

## Лабораторная работа № 11. Настройка безопасного удалённого доступа по SSH

### 11.1. Цель работы

Приобретение практических навыков по настройке удалённого доступа к серверу с помощью SSH.

### 11.2. Предварительные сведения

#### 11.2.1. Удалённый доступ по SSH

Протокол SSH (Secure Shell) позволяет организовать защищённый и безопасный удалённый доступ к узлам сети поверх небезопасных каналов связи.

Безопасность соединений по протоколу SSH обеспечивается за счёт шифрования соединения, аутентификации сервера и клиента, проверки целостности передаваемых по организованному соединению данных.

SSH-соединение имеет серверную и клиентскую части. Серверная часть на Unix/Linux узлах реализуется процессом `sshd` по умолчанию через TCP-порт 22. Настройки `sshd` обычно располагаются в файле `/etc/ssh/sshd_config`.

За клиентскую часть отвечает команда `ssh`, имеющая следующий синтаксис:

`ssh опции хост пользователь@хост команда`

Некоторые опции `ssh`:

- v — вывод отладочной информации о ходе процесса установки соединения;
- f — переход в фоновый режим;
- l пользователь — регистрация на удалённом узле под указанным в параметрах пользователем;
- p порт — подключение через указанный в параметрах порт;
- L порт:хост:хостпорт — переадресация порта локального узла на хостпорт удалённого узла;
- R порт:хост:хостпорт — переадресация порта удалённого локального хоста на хостпорт локального узла.

Подробнее об `ssh` см. в соответствующем ман руководстве.

#### 11.2.2. Безопасность при организации удалённого доступа по SSH

Использование SSH для организации удалённого доступа к узлам сети извне — удобное решение. Но при этом существует ряд угроз безопасности, если узел сети непосредственно виден из Интернета. К таким угрозам, в частности, относятся так называемые «атаки по словарю» и атаки через известные открытые на узле порты. Например, злоумышленник может использовать тот факт, что удалённый доступ по SSH обычно организуется через порт 22, а каждый узел Unix/Linux имеет учётную запись `root`. Основываясь на этой информации, злоумышленник может попытаться войти в систему как `root`, просто подбирая пароль.

Возможные меры по усилению безопасности при организации удалённого доступа:

- запрет прямого удалённого доступа для пользователя `root`;
- отключение возможности ввода пароля и переход на использование ключей безопасности при удалённом доступе;
- переадресация стандартного для SSH порта 22 на нестандартный;

- политика разрешения удалённого доступа к узлам сети по SSH лишь ограниченного круга пользователей.

### 11.3. Задание

1. Настройте запрет удалённого доступа на сервер по SSH для пользователя root (см. раздел 11.4.1).
2. Настройте разрешение удалённого доступа к серверу по SSH только для пользователей группы vagrant и вашего пользователя (см. раздел 11.4.2).
3. Настройте удалённый доступ к серверу по SSH через порт 2022 (см. раздел 11.4.3).
4. Настройте удалённый доступ к серверу по SSH по ключу (см. раздел 11.4.4).
5. Организуйте SSH-туннель с клиента на сервер, перенаправив локальное соединение с TCP-порта 80 на порт 8080 (см. раздел 11.4.5).
6. Используя удалённое SSH-соединение, выполните с клиента несколько команд на сервере (см. раздел 11.4.6).
7. Используя удалённое SSH-соединение, запустите с клиента графическое приложение на сервере (см. раздел 11.4.7).
8. Напишите скрипт для Vagrant, фиксирующий действия по настройке SSH-сервера во внутреннем окружении виртуальной машины server. Соответствующим образом внесите изменения в Vagrantfile (см. раздел 11.4.8).

## 11.4. Последовательность выполнения работы

### 11.4.1. Запрет удалённого доступа по SSH для пользователя root

1. На сервере задайте пароль для пользователя root, если этого не было сделано ранее:  

```
sudo -i  
  
passwd root
```
2. На сервере в дополнительном терминале запустите мониторинг системных событий:  

```
sudo -i  
  
journalctl -x -f
```
3. С клиента попытайтесь получить доступ к серверу посредством SSH-соединения через пользователя root:  

```
ssh root@server.user.net
```

В отчёте поясните, что при этом происходит.
4. На сервере откройте файл /etc/ssh/sshd\_config конфигурации sshd для редактирования и запретите вход на сервер пользователю root, установив:  

```
PermitRootLogin no
```
5. После сохранения изменений в файле конфигурации перезапустите sshd:  

```
systemctl restart sshd
```
6. Повторите попытку получения доступа с клиента к серверу посредством SSH-соединения через пользователя root:  

```
ssh root@server
```

В отчёте поясните, что при этом происходит.

