

Лабораторная работа №11

Настройка безопасного удалённого доступа по протоколу SSH

Сахно Никита НФИбд-02-23

Содержание

1	Цель работы	1
2	Задание	1
3	Выполнение лабораторной работы.....	2
3.1	Запрет удалённого доступа по SSH для пользователя root.....	2
3.2	Ограничение списка пользователей для удалённого доступа по SSH	3
3.3	Настройка дополнительных портов для удалённого доступа по SSH	5
3.4	Настройка удалённого доступа по SSH по ключу.....	7
3.5	Организация туннелей SSH, перенаправление TCP-портов	8
3.6	Запуск консольных приложений через SSH	8
3.7	Запуск графических приложений через SSH (X11Forwarding)	9
3.8	Внесение изменений в настройки внутреннего окружения виртуальной машины	10
4	Выводы.....	11

1 Цель работы

Приобрести практические навыки по настройке удалённого доступа к серверу с помощью SSH.

2 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.

3 Выполнение лабораторной работы

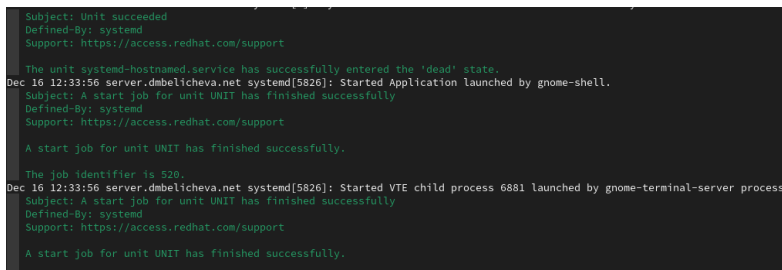
3.1 Запрет удалённого доступа по SSH для пользователя root

На сервере зададим пароль для пользователя root, если этого не было сделано ранее:

```
sudo -i  
passwd root
```

На сервере в дополнительном терминале запустим мониторинг системных событий:

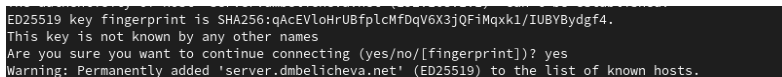
```
sudo -i  
journalctl -x -f
```



The screenshot shows the output of the `journalctl -x -f` command. It displays several log entries, including messages about the `systemd-hostnamed.service` entering the 'dead' state, a start job for unit `UNIT` finishing successfully, and a VTE child process being launched by `gnome-terminal-server`. The output is color-coded with green for success and red for errors.

Мониторинг системных событий

С клиента попытаемся получить доступ к серверу посредством SSH-соединения через пользователя root: `ssh root@server.nvsakhno.net`



The screenshot shows the output of an SSH connection attempt. It displays the key fingerprint for the host `server.nvsakhno.net` and asks for confirmation to continue connecting. The output is color-coded with green for success and red for errors.

Получение доступа к серверу посредством SSH-соединения

В доступе отказано.

На сервере откроем файл `/etc/ssh/sshd_config` конфигурации `sshd` для редактирования и запретим вход на сервер пользователю root, установив: `PermitRootLogin no`

```

GNU nano 5.6.1
# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:
#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

```

Редактирование файла

После сохранения изменений в файле конфигурации перезапустим sshd: `systemctl restart sshd`

Повторим попытку получения доступа с клиента к серверу посредством SSH-соединения через пользователя root: `ssh root@server.nvsakhno.net`

```

j# ssh root@server.dmbelicheva.net
password:
again.
password:
again.
password:
permission denied (publickey,gssapi-keyex,gssapi-with-mic,password).
j# ^C

```

Получение доступа к серверу посредством SSH-соединения

В доступе с клиента к серверу посредством SSH соединения через пользователя root отказано. Так и должно быть, ведь мы запретили вход на сервер пользователю root.

