Отчет по лабораторной работе №11

Дисциплина: Администрирование сетевых подсистем

Иванов Сергей Владимирович

Содержание

Список иллюстраций

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 (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).

3 Выполнение лабораторной работы

3.1 Запрет удалённого доступа по SSH для пользователя root

На сервере в дополнительном терминале запустим мониторинг системных событий:

sudo -i

journalctl -x -f (рис. 1).

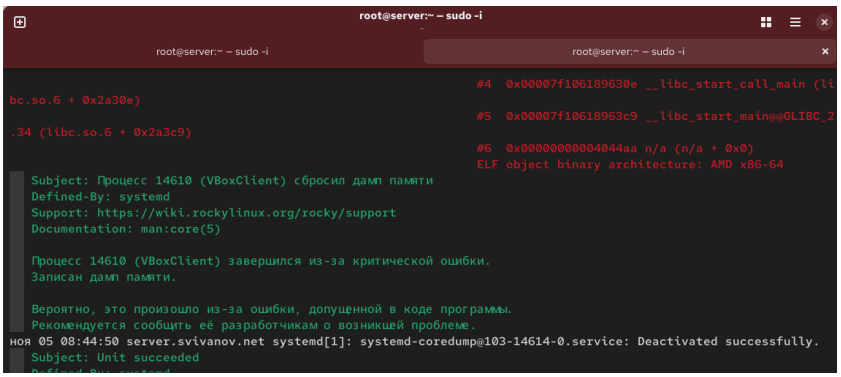

Рис. 3.1: Запуск мониторинга

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

Подключение не удалось. Это может быть связано с тем, что вход под root уже был где-то запрещен. (рис. 2).

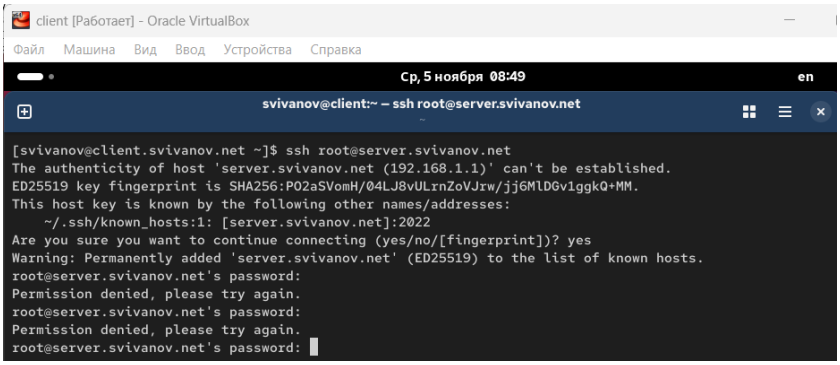

Рис. 3.2: Попытка получить доступ к серверу по SSH

На сервере откроем файл /etc/ssh/sshd_config конфигурации sshd для редактирования и запретим вход на сервер пользователю root, установив: PermitRootLogin no (рис. 3)

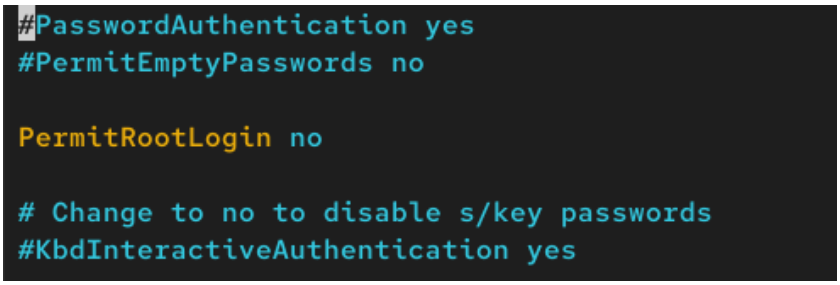

Рис. 3.3: Запрет входа для root

После сохранения изменений в файле конфигурации перезапустим sshd: systemctl restart sshd (рис. 4)