### 11.4.2. Ограничение списка пользователей для удалённого доступа по SSH

1. С клиента попытайтесь получить доступ к серверу посредством SSH-соединения через пользователя user (вместо user укажите вашего пользователя):  
`ssh user@server.user.net`  
В отчёте поясните, что при этом происходит.
2. На сервере откройте файл `/etc/ssh/sshd_config` конфигурации sshd на редактирование и добавьте строку  
`AllowUsers vagrant`
3. После сохранения изменений в файле конфигурации перезапустите sshd:  
`systemctl restart sshd`
4. Повторите попытку получения доступа с клиента к серверу посредством SSH-соединения через пользователя user:  
`ssh user@server.user.net`  
В отчёте поясните, что при этом происходит.
5. В файле `/etc/ssh/sshd_config` конфигурации sshd внесите следующее изменение:  
`AllowUsers vagrant, user`
6. После сохранения изменений в файле конфигурации перезапустите sshd и вновь попытайтесь получить доступ с клиента к серверу посредством SSH-соединения через пользователя user.  
В отчёте поясните, что при этом происходит.

### 11.4.3. Настройка дополнительных портов для удалённого доступа по SSH

1. На сервере в файле конфигурации sshd `/etc/ssh/sshd_config` найдите строку Port и ниже этой строки добавьте:  
`Port 22`  
`Port 2022`  
Эта запись сообщает процессу sshd о необходимости организации соединения через два разных порта, что даёт гарантию возможности открыть сеансы SSH, даже если была сделана ошибка в конфигурации.
2. После сохранения изменений в файле конфигурации перезапустите sshd:  
`systemctl restart sshd`
3. Посмотрите расширенный статус работы sshd:  
`systemctl status -l sshd`  
Система должна сообщить вам об отказе в работе sshd через порт 2022. Дополнительно посмотрите сообщения в терминале с мониторингом системных событий. В отчёте поясните суть системных сообщений.
4. Исправьте на сервере метки SELinux к порту 2022:  
`semanage port -a -t ssh_port_t -p tcp 2022`
5. В настройках межсетевого экрана откройте порт 2022 протокола TCP:  
`firewall-cmd --add-port=2022/tcp`  
`firewall-cmd --add-port=2022/tcp --permanent`
6. Вновь перезапустите sshd и посмотрите расширенный статус его работы. Статус должен показать, что процесс sshd теперь прослушивает два порта.
7. С клиента попытайтесь получить доступ к серверу посредством SSH-соединения через пользователя user (вместо user укажите вашего пользователя):  
`ssh user@server.user.net`  
После открытия оболочки пользователя введите `sudo -i` для получения доступа root.
8. Повторите попытку получения доступа с клиента к серверу посредством SSH-соединения через пользователя user, указав порт 2022:

```
ssh -p2022 user@server.user.net
```

После открытия оболочки пользователя введите `sudo -i` для получения доступа root.

#### 11.4.4. Настройка удалённого доступа по SSH по ключу

В этом упражнении вы создаёте пару из открытого и закрытого ключей для входа на сервер.

1. На сервере в конфигурационном файле `/etc/ssh/sshd_config` задайте параметр, разрешающий аутентификацию по ключу:

```
PubkeyAuthentication yes
```

2. После сохранения изменений в файле конфигурации перезапустите `sshd`.
3. На клиенте сформируйте SSH-ключ, введя в терминале под пользователем `user` (вместо `user` используйте ваш логин):

```
ssh-keygen
```

Когда вас спросят, хотите ли вы использовать кодовую фразу, нажмите `[Enter]`, чтобы использовать установку без пароля. При запросе имени файла, в котором будет храниться закрытый ключ, примите предлагаемое по умолчанию имя файла (`~/.ssh/id_rsa`).

Когда вас попросят ввести кодовую фразу, нажмите `[Enter]` дважды.

4. Закрытый ключ теперь будет записан в файл `~/.ssh/id_rsa`, а открытый ключ записывается в файл `~/.ssh/id_rsa.pub`.
5. Скопируйте открытый ключ на сервер, введя на клиенте (вместо `user` укажите вашего пользователя):

```
ssh-copy-id user@server.user.net
```

При запросе введите пароль пользователя на удалённом сервере.

6. Попробуйте получить доступ с клиента к серверу посредством SSH-соединения (вместо `user` используйте ваш логин):

```
ssh user@server.user.net
```

Теперь вы должны пройти аутентификацию без ввода пароля для учётной записи удалённого пользователя.

