

Лабораторная работа №11

Настройка безопасного удалённого доступа по протоколу SSH

Демидова Екатерина Алексеевна

Содержание

| | | |
|----------|---|-----------|
| 1 | Цель работы | 4 |
| 2 | Задание | 5 |
| 3 | Выполнение лабораторной работы | 6 |
| 3.1 | Запрет удалённого доступа по SSH для пользователя root | 6 |
| 3.2 | Ограничение списка пользователей для удалённого доступа по SSH | 8 |
| 3.3 | Настройка дополнительных портов для удалённого доступа по SSH | 10 |
| 3.4 | Настройка удалённого доступа по SSH по ключу | 13 |
| 3.5 | Организация туннелей SSH, перенаправление TCP-портов | 14 |
| 3.6 | Запуск консольных приложений через SSH | 15 |
| 3.7 | Запуск графических приложений через SSH (X11Forwarding) . . . | 16 |
| 3.8 | Внесение изменений в настройки внутреннего окружения виртуальной машины | 17 |
| 4 | Контрольные вопросы | 19 |
| 5 | Выводы | 21 |

Список иллюстраций

| | | |
|------|---|----|
| 3.1 | Попытка установить SSH-соединение | 7 |
| 3.2 | Запрет входа на сервер пользователю root | 7 |
| 3.3 | Повторная попытка SSH-соединение | 8 |
| 3.4 | Попытка установить SSH-соединение с клиента | 8 |
| 3.5 | Изменение разрешенных пользователей для sshd | 9 |
| 3.6 | Определение службы аутентификации пользователей | 9 |
| 3.7 | Изменение разрешенных пользователей для sshd | 9 |
| 3.8 | Временный запуск SMTP-сервера | 10 |
| 3.9 | Добавление портов в файл конфигураций | 10 |
| 3.10 | Расширенный статус работы sshd | 11 |
| 3.11 | Мониторинг системных сообщений | 11 |
| 3.12 | Просмотр расширенного статуса работы sshd после настройки ра- боты по порту 2022 | 12 |
| 3.13 | Установка SSH-соединение с клиента | 12 |
| 3.14 | Формирования SSH-ключа на клиенте | 13 |
| 3.15 | Установка SSH-соединения с сервером с клиента | 14 |
| 3.16 | Просмотр активных служб с протоколом TCP | 14 |
| 3.17 | Просмотр локального сервера в браузере на клиенте | 15 |
| 3.18 | Просмотр информации о сервере с клиента через ssh | 15 |
| 3.19 | Запуск графического приложения через ssh | 16 |
| 3.20 | Создание окружения для внесения изменений в настройки окру- жающей среды | 17 |
| 3.21 | Скрипта файла /vagrant/provision/server/ssh.sh | 17 |
| 3.22 | Изменение конфигурационного файла Vagrant | 18 |

1 Цель работы

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

2 Задание

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

3 Выполнение лабораторной работы

3.1 Запрет удалённого доступа по SSH для пользователя root

Загрузим нашу операционную систему и перейдем в рабочий каталог с проектом:

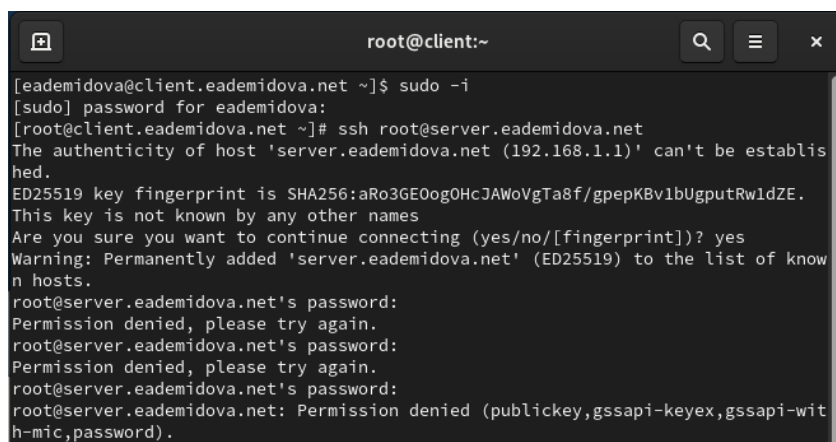
```
cd /var/tmp/eademidova/vagrant
```

Затем запустим виртуальную машину server:

```
make server-up
```

Откроем терминал и перейдем в режим суперпользователя

В дополнительном терминале запустим мониторинг системных событий с помощью команды `journalctl -x -f`. С клиента попытаемся получить доступ к серверу посредством SSH-соединения через пользователя root(рис. 3.1):

