аутентификации сообщения MAC генерируется на основе номера сообщения, так что

невозможно 2 раза использовать одно и то же сообщение. Кроме того транспортный протокол хранит в сообщениях уникальный идентефикатор сессии, генерируемый на основе случайных алгоритмов

Список источников

1: IETF, SSH Protocol Architecture , http://tools.ietf.org/html/rfc4251
2: IETF, The Secure Shell (SSH) Authentication Protocol , http://tools.ietf.org/html/rfc4252
3: IETF, The Secure Shell (SSH) Connection Protocol , http://tools.ietf.org/html/rfc4254
4: IETF, The Secure Shell (SSH) Transport Layer Protocol , http://tools.ietf.org/html/rfc4253
5: Шошмина И.В., SecureShell 2.0 , http://ns.csa.ru/CSA/tutor/SSHWEB2.htm/
6: RFC 2.0 — Русские Переводы RFC, RFC 2.0 — Русские Переводы RFC Архитектура протокола SSH ,
7. http://rfc2.ru/4251.rfc
8. http://en.wikipedia.org/wiki/SSH
9. http://www.opennet.ru/base/sec/ssh_intro.txt.html