Отчёт по лабораторной работе №11

Дисциплина: Администрирование сетевых подсистем

true

Содержание

Цель работы

Целью данной работы является приобретение практических навыков по настройке удалённого доступа к серверу с помощью SSH.

Задание

1. Настроить запрет удалённого доступа на сервер по SSH для пользователя root
2. Настроить разрешение удалённого доступа к серверу по SSH только для пользователей группы vagrant и вашего пользователя
3. Настроить удалённый доступ к серверу по SSH через порт 2022
4. Настроить удалённый доступ к серверу по SSH по ключу
5. Организовать SSH-туннель с клиента на сервер, перенаправив локальное соединение с TCP-порта 80 на порт 8080
6. Используя удалённое SSH-соединение, выполнить с клиента несколько команд на сервере
7. Используя удалённое SSH-соединение, запустить с клиента графическое приложение на сервере
8. Написать скрипт для Vagrant, фиксирующий действия по настройке SSH-сервера во внутреннем окружении виртуальной машины server. Соответствующим образом внести изменения в Vagrantfile

Выполнение лабораторной работы

Запрет удалённого доступа по SSH для пользователя root

Загрузили нашу операционную систему и перешли в рабочий каталог с проектом: `cd /var/tmp/tbmanturov/vagrant` ([рис. @fig-001])

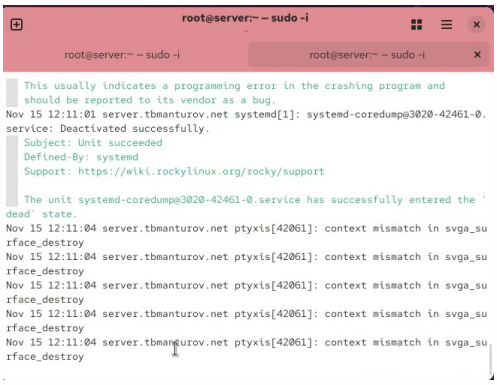
Запустили виртуальную машину server: `make server-up` ([рис. @fig-002])

На сервере задали пароль для пользователя root: `sudo -i` и `passwd root` ([рис. @fig-003])

```
[root@server.tbmanturov.net ~]# passwd root
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: password updated successfully
[root@server.tbmanturov.net ~]#
```

Задание пароля для пользователя root

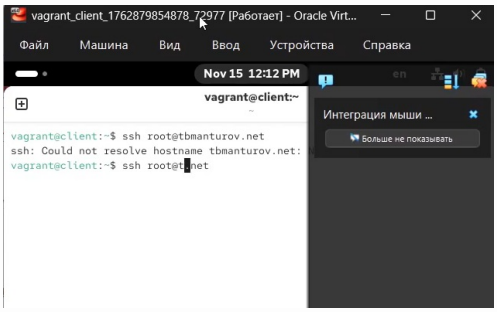
На сервере в дополнительном терминале запустите мониторинг системных событий: `sudo -i` и `journalctl -x -f` ([рис. @fig-004])



Мониторинг системных событий

Далее запустили виртуальную машину client ([рис. @fig-005])

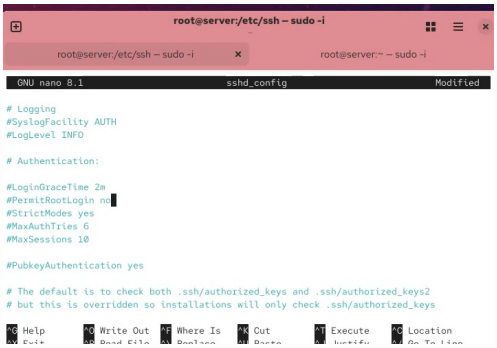
С клиента попытались получить доступ к серверу посредством SSH-соединения через пользователя root: `ssh root@server.tbmanturov.net` ([рис. @fig-006])



Попытка получить доступ к серверу посредством SSH-соединения через пользователя root

Несмотря на правильно введенный пароль для пользователя root, не получилось подключиться, так как в конфигурации ssh запрещено подключение для пользователя root с помощью пароля (по умолчанию используется настройка PermitRootLogin prohibit-password)

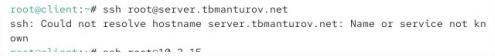
На сервере октрыли файл */etc/ssh/sshd_config* конфигурации sshd для редактирования и запретили вход на сервер пользователю root, установив: `PermitRootLogin no` ([рис. @fig-007])



Редактирование файла /etc/ssh/sshd_config (1)

После сохранения изменений в файле конфигурации перезапустили sshd: `systemctl restart sshd` ([рис. @fig-008])