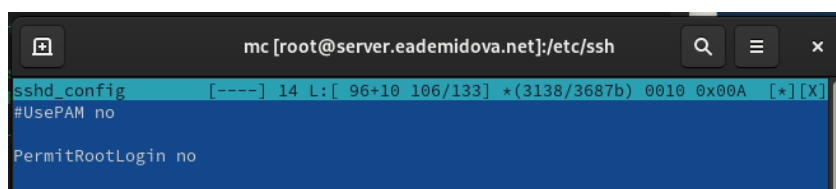


```
root@client:~  
[eademidova@client.eademidova.net ~]$ sudo -i  
[sudo] password for eademidova:  
[root@client.eademidova.net ~]# ssh root@server.eademidova.net  
The authenticity of host 'server.eademidova.net (192.168.1.1)' can't be established.  
ED25519 key fingerprint is SHA256:aRo3GE0og0HcJAWoVgTa8f/gpepKBv1bUgputRwldZE.  
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.eademidova.net' (ED25519) to the list of known hosts.  
root@server.eademidova.net's password:  
Permission denied, please try again.  
root@server.eademidova.net's password:  
Permission denied, please try again.  
root@server.eademidova.net's password:  
root@server.eademidova.net: Permission denied (publickey,gssapi-keyex,gssapi-with-mic,password).
```

Рис. 3.1: Попытка установить SSH-соединение

При попытке соединения, так как мы делаем это первый раз, добавляем сервер в список известных хостов. Затем требуется ввести пароль от пользователя root, но соединение отклоняется.

На сервере откроем файл /etc/ssh/sshd_config конфигурации sshd для редактирования и запретим вход на сервер пользователю root, установив(рис. 3.2):



```
mc [root@server.eademidova.net]:/etc/ssh  
sshd_config  [----] 14 L: [ 96+10 106/133] *(3138/3687b) 0010 0x00A [*] [X]  
#UsePAM no  
  
PermitRootLogin no  
#133
```

Рис. 3.2: Запрет входа на сервер пользователю root

После сохранения изменений в файле конфигурации перезапустим sshd с помощью команды `systemctl restart sshd`. Повторим попытку получения доступа с клиента к серверу посредством SSH-соединения через пользователя root(здесь первое удачное соединение с разрешенным входом на сервер пользователю рут, а второ подключение уже с запрещенным)(3.3):

```

h-mic,password).
[root@client.eademidova.net ~]# ssh root@server.eademidova.net
root@server.eademidova.net's password:
Activate the web console with: systemctl enable --now cockpit.socket

Last failed login: Sat Dec 9 08:01:33 UTC 2023 from 192.168.1.30 on ssh:notty
There were 7 failed login attempts since the last successful login.
[root@server ~]# ^C
[root@server ~]# exit
logout
Connection to server.eademidova.net closed.
[root@client.eademidova.net ~]# ssh root@server.eademidova.net
root@server.eademidova.net's password:
Permission denied, please try again.
root@server.eademidova.net's password:
Permission denied, please try again.
root@server.eademidova.net's password:
root@server.eademidova.net: Permission denied (publickey,gssapi-keyex,gssapi-wit
h-mic,password).
[root@client.eademidova.net ~]#

```

Рис. 3.3: Повторная попытка SSH-соединение

3.2 Ограничение списка пользователей для удалённого доступа по SSH

С клиента попытаемся получить доступ к серверу посредством SSH-соединения через пользователя eademidova(рис. 3.4):

```

logout
[eademidova@client.eademidova.net ~]$ ssh eademidova@server.eademidova.net
The authenticity of host 'server.eademidova.net (192.168.1.1)' can't be establis
hed.
ED25519 key fingerprint is SHA256:aRo3GE0og0HcJAWoVgTa8f/gpepKBv1bUgputRwldZE.
This host key is known by the following other names/addresses:
  ~/.ssh/known_hosts:1: [server.eademidova.net]:2022
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'server.eademidova.net' (ED25519) to the list of know
n hosts.
eademidova@server.eademidova.net's password:
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Sat Dec 9 08:09:41 2023 from 192.168.1.30
[eademidova@server.eademidova.net ~]$

```

Рис. 3.4: Попытка установить SSH-соединение с клиента

Соединение проходит успешно.

