

Практическая работа «Удаленный доступ: SSH»

Цель: Настройка протокола SSH для обеспечения защищенного удаленного доступа с использованием парольного доступа и доступа по ключу.

Описание: В приведенных примерах будет использоваться топология, представленная на рис. SSH_01

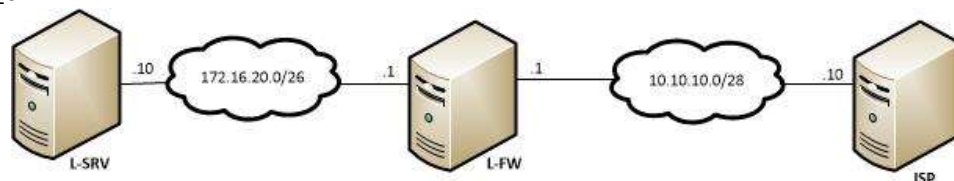


Рисунок SSH_01 Топология сети

Краткие теоретические сведения:

(информация, приведенная ниже на этой странице, взята с источника <https://losst.ru/avtorizatsiya-po-klyuchu-ssh>)

SSH или Secure Shell - это зашифрованный протокол, который часто используется для взаимодействия и удаленного управления серверами. Если необходимо что-либо сделать на удаленном сервере, скорее всего, придется воспользоваться SSH и работать через терминал. SSH существует несколько способов авторизации. Можно каждый раз вводить пароль пользователя или использовать более безопасный и надежный способ - ключи SSH. Что самое интересное, он более удобен для применения, поскольку даже не нужно будет вводить пароль.

Пароли передаются по безопасному каналу, но они недостаточно сложны для противостояния попыткам перебора. Вычислительная мощность современных систем в сочетании со специальными скриптами делают перебор очень простым. Конечно, существуют другие способы дополнительной безопасности, например, fail2ban, но аутентификация по ключу SSH более надежна.

Каждая пара ключей состоит из открытого и закрытого ключа. Секретный ключ сохраняется на стороне клиента и не должен быть доступен кому-либо еще. Утечка ключа позволит злоумышленнику войти на сервер, если не была настроена дополнительная аутентификация по паролю.

Открытый ключ используется для шифрования сообщений, которые можно расшифровать только закрытым ключом. Это свойство и используется для аутентификации с помощью пары ключей. Открытый ключ загружается на удаленный сервер, к которому необходимо получить доступ. Его нужно добавить в специальный файл `~/.ssh/authorized_keys`.

Когда клиент попытается выполнить проверку подлинности через этот ключ, сервер отправит сообщение, зашифрованное с помощью открытого ключа, если клиент сможет его расшифровать и вернуть правильный ответ - аутентификация пройдена (Рис. SSH_02).

SSH Key Authentication

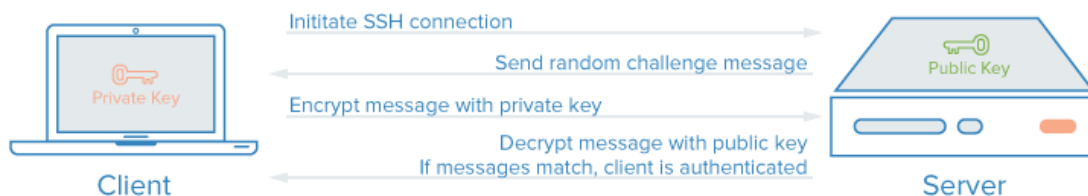


Рисунок SSH_02 Аутентификация SSH

Пример SSH_01:

В приведенной топологии (рис. SSH_01) настроить на компьютерах **L-SRV** и **ISP** удаленный доступ по протоколу SSH по следующим критериям (Таблица SSH_01):

Таблица SSH_01 Параметры подключения SSH

Устройство	Пользователи	Порт	Доступ
L-SRV	ssh_c ssh_p	1022	Пароль Ключ
ISP	ssh_isp	22	Ключ

В качестве пароля использовать **ssh_pass**

Проверку подключения осуществлять с компьютера **L-FW**

Настройка:

- 1) Настроить имена устройств и IP адресацию в соответствии с топологией.
- 2) Создать на всех устройствах пользователей, указанных в таблице SSH_01.