Повторили попытку получения доступа с клиента к серверу посредством SSH-соединения через пользователя root: `ssh root@server` ([рис. @fig-009])

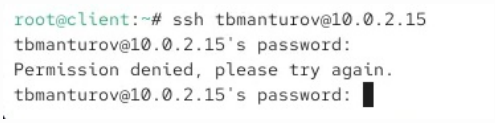


Повторная попытка получить доступ к серверу посредством SSH-соединения через пользователя root

Теперь также запрещен доступ root пользователю на сервер любыми средствами аутентификации

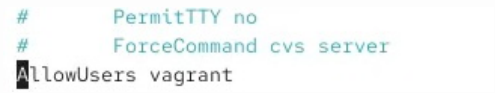
Ограничение списка пользователей для удалённого доступа по SSH

С клиента попытались получить доступ к серверу посредством SSH-соединения через пользователя tbmanturov: `ssh tbmanturov@tbmanturov.user.net`. Всё проходит успешно ([рис. @fig-009])



Попытка получить доступ к серверу посредством SSH-соединения через пользователя tbmanturov

На сервере открыли файл */etc/ssh/sshd_config* конфигурации sshd на редактирование и добавили строку `AllowUsers vagrant` ([рис. @fig-011])



Редактирование файла /etc/ssh/sshd_config (2)

После сохранения изменений в файле конфигурации перезапустили sshd: `systemctl restart sshd` ([рис. @fig-012])

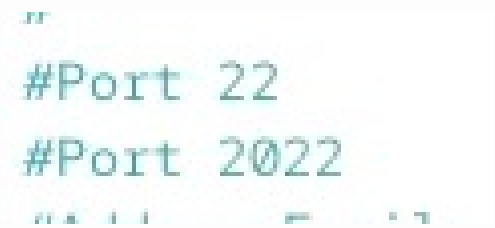
Повторили попытку получения доступа с клиента к серверу посредством SSH-соединения через пользователя tbmanturov: `ssh tbmanturov@server.tbmanturov.net` ([рис. @fig-013])

SSH сервер теперь разрешает подключение только пользователю vagrant, а все остальные пользователи (включая tbmanturov) блокируются. Директива `AllowUsers` задает “белый список” пользователей, которым разрешено подключаться по SSH

Настройка дополнительных портов для удалённого доступа по SSH

На сервере в файле конфигурации sshd */etc/ssh/sshd_config* нашли строку Port и ниже этой строки добавили ([рис. @fig-017]):

```
Port 22
Port 2022
```



Редактирование файла /etc/ssh/sshd_config (4)

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

Далее после сохранения изменений в файле конфигурации перезапустили sshd: `systemctl restart sshd` ([рис. @fig-018])

Посмотрели расширенный статус работы sshd: `systemctl status -l sshd`. Система сообщила нам об отказе в работе sshd через порт 2022 ([рис. @fig-019])

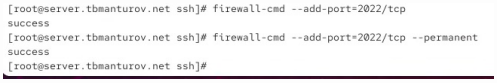
Расширенный статус работы sshd (1)

Далее исправили на сервере метки SELinux к порту 2022: `semanage port -a -t ssh_port_t -p tcp 2022` ([рис. @fig-020])

Исправление меток SELinux к порту 2022

В настройках межсетевого экрана открыли порт 2022 протокола TCP ([рис. @fig-021]):

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

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

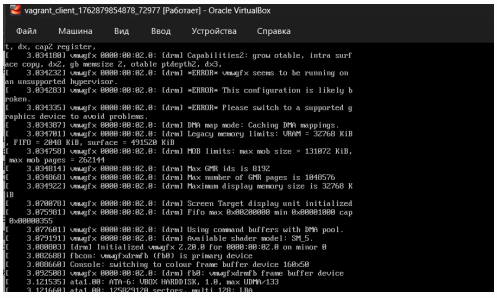


Настройка межсетевго экрана

Вновь перезапустили sshd и посмотрели расширенный статус его работы. Статус показывает, что процесс sshd теперь прослушивает два порта ([рис. @fig-023])

Расширенный статус работы sshd (2)

С клиента попытались получить доступ к серверу посредством SSH-соединения через пользователя tbmanturov: `ssh tbmanturov@server.tbmanturov.net`. После открытия оболочки пользователя ввели `sudo -i` для получения доступа root. Отлогинились от root и нашего пользователя на сервере, введя дважды `logout` ([рис. @fig-024])



