Протоколы, сервисы и услуги
в Интернет и IP-сетях

Тема № 10
Протоколы удаленного управления Telnet и SSH

доц. каф. СС и ПД, к.т.н. С. С. Владимиров

2017 г.

Протокол/служба Telnet

TELNET (TErminaL NETwork)

Служба прикладного уровня модели OSI для реализации текстового интерфейса по сети (в современной форме — при помощи транспорта TCP). Название «telnet» имеют также некоторые утилиты, реализующие клиентскую часть протокола. Современный стандарт протокола описан в RFC 854.

Назначение протокола TELNET в предоставлении достаточно общего, двунаправленного, восьмибитного байт-ориентированного средства связи. Его основная задача заключается в том, чтобы позволить терминальным устройствам и терминальным процессам взаимодействовать друг с другом. Протокол может быть использован для связи вида терминал-терминал («связывание») или для связи процесс-процесс («распределенные вычисления»).

В сессии Telnet выделяют клиентскую и серверную стороны, однако сам протокол на самом деле полностью симметричен. После установления транспортного соединения оба его конца играют роль «сетевых виртуальных терминалов» (Network Virtual Terminal, NVT), обменивающихся двумя типами данных:

- прикладные данные;
- команды протокола Telnet.

Хотя Telnet-сессии, выполняющейся по TCP, свойственен полный дуплекс, NVT должен рассматриваться как полудуплексное устройство, работающее по умолчанию в буферизированном строковом режиме.

Прикладные данные проходят через протокол без изменений, то есть на выходе второго виртуального терминала мы видим именно то, что было введено на вход первого. С точки зрения протокола данные представляют просто последовательность байтов (октетов), по умолчанию принадлежащих набору ASCII, но при включенной опции Binary — любых. Были предложены расширения для идентификации набора символов, но на практике ими не пользуются.

Протокол/служба Telnet

Все значения октетов прикладных данных кроме \377 (десятичное: 255) передаются по транспорту как есть. Октет \377 передаётся последовательностью \377\377 из двух октетов. Это связано с тем, что октет \377 используется на транспортном уровне для кодирования опций.
Протокол предоставляет по умолчанию минимальную функциональность и набор расширяющих её опций. Принцип оговоренных опций требует проводить переговоры при включении каждой из опций. Одна сторона инициирует запрос, а другая сторона может либо принять, либо отвергнуть предложение. Если запрос принимается, то опция немедленно вступает в силу. Опции описаны отдельно от протокола как такового, и их поддержка программным обеспечением произвольна. Клиенту протокола (сетевому терминалу) предписывается отвергать запросы на включение неподдерживаемых и неизвестных опций.
Терминал NVT имеет неопределённую ширину каретки и длину страницы и должен иметь представление всех 95 печатных символов US-ASCII (коды с 32 по 126).
Каждая команда TELNET является многобайтовой последовательностью, начинающейся с кода \377 «Interpret as Command» (IAC) и кода команды. Команды, отвечающие за договоренности по опции, являются трехбайтовыми последовательностями, где третий байт является кодом опции.

Исторически Telnet служил для удалённого доступа к интерфейсу командной строки операционных систем. Впоследствии его стали использовать для прочих текстовых интерфейсов, вплоть до игр MUD и анимированного ASCII-art. Теоретически, даже обе стороны протокола могут являться не только людьми, но и программами.
Иногда клиенты telnet используются для доступа к другим протоколам на основе транспорта TCP. Например, telnet используется в управляющем соединении FTP.
В протоколе не предусмотрено использование ни шифрования, ни проверки подлинности данных. Поэтому он уязвим для любого вида атак, к которым уязвим его транспорт, то есть протокол TCP. Сессия Telnet беззащитна, если только не осуществляется в полностью контролируемой сети или с применением защиты на сетевом уровне (различные реализации виртуальных частных сетей). По причине ненадёжности от Telnet как средства управления операционными системами отказались.

Протокол SSH

SSH (Secure Shell)

Сетевой протокол прикладного уровня, позволяющий производить удалённое управление операционной системой и туннелирование TCP-соединений (например, для передачи файлов). Схож по функциональности с протоколами Telnet и rlogin, но, в отличие от них, шифрует весь трафик, включая и передаваемые пароли. SSH допускает выбор различных алгоритмов шифрования. SSH-клиенты и SSH-серверы доступны для большинства сетевых операционных систем.