L-SRV (Debian):

```
useradd -m -s /bin/bash ssh_c
passwd ssh_c
ssh_pass
ssh_pass
```

ISP (Centos):

```
useradd ssh_isp
passwd ssh_isp
ssh_pass
ssh_pass
```

На устройстве L-FW создаем пользователей, для которых будет обеспечиваться доступ по ключам (**ssh_c** и **ssh_isp**).

- 3) На устройствах **L-SRV**, **ISP** включить доступ по SSH и установить необходимый порт:

L-SRV (Debian):

```
nano /etc/ssh/sshd_config
Port 1022
```

ISP (Centos):

```
vi /etc/ssh/sshd_config
Port 22
```

- 4) Перезапустить ssh на обоих устройствах:

```
systemctl restart sshd
```

- 5) Проверить доступ по паролям к устройствам L-SRV, ISP с хоста L-FW:

L-FW -> L-SRV

```
ssh ssh_c@172.16.20.10
```

Выводится ошибка:

```
root@debian:~# ssh ssh_c@172.16.20.10
ssh: connect to host 172.16.20.10 port 22: Connection refused
root@debian:~#
```

Связано это с тем, что по-умолчанию подключение осуществляется на порт 22. А в конфигурационном файле на L-SRV был установлен порт 1022.

Выполнить подключение на порт 1022 и ввести пароль **ssh_pass**

При первом подключении необходимо ввести **yes** на запрос демона sshd

```
ssh ssh_c@172.16.20.10 -p 1022
```

```
root@debian:~# ssh ssh_c@172.16.20.10 -p 1022
ssh_c@172.16.20.10's password: _
```

В случае удачного подключения в терминале будет выведено имя пользователя и имя хоста:

```
ssh_c@L-FW:~$
```

- 6) Для выхода ввести **exit**
- 7) Аналогично проверить возможность удаленного подключения под оставшимися пользователями к соответствующим устройствам.

Настройка подключения по ключу

Генерацию ключей будем проводить на устройстве, с которого будем подключаться к удаленным компьютерам, т.е. L-FW, а затем открытые пароли передадим на сервера

- 8) Зайти на L-FW под именем пользователя **ssh_p**

- 9) Сгенерировать ключ

```
ssh-keygen -t rsa -b 1024
```

- 10) Создается два файла **id_rsa** и **id_rsa.pub**. PUB ключ - это публичный, а **id_rsa** секретный. Необходимо файл **id_rsa.pub** перенести на сервер, куда будет осуществляться подключение. В данном случае – на **L-SRV** в директорию **/home/ssh_p/.ssh/** (в каталог того пользователя под которым будет соединение по ssh).

11) Перенести ключ на удаленный сервер. Для этого можно воспользоваться командой:
`ssh-copy-id ssh_p@172.16.20.10` или `ssh-copy-id ssh_p@L-SRV`

Ввести пароль

При этом на удаленном сервере (L-SRV) в каталоге `~/.ssh` создается файл `authorized_keys`, в который прописывается **открытый** ключ.

12) Проверить подключение к **L-SRV** с использованием *ключа*:

Перейти на **L-FW** и ввести команду:

```
ssh ssh_p@L-SRV -p 1022
```

Если все сделано правильно, то должно произойти подключение без запроса пароля:

```
ssh_p@L-SRV:~$
```

13) Для создания подключения по ключу на удаленный сервер **ISP** необходимо выполнить шаги 8)-12) для пользователя `ssh_isp`.

14) После создания и проверки подключений можно (и нужно) удалить файлы открытых ключей `id_rsa.pub` из каталога `~/.ssh` каждого пользователя на компьютере **L-FW**.

Таким образом, настроены подключения к удаленным серверам по протоколу `ssh`, причем для пользователей `ssh_p` и `ssh_isp` аутентификация происходит по ключам, а пользователь `ssh_c` должен вводить пароль.