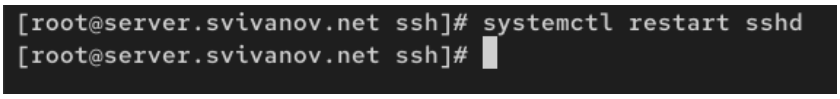

Рис. 3.4: Перезапуск sshd

Повторим попытку получения доступа с клиента к серверу посредством SSH-соединения через пользователя root: ssh root@server. Подключение не удалось. Теперь точно знаем, что это из-за того, что мы запретили доступ пользователю root. (рис. 5)

```
[svivanov@client.svivanov.net ~]$ ssh root@server.svivanov.net
root@server.svivanov.net's password:
Permission denied, please try again.
root@server.svivanov.net's password:
Permission denied, please try again.
root@server.svivanov.net's password:
Connection closed by 192.168.1.1 port 22
[svivanov@client.svivanov.net ~]$
[svivanov@client.svivanov.net ~]$
[svivanov@client.svivanov.net ~]$ ssh root@server.svivanov.net
kex_exchange_identification: read: Connection reset by peer
Connection reset by 192.168.1.1 port 22
```

Рис. 3.5: Попытка подключения

3.2 Ограничение списка пользователей для удалённого доступа по SSH

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

Подключение удалось, нам вывелось дата и время последнего захода и консоль сервера. (рис. 6)

```
[svivanov@client.svivanov.net ~]$ ssh svivanov@server.svivanov.net
svivanov@server.svivanov.net's password:
Web console: https://server.svivanov.net:9090/ or https://192.168.1.1:9090/

Last login: Wed Nov  5 08:36:21 2025
[svivanov@server.svivanov.net ~]$
```

Рис. 3.6: Попытка подключения юзера к серверу

На сервере откроем файл /etc/ssh/sshd_config конфигурации sshd на редактирование и добавим строку AllowUsers vagrant (рис. 7)

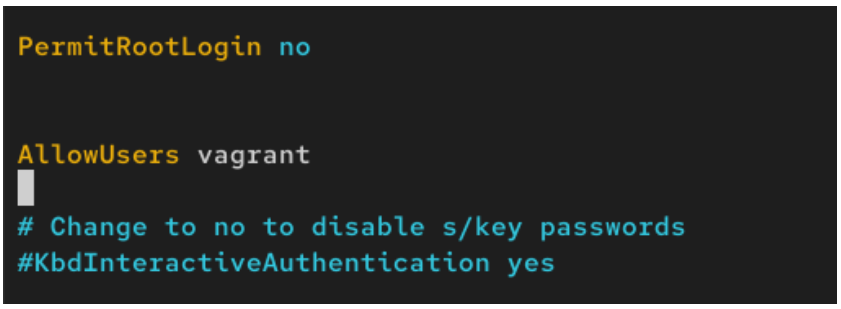

Рис. 3.7: Редактирование sshd_config

После сохранения изменений в файле конфигурации перезапустим sshd: systemctl restart sshd. Повторим попытку получения доступа с клиента к серверу посредством SSH-соединения через пользователя user: ssh user@server.user.net.

Подключение не удалось, так как мы разрешили подключаться только пользователю vagrant в конф. файле. (рис. 8)

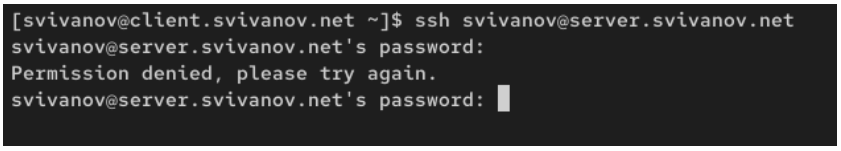

Рис. 3.8: Попытка подключения к серверу

В файле /etc/ssh/sshd_config конфигурации sshd внесем следующее изменение: AllowUsers vagrant svivanov (рис. 9)