Успешное подключение к серверу

Повторили попытку получения доступа с клиента к серверу посредством SSH-соединения через пользователя tbmanturov, указав порт 2022: `ssh -p2022 tbmanturov@server.tbmanturov.net`. После открытия оболочки пользователя ввели `sudo -i` для получения доступа root. Отлогинились от root и нашего пользователя на сервере, введя дважды `logout`

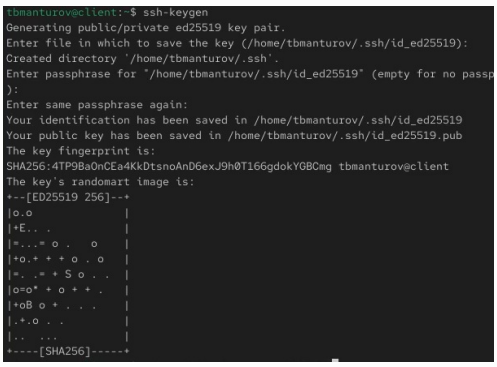
Настройка удалённого доступа по SSH по ключу

На сервере в конфигурационном файле */etc/ssh/sshd_config* задали параметр, разрешающий аутентификацию по ключу: `PubkeyAuthentication yes` ([рис. @fig-026])

Редактирование файла /etc/ssh/sshd_config (5)

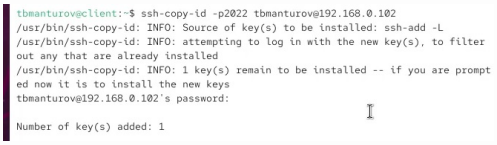
После сохранения изменений в файле конфигурации перезапустили sshd

На клиенте сформировали SSH-ключ, введя в терминале под пользователем tbmanturov: `ssh-keygen -t rsa -b 4096`. Закрытый ключ теперь будет записан в файл *~/.ssh/id_rsa*, а открытый ключ записывается в файл *~/.ssh/id_rsa.pub* ([рис. @fig-028])



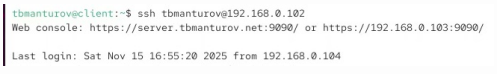
Формировнаие SSH-ключа

Скопировали открытый ключ на сервер, введя на клиенте: `ssh-copy-id tbmanturov@server.tbmanturov.net`. При запросе ввели пароль пользователя на удалённом сервере ([рис. @fig-029])



Копирование открытого ключа на серевр

Попробовали получить доступ с клиента к серверу посредством SSH-соединения: `ssh tbmanturov@server.tbmanturov.net`. Теперь мы проходим аутентификацию без ввода пароля для учётной записи удалённого пользователя. Отлогинились с сервера, используя комбинацию клавиш *Ctrl* + *d* ([рис. @fig-030])



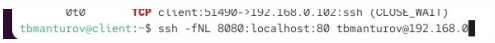
Аутентификация без ввода пароля

Организация туннелей SSH, перенаправление TCP-портов

На клиенте посмотрели, запущены ли какие-то службы с протоколом TCP: `lsof | grep TCP` ([рис. @fig-031])

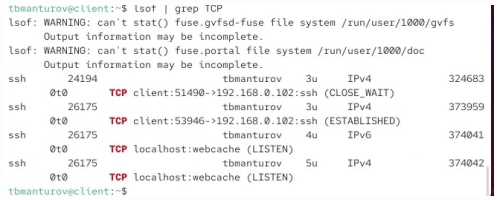
Просмотр запущенных служб с протоколом TCP

Далее перенаправили порт 80 на server.tbmanturov.net на порт 8080 на локальной машине: `ssh -fNL 8080:localhost:80 tbmanturov@server.tbmanturov.net` ([рис. @fig-032])



Перенаправление TCP порта

Вновь на клиенте посмотрели, запущены ли какие-то службы с протоколом TCP: `lsof | grep TCP` ([рис. @fig-033])



Повторный просмотр запущенных служб с протоколом TCP

Комментарии к выводу:

1. SSH порт форвардинг работает - видно TCP соединение между клиентом и сервером:

- client.tbmanturov.net:47494->ns.tbmanturov.net:ssh

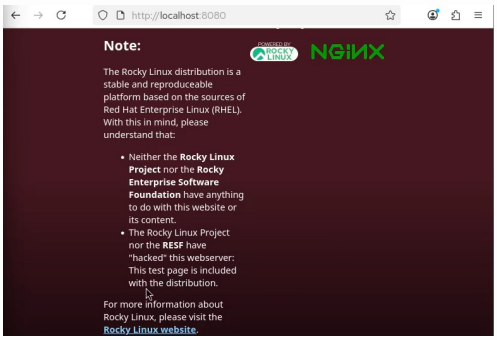
2. Созданы локальные прослушивающие сокеты на порту 8080:

- TCP localhost:webcache (LISTEN) - это порт 8080 (webcache)
- Оба для IPv4 и IPv6

3. SSH туннель активен - соединение в состоянии ESTABLISHED
4. Порт 80 сервера перенаправлен на локальный порт 8080 клиента

Команда ssh -fNL 8080:localhost:80 успешно создала SSH туннель, который перенаправляет локальный порт 8080 на порт 80 удаленного сервера через зашифрованное соединение.