Пример SSH 02:

В настроенную в предыдущем примере (`ssh_01`) схему подключения по протоколу `ssh` внести изменения, обеспечив подключение по нужным портам (таблица `SSH_01`) без необходимости их указания каждый раз в процессе аутентификации

Настройка:

1) Для выполнения данной настройки необходимо внести изменения в конфигурационный файл `/etc/ssh/ssh_config`, в котором содержатся настройки пользовательские настройки `ssh` на хосте с которого будет осуществляться подключение (**L-FW**). Порты для подключения предварительно должны быть настроены на удаленных серверах (см. п.3) пример `SSH_01`)

```
nano /etc/ssh/ssh_config
```

2) Добавить в конец файла следующие команды:

```
Host L-SRV
```

```
Port 1022
```

```
Host ISP
```

```
Port 22
```

3) Перезагрузить `sshd`

```
systemctl restart sshd
```

4) Проверить подключение к **L-SRV** без указания номера порта:

```
ssh ssh_p@L-SRV
```

5) Проверить подключение к **ISP** без указания номера порта:

```
ssh ssh_isp@ISP
```

Примечание: Порт **22** для `ssh` является стандартным портом подключения, поэтому для него настройки можно не делать.

Пример SSH 03:

Выполнить настройку **L-FW**, которая будет обеспечивать подключение по протоколу `ssh` к компьютеру **L-SRV** **по-умолчанию** под именем пользователя `ssh_p`, а к **ISP** - под `ssh_isp`.

Настройка:

L-FW:

1) Отредактировать файл `ssh_config`:

```
nano /etc/ssh/ssh_config
```

```
Host L-SRV
```

```
Port 1022
```

```
User ssh_p
```

```
Host ISP
```

```
Port 22
```

```
User ssh_isp
```

2) Перезагрузить `sshd`

```
systemctl restart sshd
```

3) Проверить подключение к **L-SRV** без указания имени пользователя:
`ssh L-SRV`

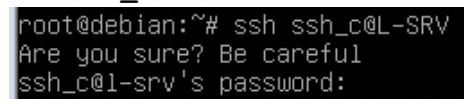
4) Проверить подключение к **ISP** без указания имени пользователя:
`ssh ISP`

Пример SSH_04:

Вывести для пользователей, подключающихся к **L-SRV** по протоколу **ssh** баннер следующего содержания: **Are you sure? Be careful**. Не выводить данный баннер для пользователя **ssh_c**

Настройка:

- 1) Зайти на устройство L-SRV по логином root
- 2) Создать нового пользователя **empl** с паролем **cisco**
- 3) Создать файл следующего содержания
`nano /etc/ssh/ssh_banner`
Are you sure? Be careful
- 4) Отредактировать конфигурационный файл `sshd_config`
`nano /etc/ssh/ssh_config`
Banner /etc/ssh/ssh_banner
- 5) Перезагрузить `sshd`
`systemctl restart sshd`
- 6) Подключиться к L-SRV с устройства L-FW под именем **ssh_c**
`ssh ssh_c@L-SRV`



Нажать CTRL+Z для отмены подключения

- 7) Подключиться под именем **empl**
Баннер выводится для обоих пользователей

Отключения баннера для ssh_c

Для выполнения данной настройки необходимо воспользоваться директивой **Match** в файле `sshd_config`.

Директива **Match** представляет собой начало условного блока. Если выполнены все критерии, указанные в строке Match, директивы в последующих строках блока выполняются, позволяя обойти значения глобальных директив файла `sshd_config` для случая, являющегося критерием директивы Match. Блоком считаются все строки, идущие после строки с критерием (Match - строки) до следующей match-строки или до конца файла. Аргумент директивы Match - одна или несколько пар записей критериев.

- 1) Зайти на устройство L-SRV по логином root
- 2) Открыть для редактирования конфигурационный файл `sshd_config`
`nano /etc/ssh/ssh_config`

3) В конце файла ввести строки
Match User ssh_c
Banner no

- 4) Перезагрузить `sshd`
`systemctl restart sshd`

- 5) Подключиться к **L-SRV** с устройства **L-FW** сначала под именем **ssh_c** а затем под именем **empl**

Баннер выводится для пользователя **empl**

Задание для самостоятельного выполнения:

Отключить на **L-SRV** парольную аутентификацию `ssh`, оставив только вход по ключу. По паролю на **L-SRV** может заходить только пользователь **empl**

Для решения данной задачи и задач, возникающих в дальнейшем можно (и нужно!!!) использовать источники информации:

- 1) **man**
- 2) Источники в интернете:
http://www.opennet.ru/base/sec/ssh_tips.txt.html
<https://www.help.ubuntu.ru/wiki/ssh>
и многие другие...