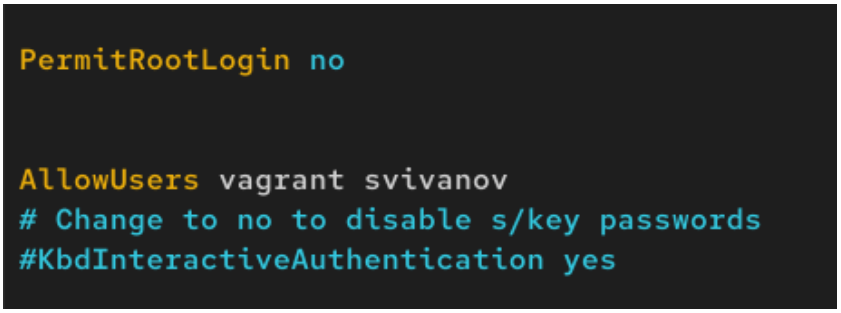

Рис. 3.9: Редактирование sshd_config

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

Подключение удалось, так как мы разрешили подключение пользователю svivanov в кофн. файле. (рис. 10)

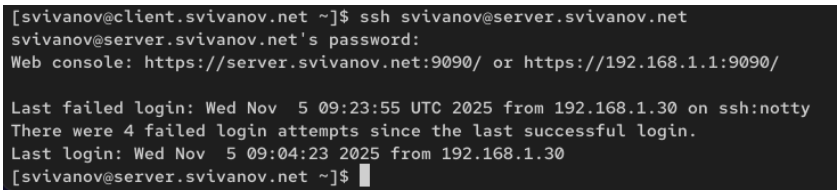

Рис. 3.10: Попытка подключения к серверу

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 (рис. 11)

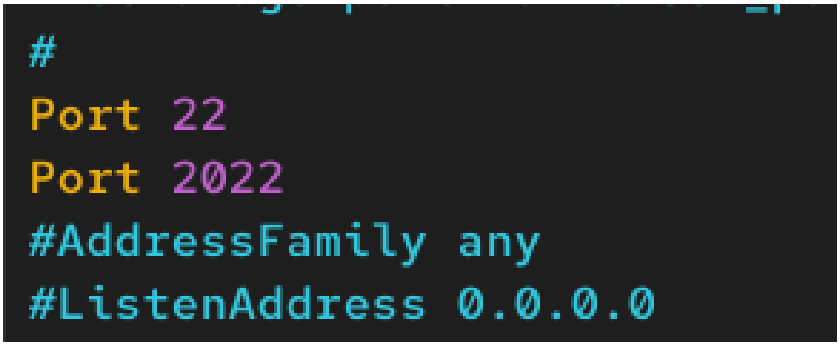

Рис. 3.11: Редактирование sshd_config

После сохранения изменений в файле конфигурации перезапустим sshd: systemctl restart sshd. Посмотрим расширенный статус работы sshd: systemctl status -l sshd. Система сообщает об отказе в работе sshd через порт 2022. (рис. 12)

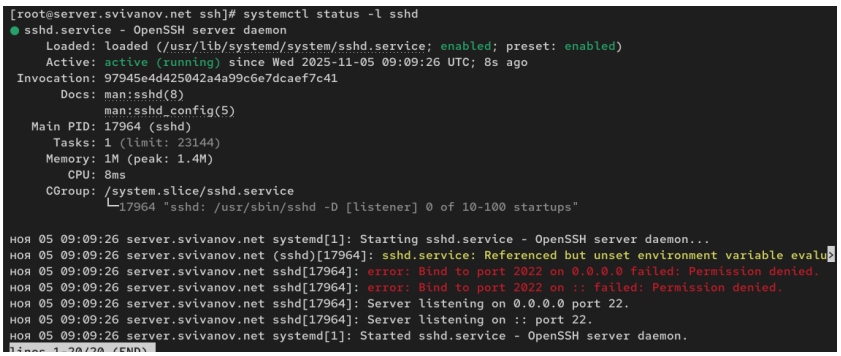

Рис. 3.12: Перезапуск и просмотр статуса службы

Дополнительно посмотрим сообщения в терминале с мониторингом системных событий.

