Отчёт по лабораторной работе 11

Настройка безопасного удалённого доступа по протоколу SSH

Сейдалиев Тагиетдин Ровшенович

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

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

2 Выполнение

2.1 Запрет удалённого доступа по SSH для пользователя root

С клиента выполняется попытка подключения к серверу от имени root.
На экране отображается запрос пароля, однако даже при корректном вводе система не допускает пользователя к удалённому доступу.
SSH-сервер отклоняет аутентификацию, что видно на скриншоте ниже.

Рис. 2.1: Попытка подключения под пользователем root

Такое поведение является ожидаемым: по умолчанию в конфигурации SSH вход root-пользователя запрещён.
В системном журнале на сервере фиксируются неоднократные отказанные попытки входа.

В файле конфигурации sshd присутствует параметр, запрещающий удалённый вход root.

Это подтверждается соответствующей строкой, представленной на скриншоте.

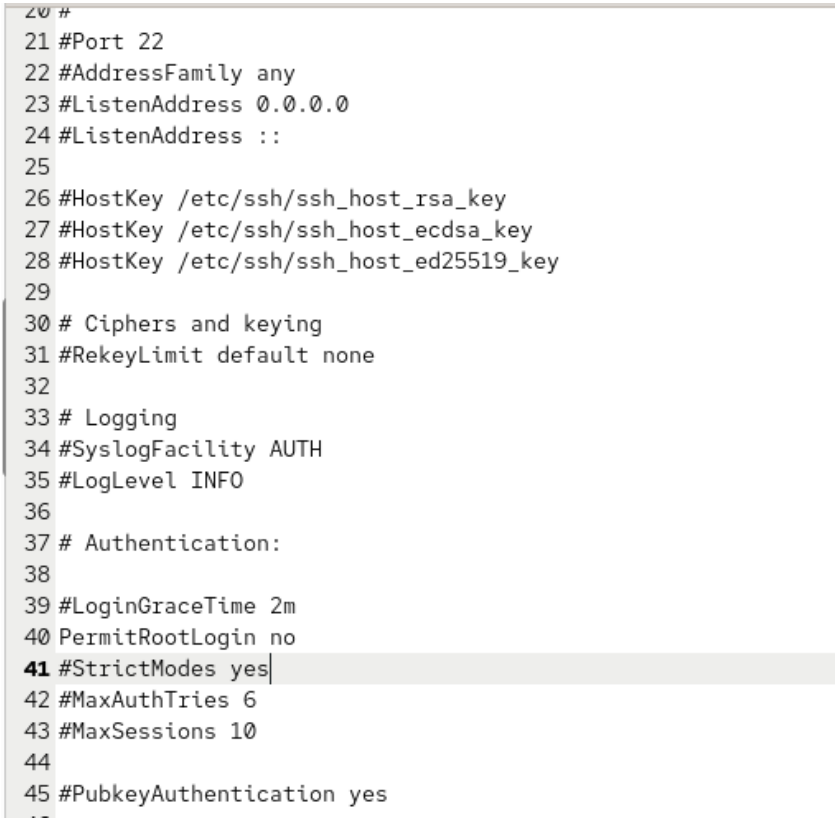

Рис. 2.2: Фрагмент sshd_config с запретом root

После сохранения конфигурации служба SSH перезапускается для применения изменений.

После перезапуска SSH-сервера попытка входа под пользователем root вновь завершается отказом.

Сервер не принимает пароль и завершает процедуру аутентификации.

Рис. 2.3: Отказ в доступе после перезапуска SSH

Удалённый доступ под root заблокирован политикой безопасности, и данное поведение является корректным.

2.2 Ограничение списка пользователей для SSH

Выполняется доступ к серверу под учётной записью trseidaliev.
Аутентификация проходит успешно, и пользователь получает доступ к системе.

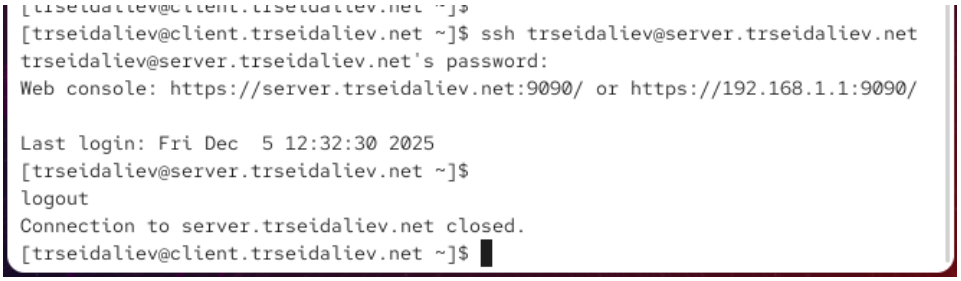

Рис. 2.4: Успешный вход пользователя trseidaliev

Это означает, что сервер разрешает вход всем локальным пользователям, кроме root, если иное не задано в конфигурации.