На сервере откроем файл /etc/ssh/sshd_config конфигурации sshd на редактирование и добавим строку(3.5):

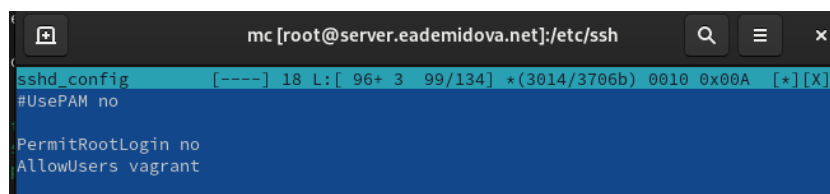


Рис. 3.5: Изменение разрешенных пользователей для sshd

После сохранения изменений в файле конфигурации перезапустим sshd.

Повторим попытку получения доступа с клиента к серверу посредством SSH-соединения через пользователя eademidova(рис. 3.6):

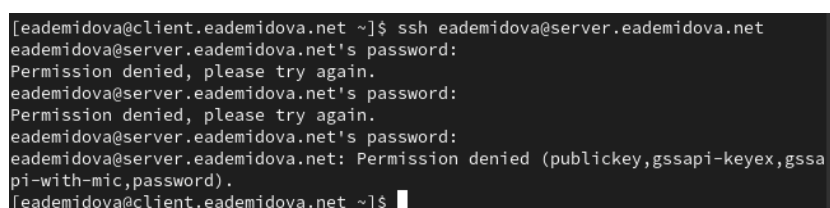


Рис. 3.6: Определение службы аутентификации пользователей

В этот раз соединение не устанавливается, так как в списке разрешенных пользователей нет нашего.

В файле /etc/ssh/sshd_config конфигурации sshd внесем следующее изменение(3.7):

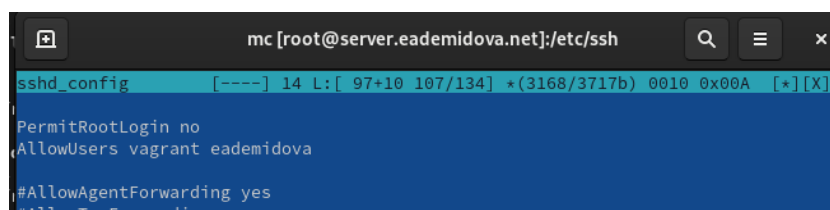


Рис. 3.7: Изменение разрешенных пользователей для sshd

Снова попытаемся установить соединение с клиента к серверу(3.8):

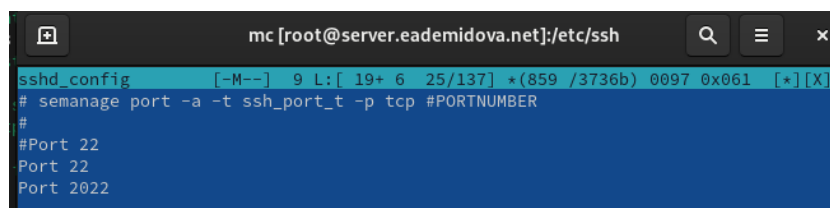
```
pi-with-mic,password).\nleademidova@client.eademidova.net ~]$ ssh eademidova@server.eademidova.net\neademidova@server.eademidova.net's password:\nActivate the web console with: systemctl enable --now cockpit.socket\n\nLast failed login: Sat Dec 9 08:13:13 UTC 2023 from 192.168.1.30 on ssh:notty\nThere were 3 failed login attempts since the last successful login.\nLast login: Sat Dec 9 08:11:11 2023 from 192.168.1.30\nleademidova@server.eademidova.net ~]$
```

Рис. 3.8: Временный запуск SMTP-сервера

В этот раз доступ получен.

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(3.9):



```
mc [root@server.eademidova.net]:/etc/ssh\nsshd_config [-M--] 9 L: [ 19+ 6 25/137] *(859 /3736b) 0097 0x061 [*][X]\n# semanage port -a -t ssh_port_t -p tcp #PORTNUMBER\n#\n#Port 22\nPort 22\nPort 2022
```

Рис. 3.9: Добавление портов в файл конфигураций

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`.

