Министерство образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение   
высшего профессионального образования   
«Уральский федеральный университет имени первого Президента России Б.Н. Ельцина»

Институт радиоэлектроники и информационных технологий - РТФ

**Учебно-методическое пособие**

**по теме: «SSH и SFTP. Настройка и организация работы. Как обеспечить безопасность сервиса. Анализ трафика»**

Екатеринбург 2020

**Оглавление**

[Введение 3](#_Toc470484352)

[1. Основы файловой системы ext4 4](#_Toc470484353)

[2. Подготовка к лабораторному практикуму 11](#_Toc470484354)

[3. Лабораторный практикум 13](#_Toc470484355)

[Заключение 16](#_Toc470484356)

[Литература 17](#_Toc470484357)

# Введение

SSH-протокол (от англ. Secure Shell) — криптографический сетевой протокол, предназначенный для удаленного доступа к операционной системе и осуществления безопасного удаленного управления в рамках незащищенной сети (например, через интернет).

SFTP-протокол (от англ. SSH File Transfer Protocol) – сетевой протокол прикладного уровня, предназначенный для передачи файлов и других действий с ними через имеющееся надежное соединение. Протокол был разработан как расширение SSH-2, предназначенное для операций с файлами по защищенному каналу, однако может работать и с другими протоколами, обеспечивающими безопасное соединение сервера с клиентом.

Чаще всего протоколы SSH и SFTP используются для удаленной работы с операционной системой или переноса большого количества файлов. Также SSH может использоваться для удаленной работы по защищенному соединению с различными сервисами провайдера, такими как программное обеспечение, операционные системы и т.д.

В данной работе будут рассмотрены принципы работы этих протоколов, т.к. защитникам информации критически необходимо знать и уметь пользоваться этими протоколами для обеспечения информационной безопасности.

# Теоретическая часть

**Протокол SSH**

SSH — это набор программ, которые позволяют выполнить вход на удалённую машину для выполнения команд на ней. Он предназначен для обеспечения защищённой зашифрованной связи между двумя узлами через незащищённую сеть. Соединения X11, произвольные порты TCP и сокеты домена UNIX также могут быть переадресованы по защищённому каналу. В SSH входят программы, которые дополнительно позволяют передавать файлы по зашифрованному соединению.

SSH несёт в себе различные улучшения безопасности, среди них аутентификация пользователя/хоста, шифрование данных и целостность данных, благодаря чему атаки вроде подслушивания (сниффинга), DNS/IP спуфинга, подделка данных (data forgery) и перехват соединения (connection hijacking) невозможны. На сегодняшний день существуют две версии протокола SSH (SSH-1 и SSH-2), причем вторая версия усовершенствована и расширена по сравнению с первой. Пользователям ftp, telnet или rlogin, которые используют протокол, передающий данные в виде открытого текста, всегда рекомендуется переключиться на SSH.

OpenSSH — это реализация с открытым исходным кодом протокола SSH, позволяющая шифровать соединение в сети посредством набора программ. Технология работает по принципу сервер-клиент. На удалённой машине, на которой вы хотите выполнять команды, нужно запустить сервер OpenSSH. К этому серверу можно подключаться с помощью клиентов OpenSSH. На одном компьютере могут быть одновременно установлены и сервер, и клиент. Их запуск и настройка выполняется независимо друг от друга.

**Утилиты SSH**

К серверным утилитам OpenSSH относятся:

* sshd (OpenSSH Daemon) — программа демон для ssh. sshd ожидает подключений от клиентов.
* sftp-server (серверная подсистема SFTP) — это программа, которая участвует в передачи файлов по протоколу SFTP. sftp-server не предназначен для прямого вызова, обычно она задействуется автоматически демоном sshd.

Основного внимания на сервере требует sshd, а программа sftp-server будет запущена автоматически по мере необходимости.

К клиентским утилитам OpenSSH относятся:

* ssh (клиентская программа SSH) — вход на удалённую машину и выполнения команд на этой удалённой машине.
* scp — копирования файлов на удалённую машину с локальной, или в обратном направлении — с локальной на удалённую. При передаче данных используется зашифрованный канал.
* sftp — передача файлов по безопасному каналу.
* ssh-keygen — утилита для создания ключей аутентификации и управления ими. Также она может использоваться для отзыва ключей.