На клиенте запустили браузер и в адресной строке ввели localhost:8080. Убедились, что отобразилась страница с приветствием «Welcome to the server.tbmanturov.net server» ([рис. @fig-034])



Запуск браузера на клиенте и ввод в адресной строке localhost:8080

Запуск консольных приложений через SSH

На клиенте открыли терминал под пользователем tbmanturov и посмотрели с клиента имя узла сервера: `ssh tbmanturov@server.tbmanturov.net hostname` ([рис. @fig-035])

```
tbmanturov@client:~$ ssh tbmanturov@192.168.0.102 hostname
server.tbmanturov.net
```

Просмотр имени узла сервера с клиента

Посмотрели с клиента список файлов на сервере: `ssh tbmanturov@server.tbmanturov.net ls -Al` ([рис. @fig-036])

```
drwxr-xr-x.  2 tbmanturov tbmanturov    6 Nov 11 17:27 Desktop
drwxr-xr-x.  2 tbmanturov tbmanturov    6 Nov 11 17:27 Documents
drwxr-xr-x.  2 tbmanturov tbmanturov    6 Nov 11 17:27 Downloads
drwx------.  4 tbmanturov tbmanturov   32 Nov 11 17:28 .local
drwxr-xr-x.  5 tbmanturov tbmanturov   54 Nov 13 11:02 .mozilla
drwxr-xr-x.  2 tbmanturov tbmanturov    6 Nov 11 17:27 Music
drwxr-xr-x.  2 tbmanturov tbmanturov    6 Nov 11 17:27 Pictures
drwxr-xr-x.  2 tbmanturov tbmanturov    6 Nov 11 17:27 Public
drwx------.  2 tbmanturov tbmanturov   29 Nov 15 17:00 .ssh
drwxr-xr-x.  2 tbmanturov tbmanturov    6 Nov 11 17:27 Templates
-rw-r-----.  1 tbmanturov tbmanturov    6 Nov 13 08:51 .vboxclient-clipboard-tty
2-control.pid
-rw-r-----.  1 tbmanturov tbmanturov    6 Nov 15 13:57 .vboxclient-clipboard-tty
2-service.pid
-rw-r-----.  1 tbmanturov tbmanturov    6 Nov 13 08:51 .vboxclient-draganddrop-t
ty2-control.pid
-rw-r-----.  1 tbmanturov tbmanturov    6 Nov 13 08:51 .vboxclient-hostversion-t
ty2-control.pid
-rw-r-----.  1 tbmanturov tbmanturov    6 Nov 13 08:51 .vboxclient-seamless-tty2
-control.pid
-rw-r-----.  1 tbmanturov tbmanturov    6 Nov 13 08:51 .vboxclient-vmsvga-sessio
n-tty2-control.pid
drwxr-xr-x.  2 tbmanturov tbmanturov    6 Nov 11 17:27 Videos
```

Просмотр списка файлов на сервере с клиента

Посмотрели с клиента почту на сервере: `ssh tbmanturov@server.tbmanturov.net MAIL=~/Maildir/ mail` ([рис. @fig-037])

```
tbmanturov@client:~$ ssh tbmanturov@192.168.0.102 MAIL=~/Maildir/mail
tbmanturov@client:~$
```

Просмотр почты на сервере с клиента

Запуск графических приложений через SSH (X11Forwarding)

На сервере в конфигурационном файле */etc/ssh/sshd_config* разрешили отображать на локальном клиентском компьютере графические интерфейсы X11: `X11Forwarding yes` ([рис. @fig-038])

```
#AllowAgentForwarding yes
#AllowTcpForwarding yes
#GatewayPorts no
X11Forwarding no
#X11DisplayOffset 10
#X11UseLocalhost yes
#PermitTTY yes
#PrintMotd yes
#PrintLastLog yes
#TCPKeepAlive yes
#PermitUserEnvironment no
```

Редактирование файла /etc/ssh/sshd_config (6)