В конфигурацию sshd добавляется строка, разрешающая доступ только пользователю vagrant.
Фрагмент файла представлен ниже.

```
25
26 #HostKey /etc/ssh/ssh_host_rsa_key
27 #HostKey /etc/ssh/ssh_host_ecdsa_key
28 #HostKey /etc/ssh/ssh_host_ed25519_key
29
30 # Ciphers and keying
31 #RekeyLimit default none
32
33 # Logging
34 #SyslogFacility AUTH
35 #LogLevel INFO
36
37 # Authentication:
38
39 #LoginGraceTime 2m
40 PermitRootLogin no
41 #StrictModes yes
42 #MaxAuthTries 6
43 #MaxSessions 10
44
45 AllowUsers vagrant
46
47 #PubkeyAuthentication yes
48
```

Рис. 2.5: Правило AllowUsers vagrant

После перезапуска SSH-сервера правила аутентификации вступают в силу.

Теперь при попытке аутентификации пользователь trseidaliev получает отказ. Сервер блокирует доступ ещё до проверки пароля, так как учётная запись отсутствует в списке разрешённых.

```
[trseidaliev@client.trseidaliev.net ~]$ ssh trseidaliev@server.trseidaliev.net
trseidaliev@server.trseidaliev.net's password:
Permission denied, please try again.
trseidaliev@server.trseidaliev.net's password:
Permission denied, please try again.
trseidaliev@server.trseidaliev.net's password:
trseidaliev@server.trseidaliev.net: Permission denied (publickey,gssapi-keyex,gs
sapi-with-mic,password).
[trseidaliev@client.trseidaliev.net ~]$
```

Рис. 2.6: Отказ в доступе пользователю не из списка AllowUsers

Это подтверждает, что директива ограничивает круг пользователей, имеющих право на удалённый вход.

В список разрешённых добавляется второй пользователь — trseidaliev. Фрагмент обновлённой конфигурации показан на скриншоте.

```
20 #
21 #Port 22
22 #AddressFamily any
23 #ListenAddress 0.0.0.0
24 #ListenAddress ::
25
26 #HostKey /etc/ssh/ssh_host_rsa_key
27 #HostKey /etc/ssh/ssh_host_ecdsa_key
28 #HostKey /etc/ssh/ssh_host_ed25519_key
29
30 # Ciphers and keying
31 #RekeyLimit default none
32
33 # Logging
34 #SyslogFacility AUTH
35 #LogLevel INFO
36
37 # Authentication:
38
39 #LoginGraceTime 2m
40 PermitRootLogin no
41 #StrictModes yes
42 #MaxAuthTries 6
43 #MaxSessions 10
44
45 AllowUsers vagrant trseidaliev
46
47 #PubkeyAuthentication yes
```

Рис. 2.7: Разрешение двух пользователей в AllowUsers

После применения изменений вход пользователя trseidaliev снова выполняется успешно.

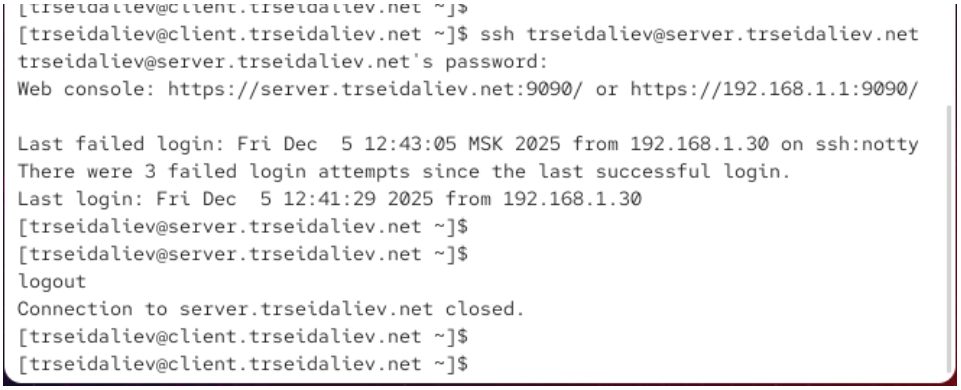

Рис. 2.8: Успешное подключение после добавления пользователя в AllowUsers

Это подтверждает корректную работу механизма ограничения и разрешения доступа через директиву AllowUsers.