Посмотрим расширенный статус работы `sshd`(3.9):

```
Support: https://access.redhat.com/support
root@server:/etc/ssh

[root@server.eademidova.net ssh]# mc

[root@server.eademidova.net ssh]# systemctl restart sshd
[root@server.eademidova.net ssh]# systemctl status -l sshd
● sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; preset: ena
   Active: active (running) since Sat 2023-12-09 08:16:56 UTC; 13s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
    Main PID: 7726 (sshd)
      Tasks: 1 (limit: 5724)
     Memory: 1.4M
        CPU: 10ms
    CGroup: /system.slice/ssh.service
            └─7726 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Dec 09 08:16:56 server.eademidova.net systemd[1]: Starting OpenSSH server daemo
Dec 09 08:16:56 server.eademidova.net sshd[7726]: main: sshd: ssh-rsa algorithm
Dec 09 08:16:56 server.eademidova.net sshd[7726]: error: Bind to port 2022 on 0
Dec 09 08:16:56 server.eademidova.net sshd[7726]: error: Bind to port 2022 on :
Dec 09 08:16:56 server.eademidova.net sshd[7726]: Server listening on 0.0.0.0 p
Dec 09 08:16:56 server.eademidova.net sshd[7726]: Server listening on :: port 2
Dec 09 08:16:56 server.eademidova.net systemd[1]: Started OpenSSH server daemon.
lines 1-19/19 (END)
```

Рис. 3.10: Расширенный статус работы sshd

Система сообщает об отказе в работе sshd через порт 2022. Дополнительно посмотрим сообщения в терминале с мониторингом системных событий(рис. 3.11):

```
The job identifier is 3466.
Dec 09 08:16:57 server.eademidova.net systemd[1]: Started dbus-:1.1-org.fedoraproject.Setroubles
hootPrivileged@1.service.
  Subject: A start job for unit dbus-:1.1-org.fedoraproject.SetroubleshootPrivileged@1.service
has finished successfully
   Defined-By: systemd
   Support: https://access.redhat.com/support

A start job for unit dbus-:1.1-org.fedoraproject.SetroubleshootPrivileged@1.service has finis
hed successfully.

The job identifier is 3542.
Dec 09 08:16:59 server.eademidova.net setroubleshoot[7727]: SELinux is preventing /usr/sbin/sshd
from name_bind access on the tcp_socket port 2022. For complete SELinux messages run: sealert -
l 74c0e7ad-d686-4d3a-8d2b-7d7c3372df2c
Dec 09 08:16:59 server.eademidova.net setroubleshoot[7727]: SELinux is preventing /usr/sbin/sshd
from name_bind access on the tcp_socket port 2022.

***** Plugin bind_ports (92.2 confi
dence) suggests *****
```

Рис. 3.11: Мониторинг системных сообщений

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

Исправим на сервере метки SELinux к порту 2022 и в настройках межсетевого экрана откроем порт 2022 протокола. Вновь перезапустим sshd и посмотрите

расширенный статус его работы. Статус показывает, что процесс sshd теперь прослушивает два порта(3.12)

```
root@server:/etc/ssh
^C
[root@server.eademidova.net ssh]# semanage port -a -t ssh_port_t -p tcp 2022
[root@server.eademidova.net ssh]# firewall-cmd --add-port=2022/tcp
success
[root@server.eademidova.net ssh]# firewall-cmd --add-port=2022/tcp --permanent
success
[root@server.eademidova.net ssh]# systemctl restart sshd
[root@server.eademidova.net ssh]# systemctl status -l sshd
● sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; preset: enabled)
   Active: active (running) since Sat 2023-12-09 08:19:26 UTC; 32s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Main PID: 7768 (sshd)
    Tasks: 1 (limit: 5724)
   Memory: 1.8M
      CPU: 10ms
   CGroup: /system.slice/sshd.service
           └─7768 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Dec 09 08:19:25 server.eademidova.net systemd[1]: Starting OpenSSH server daemon...
Dec 09 08:19:26 server.eademidova.net sshd[7768]: main: sshd: ssh-rsa algorithm is disabled
Dec 09 08:19:26 server.eademidova.net sshd[7768]: Server listening on 0.0.0.0 port 2022.
Dec 09 08:19:26 server.eademidova.net sshd[7768]: Server listening on :: port 2022.
Dec 09 08:19:26 server.eademidova.net sshd[7768]: Server listening on 0.0.0.0 port 22.
Dec 09 08:19:26 server.eademidova.net sshd[7768]: Server listening on :: port 22.
Dec 09 08:19:26 server.eademidova.net systemd[1]: Started OpenSSH server daemon.
[root@server.eademidova.net ssh]#
```