#### 11.4.5. Организация туннелей SSH, перенаправление TCP-портов

1. На клиенте посмотрите, запущены ли какие-то службы с протоколом TCP:

```
lsof | grep TCP
```

2. Перенаправьте порт 80 на `server.user.net` на порт 8080 на локальной машине (вместо `user` используйте ваш логин):

```
ssh -fNL 8080:localhost:80 user@server.user.net
```

3. Вновь на клиенте посмотрите, запущены ли какие-то службы с протоколом TCP:

```
lsof | grep TCP
```

В отчёте прокомментируйте полученную при выводе на экран информацию.

4. На клиенте запустите браузер и в адресной строке введите `localhost:8080`. Убедитесь, что отобразится страница с приветствием «Welcome to the server.user.net server».

#### 11.4.6. Запуск консольных приложений через SSH

1. На клиенте откройте терминал под пользователем `user` (вместо `user` используйте ваш логин).
2. Посмотрите с клиента имя узла сервера:

```
ssh user@server.user.net hostname
```

3. Посмотрите с клиента список файлов на сервере:

```
ssh user@server.user.net ls -Al
```

- Посмотрите с клиента почту на сервере:

```
ssh user@server.user.net MAIL=~/.Maildir/ mail
```

#### 11.4.7. Запуск графических приложений через SSH (X11Forwarding)

- На сервере в конфигурационном файле `/etc/ssh/sshd_config` разрешите отображать на локальном клиентском компьютере графические интерфейсы X11:

```
X11Forwarding yes
```

- После сохранения изменения в конфигурационном файле перезапустите `sshd`.
- Попробуйте с клиента удалённо подключиться к серверу и запустить графическое приложение, например `firefox` (вместо `user` используйте ваш логин):

```
ssh -YC user@server.user.net firefox
```

#### 11.4.8. Внесение изменений в настройки внутреннего окружения виртуальной машины

- На виртуальной машине `server` перейдите в каталог для внесения изменений в настройки внутреннего окружения `/vagrant/provision/server/`, создайте в нём каталог `ssh`, в который поместите в соответствующие подкаталоги конфигурационный файл `sshd_config`:

```
cd /vagrant/provision/server
mkdir -p /vagrant/provision/server/ssh/etc/ssh
cp -R /etc/ssh/sshd_config /vagrant/provision/server/ssh/etc/ssh/
```

- В каталоге `/vagrant/provision/server` создайте исполняемый файл `ssh.sh`:

```
cd /vagrant/provision/server
touch ssh.sh
chmod +x ssh.sh
```

Открыв его на редактирование, пропишите в нём следующий скрипт:

```
#!/bin/bash
```

```
echo "Provisioning script $0"
```

```
echo "Copy configuration files"
```

```
cp -R /vagrant/provision/server/ssh/etc/* /etc
```

```
restorecon -vR /etc
```

```
echo "Configure firewall"
```

```
firewall-cmd --add-port=2022/tcp
```

```
firewall-cmd --add-port=2022/tcp --permanent
```

```
echo "Tuning SELinux"
```

```
semanage port -a -t ssh_port_t -p tcp 2022
```

```
echo "Restart sshd service"
```

```
systemctl restart sshd
```

- Для отработки созданного скрипта во время загрузки виртуальной машины `server` в конфигурационном файле `Vagrantfile` необходимо добавить в разделе конфигурации для сервера:

```
server.vm.provision "server ssh",  
  type: "shell",  
  preserve_order: true,  
  path: "provision/server/ssh.sh"
```

## 11.5. Содержание отчёта

1. Титульный лист с указанием номера лабораторной работы и ФИО студента.
2. Формулировка задания работы.
3. Описание результатов выполнения задания:
  - скриншоты (снимки экрана), фиксирующие выполнение работы;
  - подробное описание настроек служб в соответствии с заданием;
  - полные тексты конфигурационных файлов настраиваемых в работе служб;
  - результаты проверки корректности настроек служб в соответствии с заданием (подтверждённые скриншотами).
4. Выводы, согласованные с заданием работы.
5. Ответы на контрольные вопросы.

## 11.6. Контрольные вопросы

1. Вы хотите запретить удалённый доступ по SSH на сервер пользователю root и разрешить доступ пользователю alice. Как это сделать?
2. Как настроить удалённый доступ по SSH через несколько портов? Для чего это может потребоваться?
3. Какие параметры используются для создания туннеля SSH, когда команда ssh устанавливает фоновое соединение и не ожидает какой-либо конкретной команды?
4. Как настроить локальную переадресацию с локального порта 5555 на порт 80 сервера server2.example.com?
5. Как настроить SELinux, чтобы позволить SSH связываться с портом 2022?
6. Как настроить межсетевой экран на сервере, чтобы разрешить входящие подключения по SSH через порт 2022?