2.3 Настройка дополнительных портов для удалённого доступа по SSH

В конфигурационный файл `/etc/ssh/sshd_config` добавлены строки, указывающие службе SSH прослушивать два порта — стандартный 22 и дополнительный 2022.

Это позволяет сохранить доступ к серверу даже при ошибках в конфигурации:

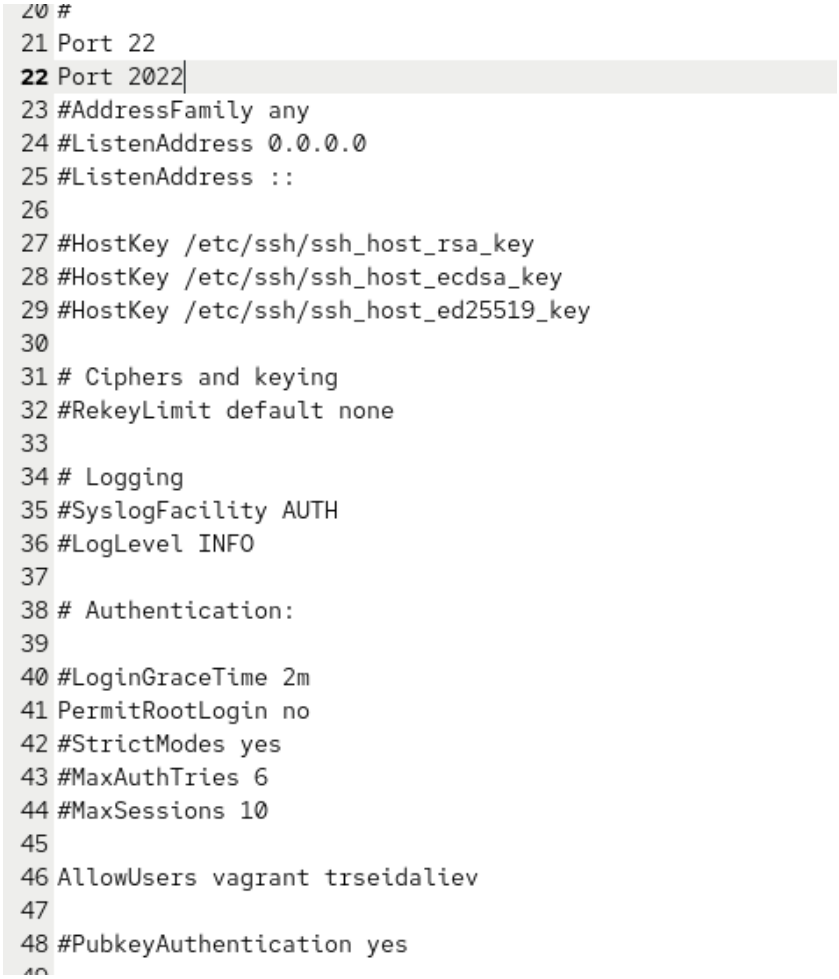

Рис. 2.9: Добавление порта 2022

После сохранения файла служба SSH была перезапущена.

При просмотре расширенного статуса работы sshd видно, что процесс не смог открыть порт 2022.

Система выводит сообщение об ошибке, указывающее на отказ доступа:

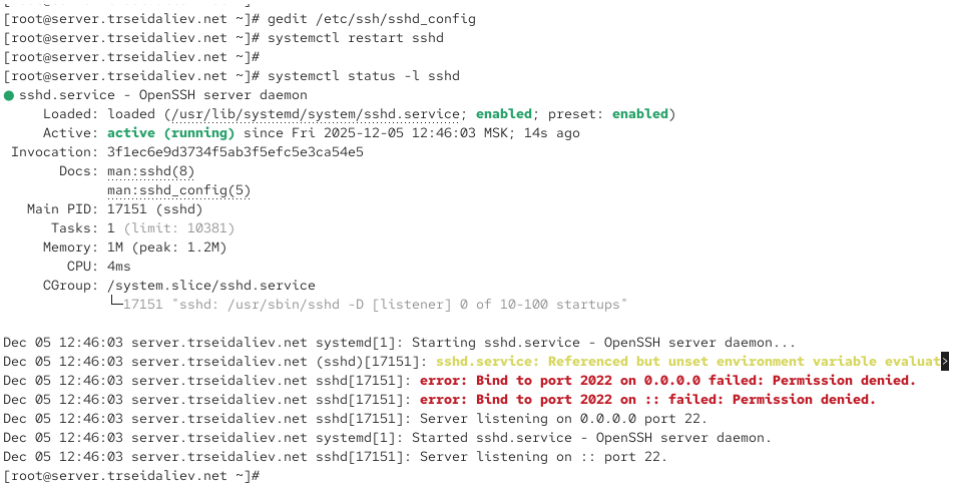

Рис. 2.10: Ошибка Permission denied при открытии порта 2022

Суть сообщения:

SELinux блокирует попытку сервиса sshd открыть нестандартный порт, поскольку для него не назначена корректная SELinux-метка.