Рис. 3.12: Просмотр расширенного статуса работы sshd после настройки работы по порту 2022

С клиента попытаемся получить доступ к серверу посредством SSH-соединения через пользователя eademidova обычным способом и указав порт 2022(рис. 3.13):

```
Connection to server.eademidova.net closed.
[eademidova@client.eademidova.net ~]$ ssh eademidova@server.eademidova.net
eademidova@server.eademidova.net's password:
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Sat Dec  9 08:14:53 2023 from 192.168.1.30
[eademidova@server.eademidova.net ~]$ sudo -i
[sudo] password for eademidova:
[root@server.eademidova.net ~]# exit
logout
[eademidova@server.eademidova.net ~]$ exit
logout
Connection to server.eademidova.net closed.
[eademidova@client.eademidova.net ~]$ ssh -p2022 eademidova@server.eademidova.net
eademidova@server.eademidova.net's password:
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Sat Dec  9 08:21:23 2023 from 192.168.1.30
[eademidova@server.eademidova.net ~]$ sudo -i
[sudo] password for eademidova:
[root@server.eademidova.net ~]#
```

Рис. 3.13: Установка SSH-соединение с клиента

3.4 Настройка удалённого доступа по SSH по ключу

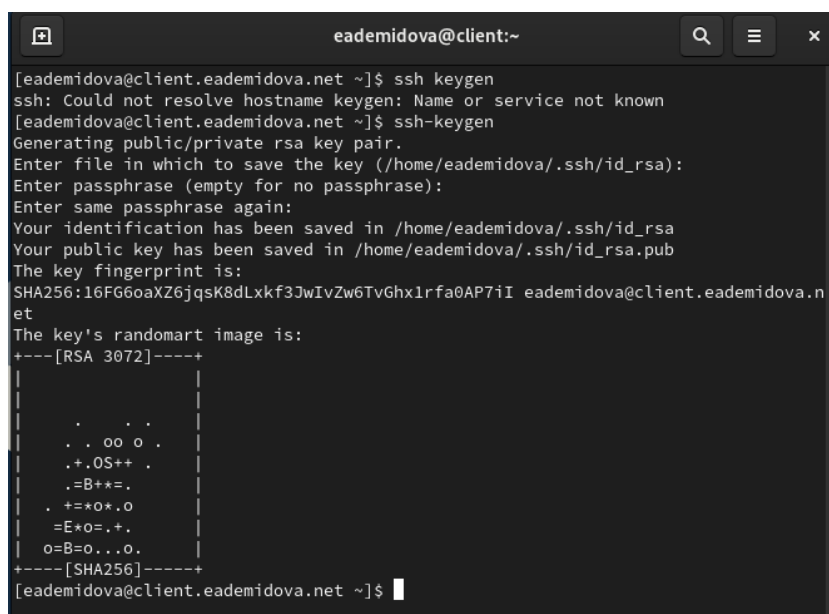
Создадим пару из открытого и закрытого ключей для входа на сервер.

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

```
PubkeyAuthentication yes
```

После сохранения изменений в файле конфигурации перезапустим `sshd`.

На клиенте сформируем SSH-ключ, введя в терминале(3.14):



```
eademidova@client:~  
[eademidova@client.eademidova.net ~]$ ssh keygen  
ssh: Could not resolve hostname keygen: Name or service not known  
[eademidova@client.eademidova.net ~]$ ssh-keygen  
Generating public/private rsa key pair.  
Enter file in which to save the key (/home/eademidova/.ssh/id_rsa):  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /home/eademidova/.ssh/id_rsa  
Your public key has been saved in /home/eademidova/.ssh/id_rsa.pub  
The key fingerprint is:  
SHA256:16FG6oaXZ6jqsK8dLxkf3JwIvZw6TvGhx1rfa0AP7iI eademidova@client.eademidova.net  
The key's randomart image is:  
+---[RSA 3072]-----+  
|  
| . . . . .  
| . . oo o .  
| .+.OS++ .  
| .=B+*=.  
| . +=*o*.o  
| =E*o=,+.  
| o=B=o...o.  
+---[SHA256]-----+  
[eademidova@client.eademidova.net ~]$
```