3.2 Ограничение списка пользователей для удалённого доступа по SSH

С клиента попытаемся получить доступ к серверу посредством SSH-соединения через пользователя dmbelicheva: `ssh nvaskhno@server.dmbelicheva.net`

```

ED25519 key fingerprint is SHA256:qAcEVloHrUBfplcMfDqV6X3jQFiMqxk1/IUBVBydgf4.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'server.dmbelicheva.net' (ED25519) to the list of known hosts.
dmbelicheva@server.dmbelicheva.net's password:
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Sat Dec 16 12:21:52 2023

```

Получение доступа к серверу посредством SSH-соединения

Соединение через пользователя nvaskhno произошло успешно.

На сервере откроем файл `/etc/ssh/sshd_config` конфигурации sshd на редактирование и добавим строку `AllowUsers vagrant`

```
GNU nano 5.6.1
# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
PermitRootLogin no
AllowUsers vagrant
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes
```

Редактирование файла

После сохранения изменений в файле конфигурации перезапустим sshd: `systemctl restart sshd`

Повторим попытку получения доступа с клиента к серверу посредством SSH-соединения через пользователя nvaskhno: `ssh nvaskhno@server.nvsakhno.net`

```
icheva.net (192.168.1.1)' can't be established.
ohrUBfplcMfDqV6X3jQFIMqk1/IUBYBydgr4.
Warning: Permanently added 'icheva.net' (ED25519) to the list of known hosts.
sword:
sword:
Access denied (publickey,gssapi-keyex,gssapi-with-mic,password).
```

Получение доступа к серверу посредством SSH-соединения

В доступе отказано.

В файле `/etc/ssh/sshd_config` конфигурации sshd внесем следующее изменение:
`AllowUsers vagrant nvaskhno`

```
# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
```

Редактирование файла

После сохранения изменений в файле конфигурации перезапустим sshd и вновь попытаемся получить доступ с клиента к серверу посредством SSH-соединения через пользователя user.

```
Last failed login: Sat Dec 16 13:12:37 UTC 2023 from 192.168.1.1 on ssh:notty
There were 6 failed login attempts since the last successful login.
Last login: Sat Dec 16 12:58:52 2023 from 192.168.1.30
```

Получение доступа к серверу посредством SSH-соединения

Теперь доступ успешно получен, поскольку мы разрешили пользователю dmbelicheva доступ к серверу посредством ssh.

3.3 Настройка дополнительных портов для удалённого доступа по SSH

На сервере в файле конфигурации sshd /etc/ssh/sshd_config найдем строку Port и ниже этой строки добавим:

```
Port 22
Port 2022
```

```
# semanage port -a -t ssh_port_t -p tcp #PORTNUMBER
#
#Port 22
Port 22
Port 2022
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::
```

Редактирование файла

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.

После сохранения изменений в файле конфигурации перезапустим sshd: `systemctl restart sshd`

Посмотрим расширенный статус работы sshd: `systemctl status -l sshd`

```
ssh.service - OpenSSH server daemon
Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; preset: enabled)
Active: active (running) since Sat 2023-12-16 13:16:33 UTC; 13s ago
Docs: man:sshd(8)
      man:sshd_config(5)
Main PID: 7612 (sshd)
Tasks: 1 (limit: 5724)
Memory: 1.6M
CPU: 33ms
CGroup: /system.slice/ssh.service
        └─7612 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Dec 16 13:16:33 server.dmbelicheva.net systemd[1]: Starting OpenSSH server daemon...
Dec 16 13:16:33 server.dmbelicheva.net sshd[7612]: main: sshd: ssh-rsa algorithm is disabled
Dec 16 13:16:33 server.dmbelicheva.net sshd[7612]: error: Bind to port 2022 on 0.0.0.0 failed: Permission denied.
Dec 16 13:16:33 server.dmbelicheva.net sshd[7612]: error: Bind to port 2022 on :: failed: Permission denied.
Dec 16 13:16:33 server.dmbelicheva.net sshd[7612]: Server listening on 0.0.0.0 port 22.
Dec 16 13:16:33 server.dmbelicheva.net sshd[7612]: Server listening on :: port 22.
Dec 16 13:16:33 server.dmbelicheva.net systemd[1]: Started OpenSSH server daemon.
```