SSH позволяет безопасно передавать в незащищённой среде практически любой другой сетевой протокол. Таким образом, можно не только удалённо работать на компьютере через командную оболочку, но и передавать по шифрованному каналу звуковой поток или видео. Также SSH может использовать сжатие передаваемых данных для последующего их шифрования, что удобно, например, для удалённого запуска клиентов X Window System.

Первая версия протокола, SSH-1, была разработана в 1995 году исследователем Тату Улёненом из Технологического университета Хельсинки (Финляндия). SSH-1 был написан для обеспечения большей конфиденциальности, чем протоколы rlogin, telnet и rsh. В 1996 году была разработана более безопасная версия протокола, SSH-2, несовместимая с SSH-1. Протокол приобрел ещё большую популярность, и к 2000 году у него было около двух миллионов пользователей. В настоящее время под термином «SSH» обычно подразумевается именно SSH-2, т. к. первая версия протокола ввиду существенных недостатков сейчас практически не применяется.
В 2006 году протокол был утвержден рабочей группой IETF в качестве Интернет-стандарта. SSH-2 определен в RFC 4250–4256.
В некоторых странах (Франция, Россия, Ирак и Пакистан) требуется специальное разрешение в соответствующих структурах для использования определённых методов шифрования, включая SSH.
Распространены две реализации SSH: частная коммерческая и бесплатная свободная. Свободная реализация называется OpenSSH.

Протокол SSH

Протокол SSH-1, в отличие от протокола telnet, устойчив к атакам прослушивания трафика («снифинг»), но неустойчив к атакам «человек посередине». Протокол SSH-2 также устойчив к атакам путем присоединения посредине (session hijacking), так как невозможно включиться в уже установленную сессию или перехватить её.
Для предотвращения атак «человек посередине» при подключении к хосту, ключ которого ещё не известен клиенту, клиентское ПО показывает пользователю «слепок ключа» (key fingerprint). Рекомендуется тщательно сверять показываемый клиентским ПО «слепок ключа» со слепком ключа сервера, желательно полученным по надёжным каналам связи или лично.
Поддержка SSH реализована во всех UNIX-подобных системах, и на большинстве из них в числе стандартных утилит присутствуют клиент и сервер ssh. Существует множество реализаций SSH-клиентов и для не-UNIX ОС. Большую популярность протокол получил после широкого развития анализаторов трафика и способов нарушения работы локальных сетей, как альтернативное небезопасному протоколу Telnet решение для управления важными узлами.

Для работы по SSH нужен SSH-сервер и SSH-клиент. Сервер прослушивает соединения от клиентских машин и при установлении связи производит аутентификацию, после чего начинает обслуживание клиента. Клиент используется для входа на удалённую машину и выполнения команд.
Для соединения сервер и клиент должны создать пары ключей — открытых и закрытых — и обменяться открытыми ключами. Обычно используется также и пароль.

Протокол SSH

Техническая информация о протоколе

SSH-сервер для работы использует TCP-порт 22. Для аутентификации сервера в SSH используется протокол аутентификации сторон на основе алгоритмов электронно-цифровой подписи RSA или DSA. Для аутентификации клиента также может использоваться ЭЦП RSA или DSA, но допускается также аутентификация при помощи пароля (режим обратной совместимости с Telnet) и IP-адреса хоста (режим обратной совместимости с rlogin). Аутентификация по паролю наиболее распространена; она безопасна, так как пароль передаётся по зашифрованному виртуальному каналу. Аутентификация по ip-адресу небезопасна, эту возможность чаще всего отключают. Для создания общего секрета (сеансового ключа) используется алгоритм Диффи–Хеллмана (DH). Для шифрования передаваемых данных используется симметричное шифрование, алгоритмы AES, Blowfish или 3DES. Целостность передачи данных проверяется с помощью CRC32 в SSH1 или HMAC-SHA1/HMAC-MD5 в SSH2.
Для сжатия шифруемых данных может использоваться алгоритм LempelZiv (LZ77).