Рис. 3.14: Формирования SSH-ключа на клиенте

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 user@server.user.net
```

Попробуем получить доступ с клиента к серверу посредством SSH-соединения(3.15):

```
eademidova@server:~  
[eademidova@client ~]$ ssh eademidova@server.eademidova.net  
eademidova@server.eademidova.net's password:  
  
[eademidova@client ~]$ mc  
  
[eademidova@client .ssh]$ ssh-copy-id eademidova@server.eademidova.net  
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter  
out any that are already installed  
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompt  
ed now it is to install the new keys  
eademidova@server.eademidova.net's password:  
  
Number of key(s) added: 1  
  
Now try logging into the machine, with: "ssh 'eademidova@server.eademidova.net  
'"  
and check to make sure that only the key(s) you wanted were added.  
  
[eademidova@client .ssh]$ ssh eademidova@server.eademidova.net  
Activate the web console with: systemctl enable --now cockpit.socket  
  
Last login: Sat Dec  9 09:26:50 2023  
[eademidova@server.eademidova.net ~]$
```

Рис. 3.15: Установка SSH-соединения с сервером с клиента

3.5 Организация туннелей SSH, перенаправление TCP-портов

На клиенте посмотрим, запущены ли какие-то службы с протоколом TCP, на данный момент их нет. Перенаправим порт 80 на server.eademidova.net на порт 8080 на локальной машине и вновь на клиенте посмотрим, запущены ли какие-то службы с протоколом TCP(рис. 3.16)

```
eademidova@client:~  
[eademidova@client.eademidova.net ~]$ lsof | grep TCP  
[eademidova@client.eademidova.net ~]$ ssh -fNL 8080:localhost:80 eademidova@server.eademidova.net  
[eademidova@client.eademidova.net ~]$ lsof | grep TCP  
ssh          7020      eademidova    3u  IPv4        70939      0t0      TCP  
client.eademidova.net:37678->mail.eademidova.net:ssh (ESTABLISHED)  
ssh          7020      eademidova    4u  IPv6        70959      0t0      TCP  
localhost:webcache (LISTEN)  
ssh          7020      eademidova    5u  IPv4        70960      0t0      TCP  
localhost:webcache (LISTEN)  
[eademidova@client.eademidova.net ~]$
```

Рис. 3.16: Просмотр активных служб с протоколом TCP

Появилось три службы, использующие TCP протокол – появился доступ к mail.eademidova.net по ssh, а также к локальному хосту по IPv4 и IPv6.

На клиенте запустим браузер и в адресной строке введем localhost:8080. Отображается страница с приветствием «Welcome to the server.eademidova.net server»(3.17):

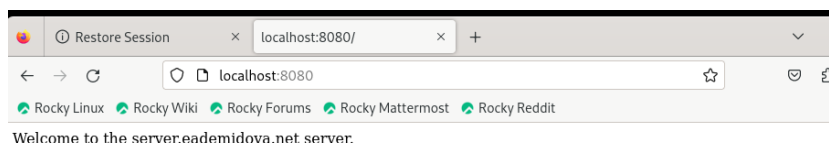


Рис. 3.17: Просмотр локального сервера в браузере на клиенте

3.6 Запуск консольных приложений через SSH

На клиенте откроем терминал под пользователем eademidova и посмотрим с клиента имя узла сервера, файлов на сервере и почту(рис. 3.18):

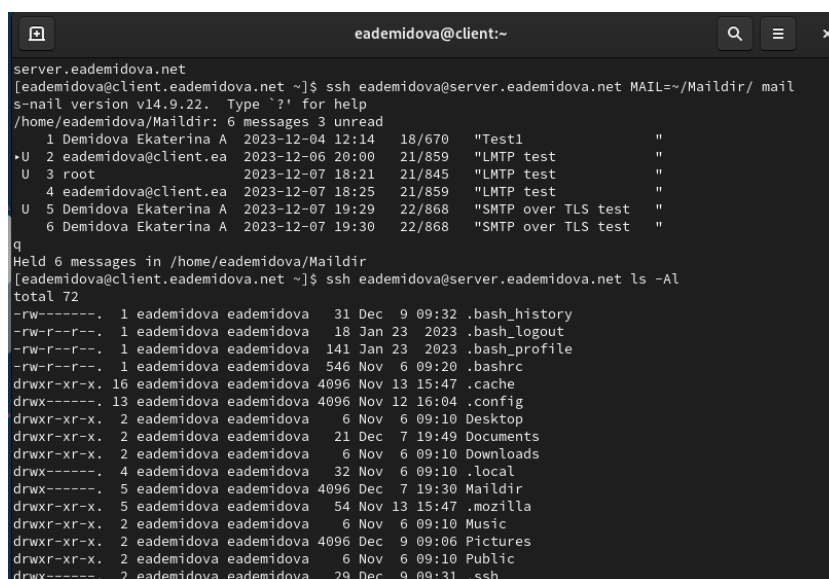


Рис. 3.18: Просмотр информации о сервере с клиента через ssh

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, прописав:

```
X11Forwarding yes
```

После сохранения изменения в конфигурационном файле перезапустим `sshd`. Попробуем с клиента удалённо подключиться к серверу и запустить графическое приложение `firefox` (рис. 3.19):

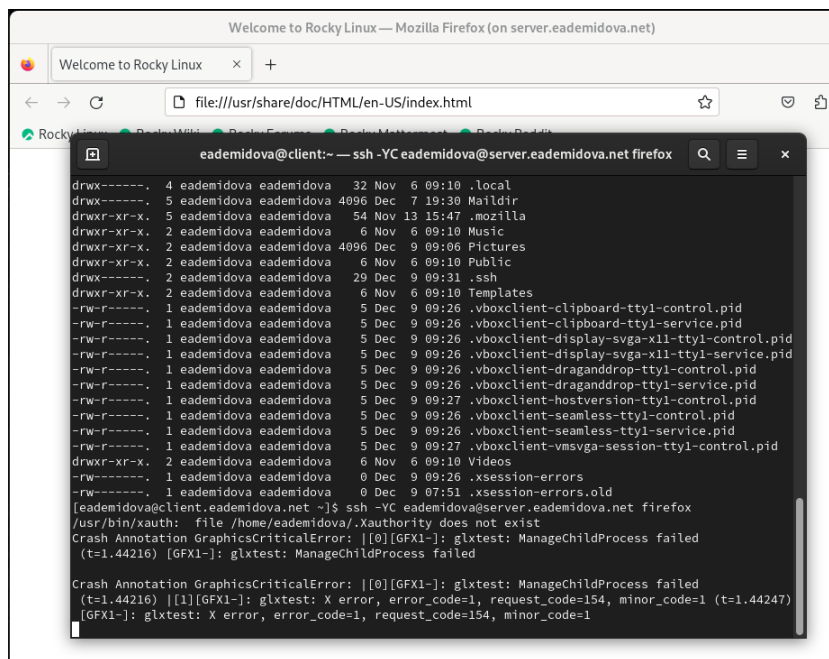
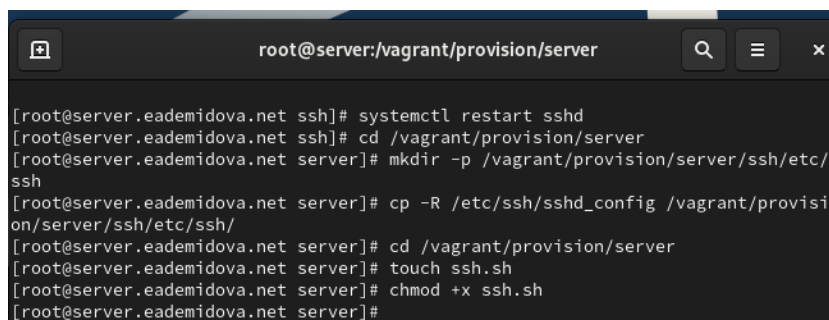


Рис. 3.19: Запуск графического приложения через ssh

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` и в каталоге `/vagrant/provision/server` создадим исполняемый файл `ssh.sh` (рис. 3.20)

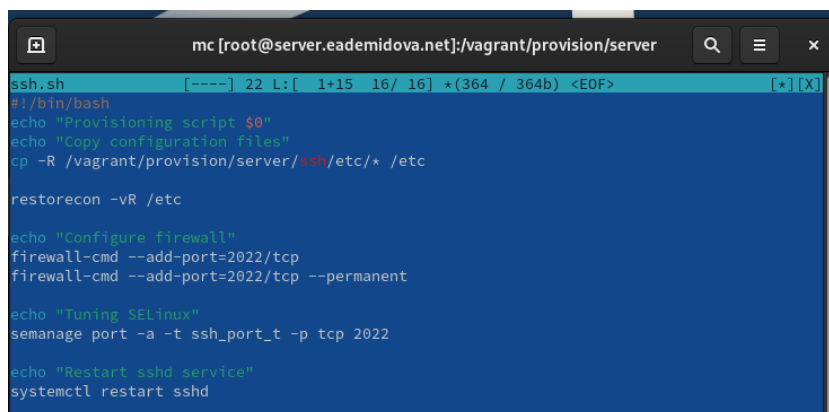