Расширенный статус работы sshd

Система должна сообщить вам об отказе в работе sshd через порт 2022. Дополнительно посмотрим сообщения в терминале с мониторингом системных событий.

```

A start job for unit dbus-1.1-org.fedoraproject.SetroubleshootPrivileged2.service has finished successfully.

The job identifier is 3888.
Dec 16 13:16:41 server.dmbelicheva.net setroubleshoot[7613]: SELinux is preventing /usr/sbin/sshd from name_bind access on the tcp_socket port 2022. For complete SELinux messages run: sealert -l 8ed6ac2a-9568-4b46-8178-f5c769c989e1
Dec 16 13:16:41 server.dmbelicheva.net setroubleshoot[7613]: SELinux is preventing /usr/sbin/sshd from name_bind access on the tcp_socket port 2022.

***** Plugin bind_ports (92.2 confidence) suggests *****

If you want to allow /usr/sbin/sshd to bind to network ports, then you need to modify the port type.
Do
# semanage port -a -t PORT_TYPE -p tcp 2022
where PORT_TYPE is one of the following: ssh_port_t,

```

Мониторинг системных событий

Видно, что отказ происходит из-за запрета SELinux на работу с этим портом.

Исправим на сервере метки SELinux к порту 2022: `semanage port -a -t ssh_port_t -p tcp 2022`

В настройках межсетевого экрана откроем порт 2022 протокола TCP:

```

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

```

```

# semanage port -a -t ssh_port_t -p tcp 2022
# firewall-cmd --add-port=2022/tcp
# firewall-cmd --add-port=2022/tcp --permanent

```

Настройка межсетевого экрана

Вновь перезапустим `sshd` и посмотрим расширенный статус его работы. Статус должен показать, что процесс `sshd` теперь прослушивает два порта.

```

• sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; preset: enabled)
   Active: active (running) since Sat 2023-12-16 13:20:07 UTC; 3s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Main PID: 7671 (sshd)
     Tasks: 1 (limit: 5724)
    Memory: 1.6M
       CPU: 23ms
   CGroup: /system.slice/ssh.service
           └─7671 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Dec 16 13:20:07 server.dmbelicheva.net systemd[1]: Starting OpenSSH server daemon...
Dec 16 13:20:07 server.dmbelicheva.net sshd[7671]: main: sshd: ssh-rsa algorithm is disabled
Dec 16 13:20:07 server.dmbelicheva.net sshd[7671]: Server listening on 0.0.0.0 port 2022.
Dec 16 13:20:07 server.dmbelicheva.net sshd[7671]: Server listening on :: port 2022.
Dec 16 13:20:07 server.dmbelicheva.net sshd[7671]: Server listening on 0.0.0.0 port 22.
Dec 16 13:20:07 server.dmbelicheva.net sshd[7671]: Server listening on :: port 22.
Dec 16 13:20:07 server.dmbelicheva.net systemd[1]: Started OpenSSH server daemon.

```

Расширенный статус работы sshd

С клиента попытаемся получить доступ к серверу посредством SSH-соединения через пользователя `nvsakhno`: `ssh nvsakhno@server.nvsakhno.net`

```

dmbelicheva@server.dmbelicheva.net ~$ password:
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Sat Dec 16 13:28:11 2023 from 192.168.1.1

```

Получение доступа к серверу посредством SSH-соединения

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

Повторим попытку получения доступа с клиента к серверу посредством SSH-соединения через пользователя user, указав порт 2022: `ssh nvsakhno@server.nvsakhno.net`

```
Activate the web console with: systemctl enable --now cockpit.socket
Last login: Sat Dec 16 13:31:40 2023 from 192.168.1.1
```

Получение доступа к серверу посредством SSH-соединения через порт 2022

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

3.4 Настройка удалённого доступа по SSH по ключу

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

На сервере в конфигурационном файле `/etc/ssh/sshd_config` зададим параметр, разрешающий аутентификацию по ключу: `PubkeyAuthentication yes`

```
# Authentication:
#LoginGraceTime 2m
PermitRootLogin no
AllowUsers vagrant dmbelicheva
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

PubkeyAuthentication yes
```

Редактирование файла

После сохранения изменений в файле конфигурации перезапустим sshd.