Это минимум, которые может понадобиться большинству пользователей для создания ключей, подключения к удалённой машине и при удалённом копировании файлов.

**Базовый синтаксис**

$ ssh [опции] имя пользователя@сервер [команда]

**Опции команды ssh**

f - перевести ssh в фоновый режим

g - разрешить удаленным машинам обращаться к локальным портам

l - имя пользователя в системе

n - перенаправить стандартный вывод в /dev/null

p - порт ssh на удаленной машине

q - не показывать сообщения об ошибках

v - режим отладки

x - отключить перенаправление X11

X - включить перенаправление Х11

C - включить сжатие

**Настройка сервера OpenSSH и безопасность**

Для настроек доступны следующие конфигурационные файлы:

/etc/ssh/sshd\_config - файл конфигурации сервера OpenSSH;

/etc/ssh/ssh\_config - файл конфигурации клиентской части OpenSSH;

~/.ssh/ - директория, в которой хранятся пользовательские SSH настройки;

~/.ssh/authorized\_keys или ~/.ssh/authorized\_keys - список ключей (RSA или DSA), которые используются для подключения к пользовательским аккаунтам;

/etc/nologin - если данный файл существует в системе, то sshd запретит подключаться всем пользователям кроме root в систему;

/etc/hosts.allow и /etc/hosts.deny - система запрета (часть безопасности). Работает по аналогии с ACL;

SSH порт по умолчанию - 22

Протокол SSH первой версии имеет проблемы с безопасностью, которые закрыты во второй версии. Поэтому, следует использовать вторую версию. Для этого нужно убедиться, что в файле /etc/ssh/sshd\_config указана опция Protocol 2. В версии 2 используются следующие алгоритмы шифрования: AES-128, AES-192, AES-256, blowfish, CAST-128, ArcFour.

Безопасность протокола SSH обеспечивается следующими программными решениями:

* Шифрование всего трафика, проходящего через SSH соединение, выполняемое по одному из возможных алгоритмов, выбираемых в процессе переговоров сторон сеанса связи. Шифрование трафика соединения, препятствует его перехвату и использованию в злонамеренных целях. За счет выбора различных алгоритмов шифрования, систему становится очень гибкой, позволяя, например, не использовать алгоритмы, в которых были обнаружены уязвимости или потенциальные угрозы безопасности, или использовать только те алгоритмы, которые поддерживаются каждой из сторон;
* Аутентификация SSH сервера производится всегда, при любом соединении, что не позволяет подменить трафик или сам сервер;
* Аутентификация SSH клиента может происходить различными способами, что, с одной стороны, делает сам процесс аутентификации более безопасным, с другой, делает систему еще более гибкой, упрощая работу с ней;
* Контроль целостности сетевых пакетов, дает возможность отследить незаконные изменения трафике соединения, в случае обнаружения данного факта, соединение обрывается незамедлительно;
* Временные параметры аутентификации, предотвращают возможность использования перехваченных и через некоторое время, расшифрованных данных соединения.

Протокол SSH поддерживает довольно разнообразные методы аутентификации и авторизации удаленных клиентов на SSH сервере, вот некоторые из них:

* GSSAPI-based аутентификация
* Host-based аутентификация;
* Аутентификация пользователя с помощью публичного ключа;
* Аутентификация "вызов-ответ" (challenge-response);
* Аутентификация пользователя по паролю;

SSH так же поддерживает дополнительные методы аутентификации пользователя, в зависимости от конкретной операционной системы ( например bsd\_auth или PAM). В общем случае, аутентификация пользователя происходит на основе открытых ключей. Клиент, пытающийся установить удаленное SSH соединение, шифрует данные, известным ему открытым серверным ключом, который он получает при первом подключении к серверу, и передает их на SSH сервер. Сервер в свою очередь расшифровывает данные, только ему известным, секретным ключом, и отсылает их клиенту. В такой схеме, клиент может быть уверен, что сервер является именно тем, за кого себя выдает. Таким образом, можно не полагаться на DNS и маршрутизацию, даже если злоумышленнику удалось подделать запись DNS или перенаправить пакеты на свой хост, аутентификация не будет пройдена, так как посторонние хосты не обладают необходимыми для этого ключами.