На терминале с мониторингом системных событий я нашел только сообщения, связанные с службой named (DNS-сервер). Скорее всего я пролистал необходимые сообщения связанные с SSH-сервером, так как сообщения в мониторинге выходят очень быстро и часто. Там должно было быть что-то вроде error: Bind to port 2022 failed. Permission denied; как на предыдущем скриншоте (рис. 12), где видно часть мониторинга связанного со службой sshd. (рис. 13)

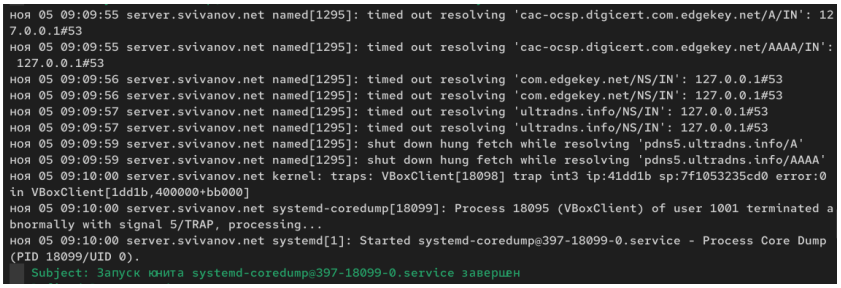

Рис. 3.13: Просмотр мониторинга системных событий

Исправим на сервере метки SELinux к порту 2022: semanage port -a -t ssh_port_t -p tcp 2022 (рис. 14)

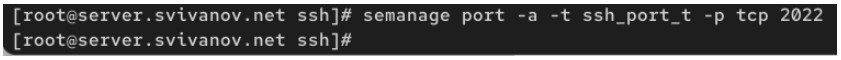

Рис. 3.14: Исправление меток SELinux

В настройках межсетевого экрана откроем порт 2022 протокола TCP:

firewall-cmd –add-port=2022/tcp

firewall-cmd –add-port=2022/tcp –permanent (рис. 15)

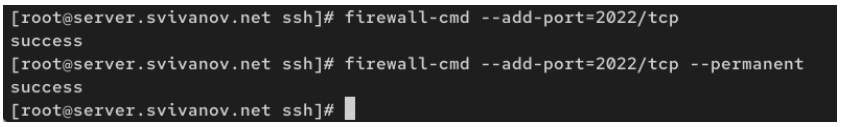

Рис. 3.15: Настройка firewall

Вновь перезапустим sshd и посмотрим расширенный статус его работы. Статус показывает, что процесс sshd теперь прослушивает два порта (22 и 2022). (рис. 16)

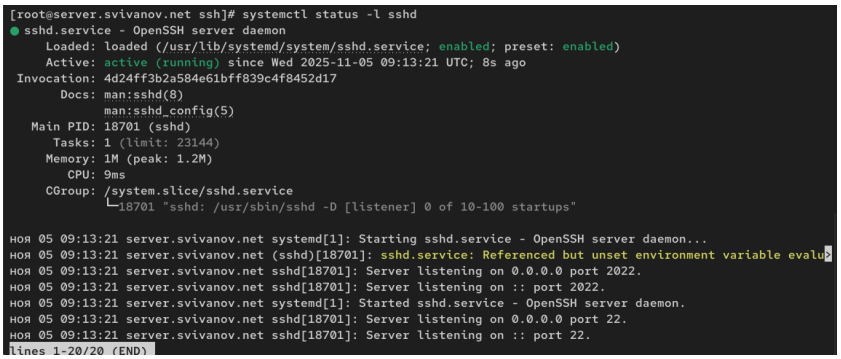

Рис. 3.16: Перезауск службы и просмотр статуса

С клиента попытаемся получить доступ к серверу посредством SSH-соединения через пользователя:

ssh user@server.user.net.