```
root@server:/vagrant/provision/server

[root@server.eademidova.net ssh]# systemctl restart sshd
[root@server.eademidova.net ssh]# cd /vagrant/provision/server
[root@server.eademidova.net server]# mkdir -p /vagrant/provision/server/ssh/etc/ssh
[root@server.eademidova.net server]# cp -R /etc/ssh/sshd_config /vagrant/provision/server/ssh/etc/ssh/
[root@server.eademidova.net server]# cd /vagrant/provision/server
[root@server.eademidova.net server]# touch ssh.sh
[root@server.eademidova.net server]# chmod +x ssh.sh
[root@server.eademidova.net server]#
```

Рис. 3.20: Создание окружения для внесения изменений в настройки окружающей среды

Пропишем скрипт в `/vagrant/provision/server/ssh.sh` (3.21):



```
mc [root@server.eademidova.net]:/vagrant/provision/server

ssh.sh  [-----] 22 L: [ 1+15 16/ 16] *(364 / 364b) <EOF>  [*][X]
#!/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
```

Рис. 3.21: Скрипта файла `/vagrant/provision/server/ssh.sh`

Для отработки созданного скрипта во время загрузки виртуальной машины

server в конфигурационном файле Vagrantfile добавим следующую запись в разделе конфигурации для сервера(3.22):



The image shows a code editor window titled '*Vagrantfile' with the path '/var/tmp/eademidova/vagrant'. The editor contains a Vagrantfile configuration snippet. Line 78 is highlighted. The configuration includes two provisioners for the 'server' VM: 'server_mail' and 'server_ssh'. Both are of type 'shell' and have 'preserve_order: true'. The 'server_ssh' path is 'provision/server/ssh.sh'. The snippet ends with 'end' on line 87.

```
78     path: "provision/server/firewall.sh"
79   server.vm.provision "server_mail",
80     type: "shell",
81     preserve_order: true,
82     path: "provision/server/mail.sh"
83   server.vm.provision "server_ssh",
84     type: "shell",
85     preserve_order: true,
86     path: "provision/server/ssh.sh"
87   end
88 end
```

Рис. 3.22: Изменение конфигурационного файла Vagrant

4 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?
7. В файле /etc/ssh/sshd_config конфигурации прописать PermitRootLogin no и AllowUsers alice.
8. Для настройки удалённого доступа по SSH через несколько портов нужно отредактировать файл конфигурации SSH (/etc/ssh/sshd_config) и добавить строку Port <порт>.
9. Для установки фонового соединения без команды используется параметр -N при использовании команды ssh: ssh -N <hostname>.

10. Для настройки локальной переадресации с локального порта 5555 на порт 80 сервера `server2.example.com` следует использовать команду: `ssh -fNL 80:localhost:55555 server2.example.com`.
11. Для настройки SELinux и разрешения SSH связываться с портом 2022 можно использовать команду: `semanage port -a -t ssh_port_t -p tcp 2022`.
12. Для разрешения входящих подключений по SSH через порт 2022 с помощью межсетевого экрана следует использовать команду: `firewall-cmd --add-port=2022/tcp --permanent`.

5 Выводы

В результате выполнения данной работы были приобретены практические навыки по настройке удалённого доступа к серверу с помощью SSH.