**Создание и настройка ключей OpenSSH**

Вход в SSH по публичному ключу (без пароля) достаточно удобен и безопасен.

Процесс настройки аутентификации по публичному ключу состоит из следующих шагов:

* Командой создаётся пара «публичный ключ — приватный ключ».
* Публичный ключ копируется на компьютер с сервером SSH, то есть на компьютер, к которому будет осуществляться подключение и на котором будут выполнятся команды.
* Затем подключение выполняется обычным способом, но ввод пароля уже не требуется.

Публичный ключ, который копируется на удалённый сервер, не является секретным. Один и тот же ключ можно использовать на разных серверах. Главное — хранить в секрете приватный ключ.

**Протокол SFTP**

Для надежной работы через SFTP-протокол необходимо иметь установленное защищенное соединение (например, SSH), которое проводит аутентификацию клиента и сервера и устанавливает факт их надежности, поскольку сам SFTP-протокол не проводит аутентификацию и не обеспечивает безопасность.

SFTP имеет ряд преимуществ перед своими предшественниками — FTP и SCP — таких, как прерывание передачи файла, удаление, возобновление передачи, связь переданных файлов с основными атрибутами, например, меткой даты/времени, а также более высокая платформонезависимость. SFTP-протокол реализуется через SFTP-сервер и SFTP-клиент, которые являются подсистемами OpenSSH.

Почти во всех случаях, SFTP предпочтительнее FTP, из-за его встроенной поддержки шифрования. FTP - небезопасный протокол, который следует использовать лишь в ограниченных случаях или в сети, которой Вы доверяете.

По умолчанию, SFTP использует протокол SSH для авторизации и установки безопасного соединения. По этой причине в SFTP доступны те же методы авторизации, которые существуют в SSH. Используйте утилиту командной строки sftp для подключения к удаленной системе sftp. Для этого вам нужны пользователь sftp и имя хоста или IP-адрес удаленного хоста: *sftp user@remote.host*

**Некоторые команды SFTP**

sftp> pwd - для проверки текущего рабочего каталога

sftp> ls - выведите список всех файлов и каталогов в текущем каталоге

sftp> cd – для перехода в другие каталоги

sftp> put - для загрузки файла из одной папки в другую

sftp> get - загрузить файлы c удалённого каталога

sftp> exit – выход из сеанса SFTP

# Подготовка к лабораторному практикуму

Лабораторную работу предлагается выполнить в программе виртуализации для операционных систем VMware Workstation. Так же для работы необходимо использовать 2 операционные системы Debian 10.4 (достаточно установить одну ОС, а затем клонировать ее), одна из которых будет выступать в качестве клиента, а другая в качестве сервера.