После открытия оболочки пользователя введем sudo -i для получения доступа root. Отлогинимся от root и пользователя на сервере, введя дважды logout. (рис. 17)

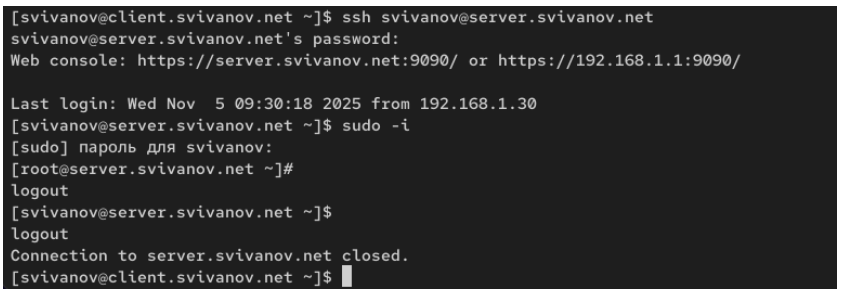

Рис. 3.17: Подключение к серверу, получение root

С клиента попытаемся получить доступ к серверу посредством SSH-соединения через пользователя, указав порт 2022:

ssh user@server.user.net.

После открытия оболочки пользователя введем sudo -i для получения доступа root. Отлогинимся от root и пользователя на сервере, введя дважды logout. (рис. 18)

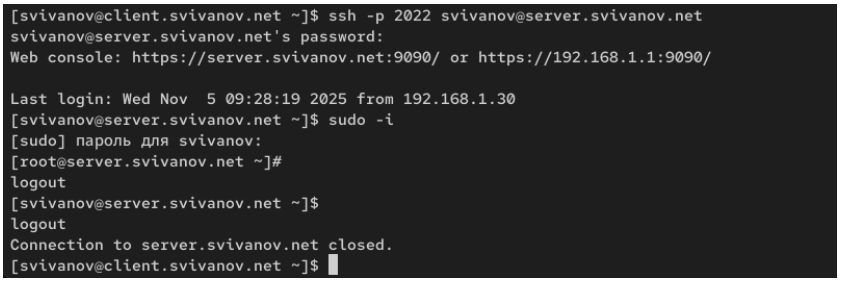

Рис. 3.18: Подключение к серверу через порт 2022, получение root

3.4 Настройка удалённого доступа по SSH по ключу

На сервере в конфигурационном файле /etc/ssh/sshd_config зададим параметр, разрешающий аутентификацию по ключу: PubkeyAuthentication yes. (рис. 19)

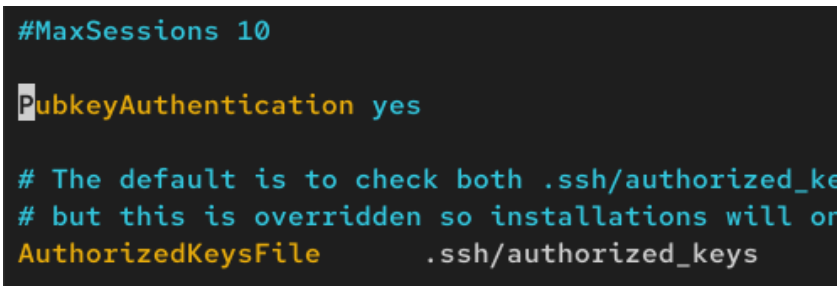

Рис. 3.19: Редактирование sshd_config

После сохранения изменений в файле конфигурации перезапустим sshd. На клиенте сформируем SSH-ключ, введя в терминале под пользователем: ssh-keygen (рис. 20)

