SSH (OpenSSH)

OpenSSH (англ. Open Secure Shell — открытая безопасная оболочка) — набор программ, предоставляющих шифрование сеансов связи по компьютерным сетям с использованием протокола SSH.

Набор OpenSSH содержит следующие компоненты:

- **ssh**

  Замена для rlogin и telnet.

- **scp**

  Замена для rcp, использующая в современных версиях OpenSSH протокол SFTP (ранее использовался менее надёжный и гибкий SCP).

- **sftp**

  Замена для FTP-клиента, использующая протокол SFTP.

- **sshd**

  Демон, собственно предоставляющий защищённый доступ к ресурсам. Включает реализацию серверной части SFTP, пригодную для организации chroot-доступа для пользователей без необходимости копирования каких-либо файлов внутрь chroot.

- **sftp-server**

  Отдельная реализация подсистемы SFTP (серверная часть). Обладает бóльшими возможностями, чем встроенная в *sshd*.

- **ssh-keygen**

  Генератор пар ключей.

- **ssh-keysign**

  Утилита для проверки ключей хостов. Задействуется при использовании аутентификации по хостам (аналогично rsh) вместо проводимой по умолчанию аутентификации по пользователям.

- **ssh-keyscan**

  Вспомогательная утилита. Позволяет собирать публичные ключи с других хостов.

- **ssh-agent**

  Вспомогательная утилита. Поддерживает кэш закрытых ключей. Кэширование позволяет избегать частого ввода пароля для расшифровки ключей перед их использованием.

- **ssh-add**

  Вспомогательная утилита. Добавляет ключи в кэш *ssh-agent*.

Связанные файлы

- **/etc/ssh/** - место хранения файлов конфигурации
  - **sshd_config** - конфигурация ssh-сервера
  - **ssh_config** - конфигурация клиента

В **Debian** настройки клиентской части ssh делятся на глобальные и пользовательские. Глобальные клиентские настройки находятся в файле **/etc/ssh/ssh_config** и применяются ко всем пользователям. Пользовательские настройки могут находиться в домашнем каталоге пользователя, в **~/.ssh/config** и применяются к одному пользователю. **Файл пользовательских настроек не создаётся автоматически** в отличие от файла глобальных настроек клиентской части ssh.

Более расширенный разбор параметров файлов sshd_config и ssh_config:

https://www.aitishnik.ru/linux/ssh-debian/nastroyka-openssh.html

Конфигурация sshd_config

Параметр	Описание
Port 22	По умолчанию используется 22 порт. Можно указать кастомный. Теперь, чтобы подключиться к серверу нужно будет явно указать порт. **Например**, так: $ **ssh –l andrey –p 2203 192.168.123.254**
ListenAddress ::	Эти строки отвечают за настройку разграничений по сетевым интерфейсам, сетевому адресу или имени компьютера. По умолчанию сервер «слушает» (принимает подключения) на всех сетевых интерфейсах.
ListenAddress 0.0.0.0	
Protocol 2	Отвечает за версию протокола SSH
HostKey /etc/ssh/ssh_host_rsa_key	Строки **HostKey** необходимы для второй версии протокола SSH и отвечают за названия файлов ключей и их расположение. Первая строка отвечает за пару ключей RSA, вторая соответственно за пару ключей DSA. К названиям открытых (публичных) ключей добавляется **.pub**. Эти ключи используются при аутентификации с ключом хоста. Можно поменять слово host в названии ключей на имя нашего сервера, но мы сделаем это в части, посвященной генерации ключей.
PermitRootLogin no/yes	Разрешает или запрещает вход по SSH под суперпользователем
AllowUsers ssh_p	Добавляем параметр AllowUsers, **которого нет в конфигурационном файле по умолчанию**. Этот параметр разрешает доступ к серверу по протоколу SSH только для перечисленных пользователей.
RSAAuthentication yes	Оставляем включенной аутентификацию RSA. Включена и работает по умолчанию, если только не указать значение **no**
AuthorizedKeysFile .ssh/authorized_keys	Параметр определяет файл, в котором содержатся публичные ключи, используемые для аутентификации пользователей по открытому ключу. В записи могут присутствовать переменные, например %h означает домашний каталог пользователя, а %u – имя пользователя.

Этапы настройки:

1. **Создание пользователя**

```
useradd -m -s /bin/bash $USERNAME
passwd $USERNAME
su $USERNAME
mkdir .ssh
touch .ssh/authorized_keys
```

2. **Редактирование файла sshd_conf на сервере**

```
...
Port 22

HostKey /etc/ssh/ssh_host_rsa_key

PermitRootLogin yes
AllowUsers root ssh_p ssh_c

PubkeyAuthentication yes
AuthorizedKeysFile  .ssh/authorized_keys
...
```

3. **Генерация ключей на удаленной машине**

```
ssh-keygen -t rsa
```

В процессе создания ключей, можно изменить путь сохранения

4. **Передача публичного ключа на сервер**

```
scp .ssh/id_rsa.pub ssh_p@10.10.10.1:~/
```

Можно также воспользоваться утилитой **ssh-copy-id**.

У утилиты есть всего один ключ ***-i***. Для начала работы генерируется пара ключей, *при переносе обязательно нужно перейти в директорию .ssh* где и находится наша пара

```
$cd .ssh
$ssh-copy-id -i id_rsa.pub user@host
user@host's password:
```

5. **Добавление публичного ключа удаленной машины в авторизированный список**

```
cat id_rsa.pub >> .ssh/authorized_keys
```

Источники:

- https://www.aitishnik.ru/linux/ssh-debian.html
- https://itproffi.ru/ustanovka-i-nastrojka-servera-ssh-v-linux/
- https://habr.com/ru/post/122445/