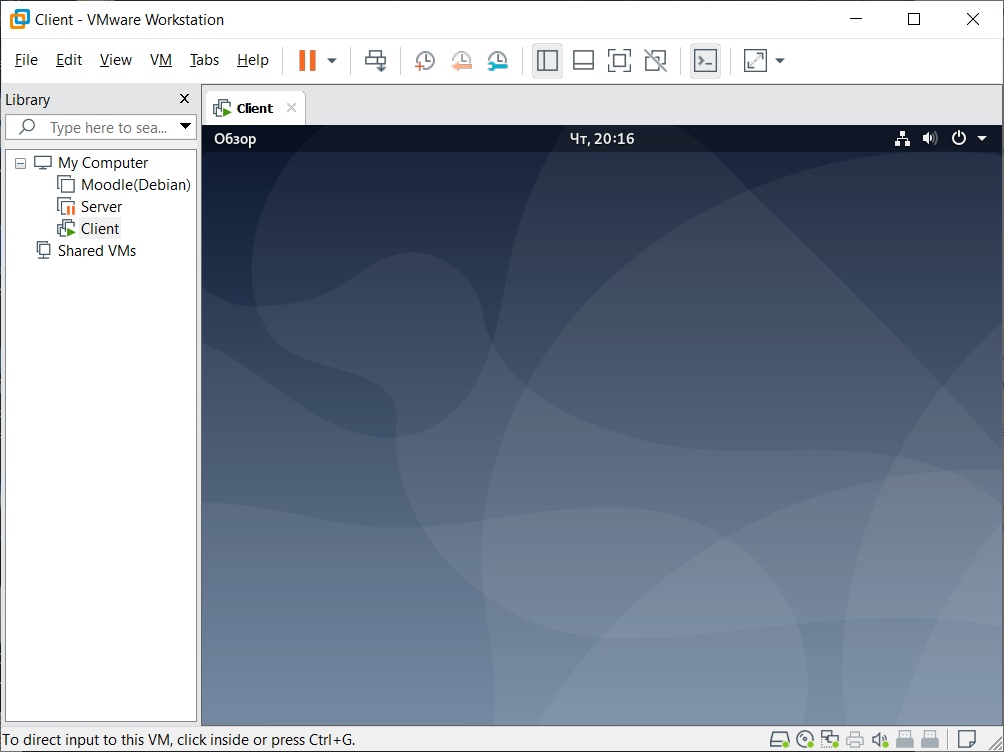


Рисунок 1 – Начальное окно VMware Workstation

Так же необходимо проверить, что служба OpenSSH усыновлена в операционной системе (*sudo apt-get install ssh)*.

Для выполнения лабораторной работы вы можете использовать как любой образ Linux, и при этом вовсе не обязательно выполнять работу с использованием виртуальной машины. В таком случае ход выполнение лабораторного практикума может измениться.

# Лабораторный практикум

1. Ознакомьтесь с теоретическим материалом.
2. Запустите VMware. В настройках сети (Edit -> Virtual Networking Editor -> Change Settings) убедитесь, что они соответствуют настройкам на Рисунке 2.

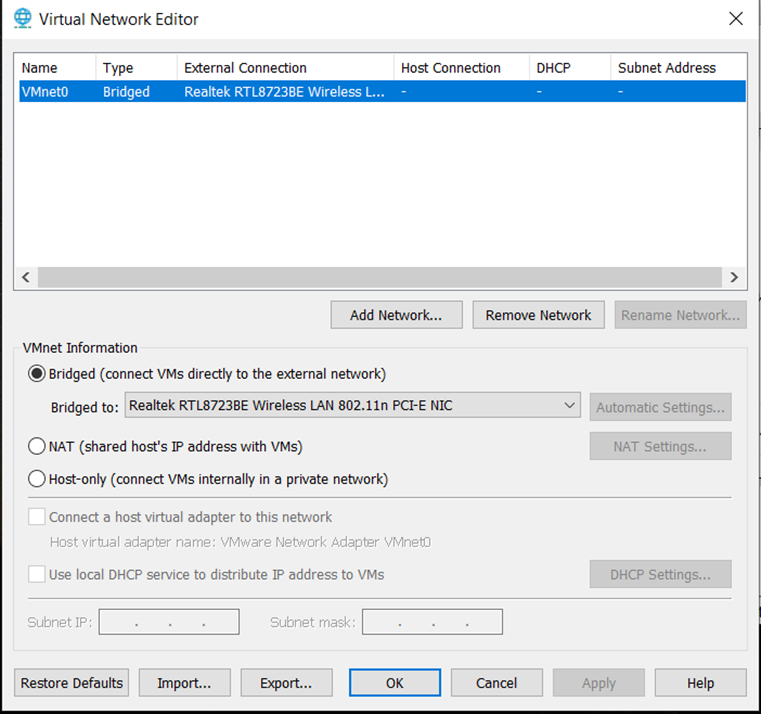


Рисунок 2 – Настройка VMware

1. Проверьте, что сетевой адаптер работает в режиме моста (bridged), как на Рисунке 3 (Settings -> Network Adapter).