На клиенте сформируем SSH-ключ, введя в терминале под пользователем nvsakhno: `ssh-keygen`

Когда спросят, хотим ли мы использовать кодовую фразу, нажмем Enter, чтобы использовать установку без пароля. При запросе имени файла, в котором будет храниться закрытый ключ, примем предлагаемое по умолчанию имя файла (`~/.ssh/id_rsa`). Когда попросят ввести кодовую фразу, нажмем Enter дважды.

```
+---[RSA 3072]-----+
|
|  . . . .
|  ..oo .
|  .o..o
|  .ooo..
|  o +S+o++o.
|  .o+.oo0o=.
|  ..E o..B.B
|  ... O..B0o
|  .. O..++
+---[SHA256]-----+
```

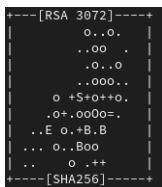
Формирование ключа ssh

Закрытый ключ теперь будет записан в файл `~/.ssh/id_rsa`, а открытый ключ записывается в файл `~/.ssh/id_rsa.pub`.

Скопируем открытый ключ на сервер, введя на клиенте: `ssh-copy-id nvsakhno@server.nvsakhno.net`

При запросе введем пароль пользователя на удалённом сервере.

Попробуем получить доступ с клиента к серверу посредством SSH-соединения: `ssh nvsakhno@server.nvsakhno.net`



Копирование открытого ssh ключа и получение доступа к серверу

Теперь пройдем аутентификацию без ввода пароля для учётной записи удалённого пользователя.

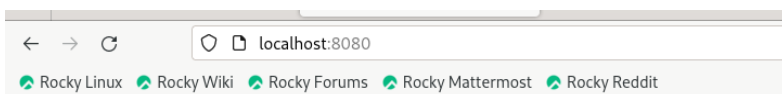
3.5 Организация туннелей SSH, перенаправление TCP-портов

На клиенте посмотрим, запущены ли какие-то службы с протоколом TCP: `lsof | grep TCP`

Перенаправим порт 80 на `server.dmbelicheva.net` на порт 8080 на локальной машине: `ssh -fNL 8080:localhost:80 nvsakhno@server.nvsakhno.net`

Вновь на клиенте посмотрим, запущены ли какие-то службы с протоколом TCP: `lsof | grep TCP`

На клиенте запустим браузер и в адресной строке введем `localhost:8080`. Убедимся, что отобразится страница с приветствием «Welcome to the server.dmbelicheva.net server».



localhost:8080

3.6 Запуск консольных приложений через SSH

На клиенте откройте терминал под пользователем `dmbelicheva`. Посмотрите с клиента имя узла сервера: `ssh nvsakhno@server.nvsakhno.net hostname`

Посмотрите с клиента список файлов на сервере: `ssh nvsakhno@server.nvsakhno.net ls -Al`