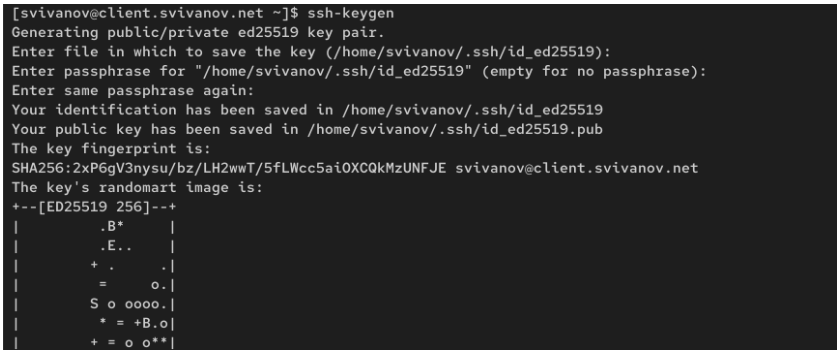

Рис. 3.20: Создание SSH ключа

Скопируем открытый ключ на сервер, введя на клиенте: ssh-copy-id user@server.user.net (рис. 21)

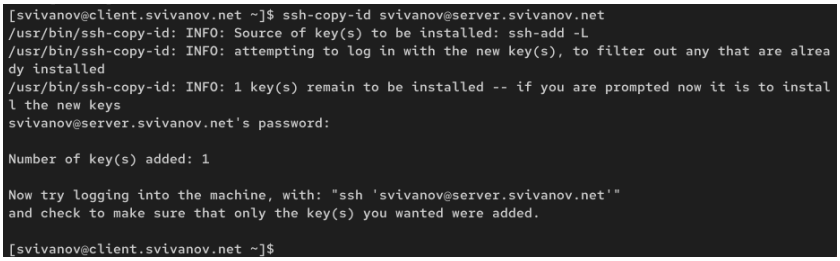

Рис. 3.21: Копирование ключа на сервер

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

Теперь мы прошли аутентификацию без ввода пароля для учётной записи удалённого пользователя. Отлогимся с сервера, используя комбинацию клавиш Ctrl + d. (рис. 22)

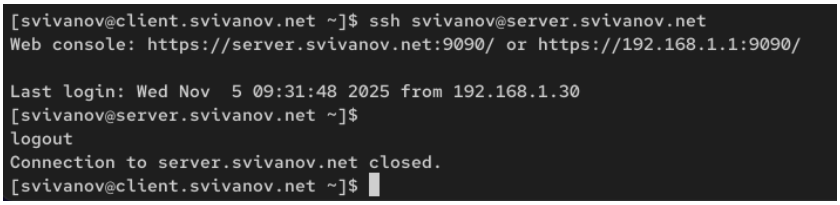

Рис. 3.22: Подключение к серверу

3.5 Организация туннелей SSH, перенаправление TCP-портов

На клиенте посмотрим, запущены ли какие-то службы с протоколом TCP: lsof | grep TCP. Видим, что запущены. (рис. 23)

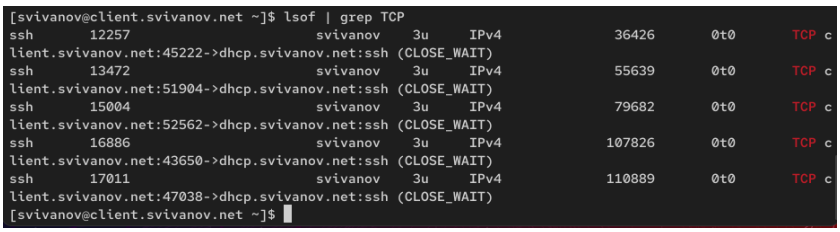

Рис. 3.23: Просмотр служб с протоколом TCP

Перенаправим порт 80 на server.user.net на порт 8080 на локальной машине: ssh -fNL 8080:localhost:80 user@server.user.net (рис. 24)

Рис. 3.24: Перенаправление порта

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

Вижу, что появилось 2 новых процесса, которые слушают на localhost:webcache. (рис. 25)