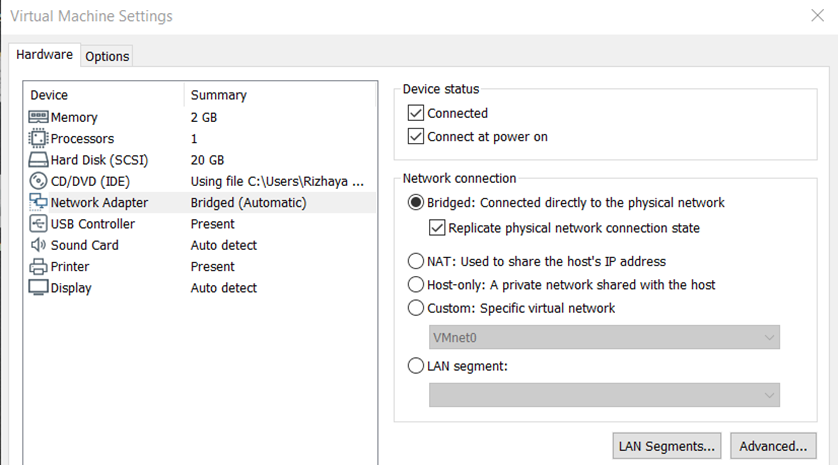


Рисунок 3 – Настройка сетевого адаптера

1. Запустите обе виртуальные машины. Определите, какую из них вы будете использовать в качестве клиента, а какую в качестве сервера.
2. Настройте сетевое окружение таким образом, чтобы виртуальные машины могли сообщаться между собой. Используйте команду *ip a*, чтобы выяснить IP-адрес машины. Запомните или запишите их для дальнейшей работы.
3. Проверьте соединение между машинами в обоих направлениях при помощи команды *ping*.
4. Создайте приветственное сообщение при успешном соединении по SSH. За это отвечает файл /etc/issue. Откройте его и введите любой текст, содержащий ваши ФИО:

*nano /etc/issue*

1. Сделайте скриншот приветственного сообщения.
2. Установите SSH-соединение с клиента на сервер посредством команды

*ssh ip-address*

1. Проверьте, удалось ли установить соединение по SSH. Для этого создайте файл 1.txt на сервере:

*touch 1.txt*

1. Попробуйте переместить файл. Создайте текстовый файл на клиенте с вашим именем и фамилией в названии, а также запишите внутрь фамилию.

*scp ~/test.txt user@host:documents*

1. Проверьте, что документ переместился на сервер корректно. Какие права доступа есть к этому документу?
2. Сделайте скриншот содержимого этого файла.
3. Разорвите соединение между клиентом и сервером.
4. Откройте для редактирования файл, содержащий настройки сервера SSH:

*su nano /etc/ssh/sshd.conf*

1. Посмотрите конфигурацию для серверов по умолчанию. Какие из них вам знакомы и за что отвечают? Сделайте скриншот этих настроек.
2. Поменяйте значение порта по умолчанию на любое другое свободное.
3. Сделайте так, чтобы для SSH соединения использовалась только вторая версия протокола.
4. Запретите Root доступ по SSH.
5. После изменения параметров перезапустите сервер следующей командой:

*service ssh restart*

1. Установите SSH соединение. Все ли внесенные вами изменения применились?
2. Вновь откройте файл *sshd.conf* и восстановите настройки по умолчанию.
3. Отключите проверку отпечатка ключа SSH:

*ssh -o "StrictHostKeyChecking=no" user@host*

1. Используйте команду *ssh-keygen -t rsa* для генерации пары ключей. Поле имя файла оставьте пустым по умолчанию, пароль вводится по желанию. Этот пароль позволяет установить дополнительную защиту — при подключении с помощью ключей не будет спрашиваться пароль пользователя, но будет спрашиваться пароль самого ключа.
2. Проверьте, что в результате было создано два файла:

~/.ssh/id\_rsa

~/.ssh/id\_rsa.pub

1. Скопируйте второй файл на удалённый компьютер, где запущен сервер SSH.
2. Создайте на сервере каталог .ssh:

*ssh user@host mkdir .ssh*

1. Скопировать содержимое файла id\_rsa.pub на сервер в файл *~/.ssh/authorized\_keys*.