Посмотрите с клиента почту на сервере: `ssh nvsakhno@server.nvsakhno.net`
`MAIL=~/.Maildir/ mail`

```
301 Dec 16 13:33 .bash_history
18 Jan 23 2023 .bash_logout
141 Jan 23 2023 .bash_profile
546 Nov 6 11:06 .bashrc
4096 Nov 13 17:24 .cache
4096 Nov 24 17:05 .config
6 Nov 6 10:54 Desktop
18 Dec 2 19:06 Documents
6 Nov 6 10:54 Downloads
32 Nov 6 10:54 .local
4096 Dec 11 10:30 Maildir
54 Nov 13 17:24 .mozilla
6 Nov 6 10:54 Music
6 Nov 6 10:54 Pictures
6 Nov 6 10:54 Public
71 Dec 16 13:39 .ssh
6 Nov 6 10:54 Templates
6 Dec 16 12:21 .vboxclient-clipboard-tty1-control.pid
6 Dec 16 12:21 .vboxclient-clipboard-tty1-service.pid
6 Dec 16 12:21 .vboxclient-display-svg-x11-tty1-control.pid
6 Dec 16 12:21 .vboxclient-display-svg-x11-tty1-service.pid
6 Dec 16 12:21 .vboxclient-draganddrop-tty1-control.pid
6 Dec 16 12:21 .vboxclient-draganddrop-tty1-service.pid
6 Dec 16 12:22 .vboxclient-hostversion-tty1-control.pid
6 Dec 16 12:21 .vboxclient-seamless-tty1-control.pid
6 Dec 16 12:21 .vboxclient-seamless-tty1-service.pid
6 Dec 16 12:22 .vboxclient-vmsvga-session-tty1-control.pid
6 Nov 6 10:54 Videos
318 Dec 16 12:21 .xsession-errors
318 Dec 11 09:24 .xsession-errors.old
```

Запуск консольных приложений через SSH

3.7 Запуск графических приложений через SSH (X11Forwarding)

На сервере в конфигурационном файле `/etc/ssh/sshd_config` разрешим отображать на локальном клиентском компьютере графические интерфейсы X11: `X11Forwarding yes`

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

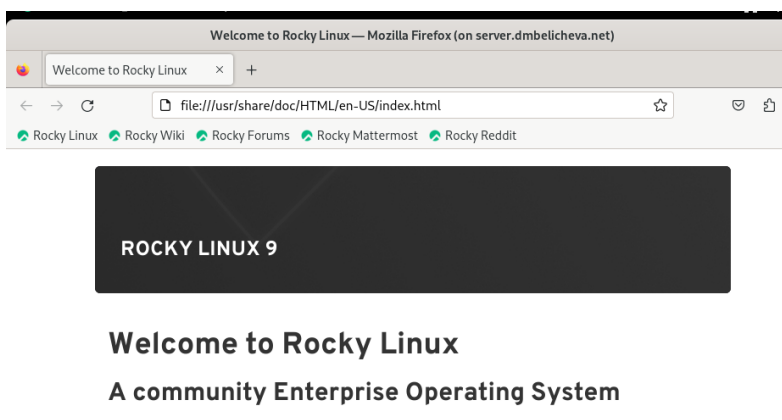
Редактирование файла

После сохранения изменения в конфигурационном файле перезапустим `sshd`. Попробуем с клиента удалённо подключиться к серверу и запустить графическое приложение, например `firefox`: `ssh -YC user@server.nvsakhno.net firefox`

```
ssh -YC user@server.nvsakhno.net firefox
.xauthority does not exist
[0][GFX1-]: glxtest: ManageChildProcess failed
ldProcess failed

[0][GFX1-]: glxtest: ManageChildProcess failed
or, error_code=1, request_code=154, minor_code=1 (t=4.72659) [GFX1-]: glxtest:
s, minor_code=1
```

Запуск графических приложений через SSH



Результат запуска графического приложения через SSH

3.8 Внесение изменений в настройки внутреннего окружения виртуальной машины

На виртуальной машине server перейдем в каталог для внесения изменений в настройки внутреннего окружения `/vagrant/provision/server/`, создадим в нём каталог `ssh`, в который поместим 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` создадим исполняемый файл `ssh.sh`:

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

Открыв его на редактирование, пропишем в нём следующий скрипт:

```
root@server:/vagrant/provision/server
GNU nano 5.6.1 ssh.sh
#!/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

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"
```

4 Выводы

В процессе выполнения данной лабораторной работы я приобрел практические навыки по настройке удалённого доступа к серверу с помощью SSH.