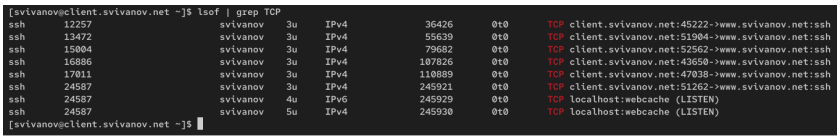

Рис. 3.25: Просмотр служб с протоколом TCP

На клиенте запустим браузер и в адресной строке введем localhost:8080. Убедимся, что отобразится страница с приветствием «Welcome to the server.user.net server». (рис. 26)

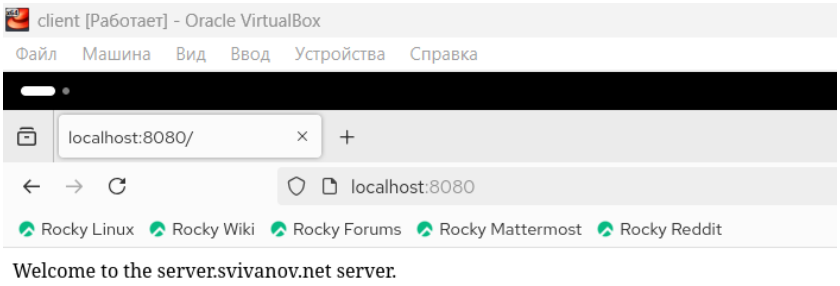

Рис. 3.26: Страница с приветствием в браузере

3.6 Запуск консольных приложений через SSH

На клиенте откроем терминал под пользователем. Посмотрим с клиента имя узла сервера:

ssh user@server.user.net hostname

Посмотрим с клиента список файлов на сервере:

ssh user@server.user.net ls -Al (рис. 27)

```
[svivanov@client.svivanov.net ~]$ ssh svivanov@server.svivanov.net hostname
server.svivanov.net
[svivanov@client.svivanov.net ~]$ ssh svivanov@server.svivanov.net ls -Al
total 64
drwxr-xr-x.  2 svivanov svivanov 4096 сен 22 14:47 aus
-rw-------.  1 svivanov svivanov  800 ноя  5 09:31 .bash_history
-rw-r--r--.  1 svivanov svivanov   18 окт 29  2024 .bash_logout
-rw-r--r--.  1 svivanov svivanov  144 окт 29  2024 .bash_profile
-rw-r--r--.  1 svivanov svivanov  603 сен  4 13:30 .bashrc
drwx------. 12 svivanov svivanov 4096 сен  9 09:35 .cache
drwx------. 13 svivanov svivanov 4096 сен 22 14:45 .config
drwx------.  4 svivanov svivanov   32 сен  4 13:06 .local
drwx------.  5 svivanov svivanov 4096 окт 27 11:30 Maildir
drwxr-xr-x.  5 svivanov svivanov   54 сен  9 09:35 .mozilla
drwx------.  2 svivanov svivanov   29 ноя  5 10:20 .ssh
```

Рис. 3.27: Просмотр файлов на сервере

Посмотрим с клиента почту на сервере: ssh user@server.user.net MAIL=~/Maildir/ mail. (рис. 28)

```
[svivanov@client.svivanov.net ~]$ ssh svivanov@server.svivanov.net MAIL=~/Maildir/ mail
s-nail version v14.9.24.  Type `?' for help
/home/svivanov/Maildir: 11 messages 4 deleted
▸   1 Sergey Ivanov         2025-10-18 08:40   18/674   "Тест 2              "
    2 Sergey Ivanov         2025-10-18 08:39   19/782   "Тестирование        "
    3 Sergey Ivanov         2025-10-18 08:46   18/648   "Тест 4              "
    4 Sergey Ivanov         2025-10-18 09:06   18/628   "Test 5              "
    5 svivanov@client.sviv  2025-10-25 09:29   21/803   "LMTP test           "
    6 Sergey Ivanov         2025-10-25 15:52   22/791   "efwef               "
   11 Sergey Ivanov         2025-10-27 11:30   22/823   "uifijewhiewlighierhig"
```