*cat .ssh/id\_rsa.pub | ssh user@host 'cat >> .ssh/authorized\_keys'*

1. Выполните подключение с помощью клиента SSH. Требуется ли ввод пароля? Объясните почему.
2. Tshark — полнофункциональная, основанная на командной строке программа для мониторинга сетевого трафика. Для её установки зайдите с правами суперпользователя *su*
3. Используйте команду *apt-get install tshark*
4. Обновите модули командой *apt-get install*, *apt upgrade*, *apt update* (возможно, потребуется перезагрузка)
5. Используйте команду *tshark -D* для просмотра списка доступных интерфейсов. Какие из них вы узнаете?
6. Повторно установите SSH-соединение между клиентом и сервером.
7. Проанализируйте входящий трафик:

*tshark -i eth0*

1. Сделайте скриншот полученного SSH трафика.
2. Сформируйте и проанализируйте трафик, отличный от SSH. В чем сходства и различия между этим трафиком и трафиком SSH?
3. Подключитесь к серверу по SFTP при помощи команды:

*sftp username@remote\_hostname\_or\_IP*

1. Найдите на сервере файл со своей фамилией.
2. Установите соответствие между командами, используемыми в SFTP, запишите ответ в строку в документ с вашей фамилией на сервере.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | Перечисление файлов и каталогов в локальной, системе |  | a | ls |
| 2 | Перечисление файлов и каталогов в удаленной системе. |  | b | lpwd |
| 3 | Команда выхода в локальную оболочку |  | c | Lls |
| 4 | Узнать текущую локальную директорию |  | d | get -Pr |
| 5 | Получить файл с удаленной системы, сохранив соответствующие привилегии |  | e | ! |

1. Проверьте, достаточно ли места для передачи нужного Вам файла при помощи команды *df -h*
2. Получите удаленный файл с вашей фамилией при помощи команды *get*. Убедитесь в корректности передачи.
3. Сделайте скриншот терминала в момент передачи файла.
4. Выставьте права только на чтение на файл с вашей фамилией, находящийся на сервере, используйте команду *chmod 444*. Спросили ли у вас пароль при попытке изменения уровня прав? Сделайте вывод.
5. При помощи команды *get -Pr* скопируйте свой документ с новыми правами на клиент. Получилось ли сделать это?
6. Вернитесь на сервер и попробуйте записать что-нибудь в ваш файл. Сделайте вывод о безопасности использования SFTP соединения?
7. При помощи утилиты tshark проанализируйте трафик, создаваемый SFTP-соединением. Чем он отличается от трафика SSH?
8. Завершите SFTP-сессию. Сделайте вывод о преимуществах STFP соединения.

# Заключение

SSH использует шифрование с открытым ключом и был разработан в качестве замены Telnet и других небезопасных протоколов. SSH второй версии в настоящее время является доминирующим протоколом для доступа к учетным записям оболочки и в наши дни используется для входа и выполнения кода на удаленных хостах, просмотра веб-страниц с использованием зашифрованных прокси-клиентов и передачи файлов. Клиенты SSH доступны для всех основных операционных систем. Системы на основе Unix, включая Linux и Mac OS X, могут использовать OpenSSH.

Приложения для передачи файлов и VPN по умолчанию не используют шифрование (SSH), но используют SFTP - протокол передачи файлов SSH. SFTP всегда используется для передачи файлов по SSH, но на самом деле он спроектирован так, что его можно использовать в соответствии с другими протоколами. SFTP, являясь расширением SSH-2, использует защиту с помощью открытого ключа. Это означает, что данные зашифрованы, когда они передаются, и потенциальные перехваты относительно бесполезны.

Умение работать с протоколами SSH и SFTP важны для любого человека, отвечающего за информационную безопасность.

# Литература

1. Алгоритмы телекоммуникационных сетей. Автор Ю.А. Семенов
2. Документация OpenSSH / <https://www.openssh.com/manual.html>
3. Полное руководство по SSH в Linux и Windows / <https://hackware.ru/?p=9873>