Советы по безопасности использования SSH

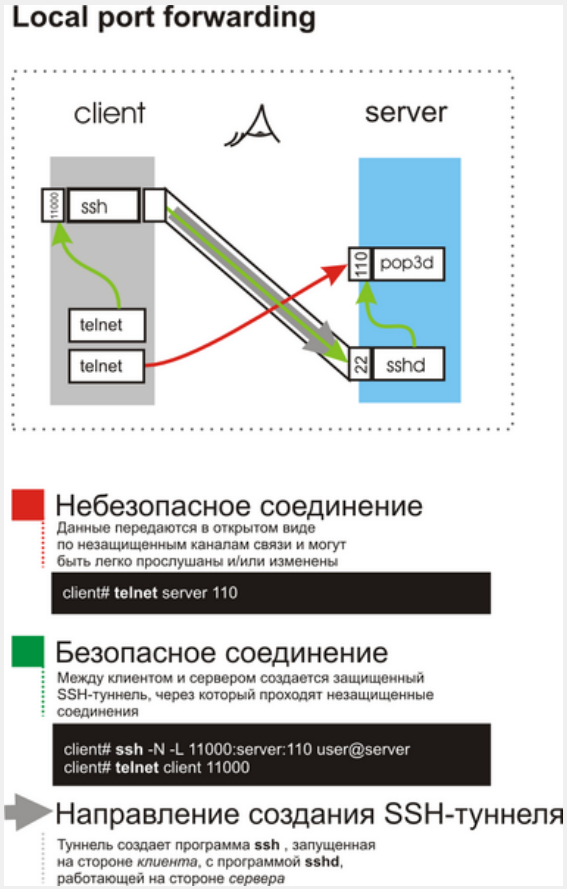
- Запрещение удалённого root-доступа.
- Запрещение подключения с пустым паролем или отключение входа по паролю.
- Выбор нестандартного порта для SSH-сервера.
- Использование длинных SSH2 RSA-ключей (2048 бит и более). Системы шифрования на основе RSA считаются надёжными, если длина ключа не менее 1024 бит.
- Ограничение списка IP-адресов, с которых разрешён доступ.
- Запрещение доступа с некоторых потенциально опасных адресов.
- Регулярный просмотр сообщений об ошибках аутентификации.
- Установка систем обнаружения вторжений (IDS).

SSH-туннелирование

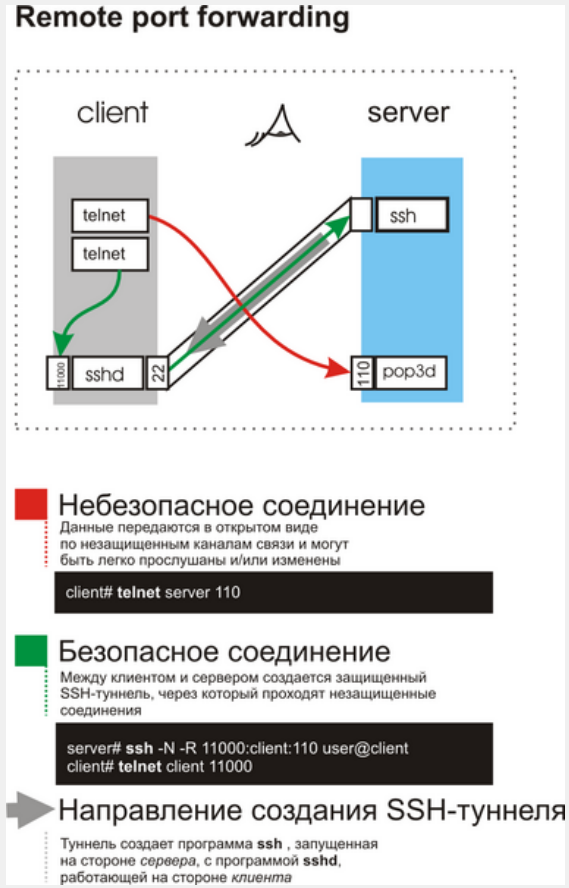
SSH-туннелирование

SSH-туннель — это туннель, создаваемый посредством SSH-соединения и используемый для шифрования туннелированных данных. Используется для того, чтобы обезопасить передачу данных в Интернете (аналогичное назначение имеет IPsec). При пересылке через SSH-туннель незашифрованный трафик любого протокола шифруется на одном конце SSH-соединения и расшифровывается на другом.

Local Port Forwarding



Remote Port Forwarding



Источники

- Материалы с сайта `https://wikipedia.org/`
- Telecommunication technologies — телекоммуникационные технологии / Ю. А. Семенов. URL: `http://book.itep.ru/`