После сохранения изменения в конфигурационном файле перезапустили sshd

Далее попробовали с клиента удалённо подключиться к серверу и запустить графическое приложение, например firefox: `ssh -YC tbmanturov@server.tbmanturov.net firefox` ([рис. @fig-040])

```
tbmanturov@client:~$ ssh -YC tbmanturov@192.168.0.102 firefox
Warning: No xauth data; using fake authentication data for X11 forwarding.
X11 forwarding request failed on channel 0
Error: no DISPLAY environment variable specified
tbmanturov@client:~$
```

Подключение к серверу с клиента и запуск графического приложения

firefox

Внесение изменений в настройки внутреннего окружения виртуальной машины

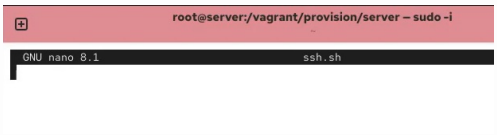
На виртуальной машине server перешли в каталог для внесения изменений в настройки внутреннего окружения */vagrant/provision/server/*, создали в нём каталог ssh, в который поместили в соответствующие подкаталоги поместили конфигурационный файл sshd_config ([рис. @fig-041]):

```
cd /vagrant/provision/server

mkdir -p /vagrant/provision/server/ssh/etc/ssh

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

Копирование конфигурационного файла sshd_config

В каталоге /vagrant/provision/server создали исполняемый файл ssh.sh: Открыв его на редактирование, прописали в нём следующий скрипт ([рис. @fig-043]):

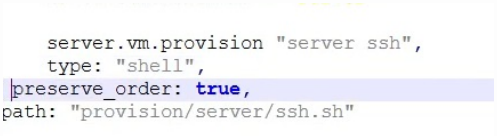
```
#!/bin/bash
echo "Provisioning script $0"
echo "Copy configuration files"
cp -R /vagrant/provision/server/ssh/etc/* /etc
restorecon -vR /etc
echo "Configure firewall"
firewall-cmd --add-port=2022/tcp
firewall-cmd --add-port=2022/tcp --permanent
echo "Tuning SELinux"
semanage port -a -t ssh_port_t -p tcp 2022
echo "Restart sshd service"
systemctl restart sshd
```



Редактирование файла ssh.sh

Для отработки созданного скрипта во время загрузки виртуальной машины server в конфигурационный файл Vagrantfile добавили в разделе конфигурации для сервера ([рис. @fig-044]):

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



Редактирование файла Vagrantfile

После этого можно выключать виртуальные машины server и client: `make server-halt` и `make client-halt` ## Контрольные вопросы + ответы

1. Вы хотите запретить удалённый доступ по SSH на сервер пользователю root и разрешить доступ пользователю alice. Как это сделать?

В конфигурационном файле SSH /etc/ssh/sshd_config:

```
# Запрет удалённого доступа пользователю root
PermitRootLogin no

# Разрешение доступа пользователю alice
AllowUsers alice
```

После внесения изменений, необходимо перезапустить службу SSH: sudo service ssh restart

2. Как настроить удалённый доступ по SSH через несколько портов? Для чего это может потребоваться?

В конфигурационном файле /etc/ssh/sshd_config добавьте строки:

```
# Первый порт (по умолчанию 22)
Port 22
# Второй порт
Port 2022
```

После изменений перезапустите службу SSH. Это может быть полезно для повышения безопасности, а также для избежания конфликтов с другими службами, использующими порт 22.

3. Какие параметры используются для создания туннеля SSH, когда команда ssh устанавливает фоновое соединение и не ожидает какой-либо конкретной команды?

ssh -N -f -L local_port:destination_host:remote_port user@ssh_server -N: Не выполнять команду на удаленном хосте. -f: Перевести ssh в фоновый режим после установки туннеля.

4. Как настроить локальную переадресацию с локального порта 5555 на порт 80 сервера server2.example.com?

ssh -L 5555:server2.example.com:80 user@ssh_server

Теперь, при подключении к локальному порту 5555, трафик будет перенаправляться через SSH к порту 80 на сервере server2.example.com.

5. Как настроить SELinux, чтобы позволить SSH связываться с портом 2022?

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

Данная команда добавляет правило SELinux, разрешая использование порта 2022 для сервиса ssh.

6. Как настроить межсетевой экран на сервере, чтобы разрешить входящие подключения по SSH через порт 2022?

sudo firewall-cmd –permanent –add-port=2022/tcp

sudo firewall-cmd –reload

Эти команды добавляют правило в межсетевой экран для разрешения входящих подключений по SSH через порт 2022 и перезагружают конфигурацию межсетевого экрана.

Выводы

В ходе выполнения лабораторной работы №11 мы приобрели практические навыки по настройке удалённого доступа к серверу с помощью SSH.

Список литературы

1. Лабораторая работа №11