Рис. 3.28: Просмотр почты на сервере

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. (рис. 29)

```
#GatewayPorts no
X11Forwarding yes
#X11DisplayOffset 10
```

Рис. 3.29: Редактирование sshd_config

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.user.net firefox. (рис. 30)

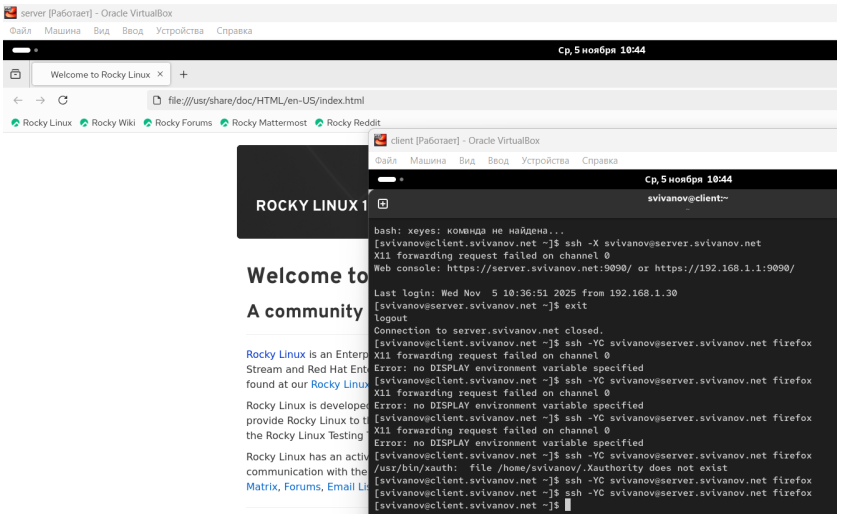

Рис. 3.30: Запуск браузера на сервере через клиента

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/. В соответствующие подкаталоги поместим конфигурационные файлы sshd_config: (рис. 31)

```
[root@server.svivanov.net ssh]# cd /vagrant/provision/server
[root@server.svivanov.net server]# mkdir -p /vagrant/provision/server/ssh/etc/ssh
[root@server.svivanov.net server]# cp -R /etc/ssh/sshd_config /vagrant/provision/server/ssh/etc/ssh/
[root@server.svivanov.net server]#
```

Рис. 3.31: Замена конф. файлов

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: (рис. 32)

```
#!/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. 3.32: Создание скрипта

Для отработки созданного скрипта во время загрузки виртуальной машины server в конфигурационном файле Vagrantfile необходимо добавить в разделе конфигурации для сервера. (рис. 33)

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

Рис. 3.33: Коррекция vagrantfile

4 Ответы на контрольные вопросы

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

В файле /etc/ssh/sshd_config:

PermitRootLogin no

AllowUsers alice

После изменений выполнить:

systemctl restart sshd

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

В файле /etc/ssh/sshd_config:

Port 22 Port 2022

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

ssh -f -N -L локальный_порт:целевой_хост:целевой_порт пользователь@ssh_сервер

Опции:

-f — переход в фоновый режим

-N — не выполнять удалённую команду

-L — локальное перенаправление портов

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

ssh -L 5555:server2.example.com:80 пользователь@ssh_шлюз

Или если server2 доступен напрямую:

ssh -L 5555:localhost:80 пользователь@server2.example.com

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

semanage port -a -t ssh_port_t -p tcp 2022

Проверка текущих разрешённых портов

semanage port -l | grep ssh

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

firewall-cmd –add-port=2022/tcp

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

firewall-cmd –reload

5 Выводы

В ходе выполнения лабораторной работы мы приобрели практические навыки по настройке удалённого доступа к серверу с помощью SSH.