Для разрешения sshd использовать порт 2022 была назначена корректная метка SELinux и открыт порт в межсетевом экране.

После выполнения необходимых команд sshd успешно стартовал и начал прослушивать оба порта:

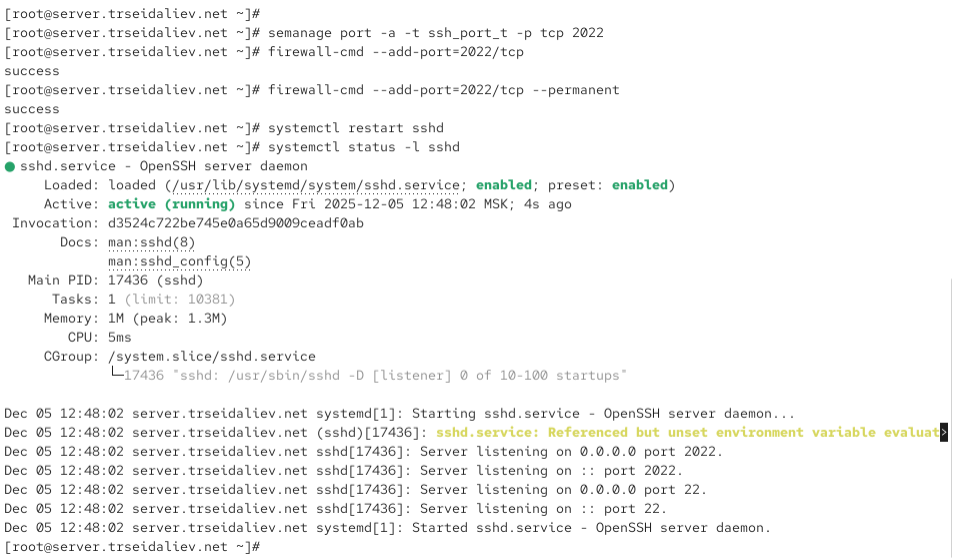

Рис. 2.11: Успешное прослушивание порта 22 и 2022

Теперь в статусе видно, что сервер прослушивает соединения одновременно на двух портах.

Выполнено подключение к серверу от имени пользователя trseidaliev через стандартный порт 22.

Аутентификация прошла успешно, доступ к системе получен.

После выполнения команды `sudo -i` получен доступ root.

Выход из сеанса выполнен двумя командами logout.

Аналогичное подключение выполнено, но с указанием дополнительного порта:

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

Last login: Fri Dec  5 12:44:14 2025 from 192.168.1.30
[trseidaliev@server.trseidaliev.net ~]$ sudo -i
[sudo] password for trseidaliev:
[root@server.trseidaliev.net ~]#
logout
[trseidaliev@server.trseidaliev.net ~]$
logout
Connection to server.trseidaliev.net closed.
[trseidaliev@client.trseidaliev.net ~]$
[trseidaliev@client.trseidaliev.net ~]$ ssh trseidaliev@server.trseidaliev.net -p2022
trseidaliev@server.trseidaliev.net's password:
Web console: https://server.trseidaliev.net:9090/ or https://192.168.1.1:9090/

Last login: Fri Dec  5 12:48:34 2025 from 192.168.1.30
[trseidaliev@server.trseidaliev.net ~]$ sudo -i
[sudo] password for trseidaliev:
[root@server.trseidaliev.net ~]#
logout
[trseidaliev@server.trseidaliev.net ~]$
logout
Connection to server.trseidaliev.net closed.
[trseidaliev@client.trseidaliev.net ~]$
```

Рис. 2.12: Успешное подключение через порт 2022

Вход через порт 2022 проходит корректно, что подтверждает правильную работу SELinux-меток и правил firewall.

Аутентификация завершается успешно, доступ root также получается через `sudo -i`.

2.4 Настройка удалённого доступа по SSH по ключу

В конфигурацию sshd добавлен параметр, разрешающий вход по ключам:

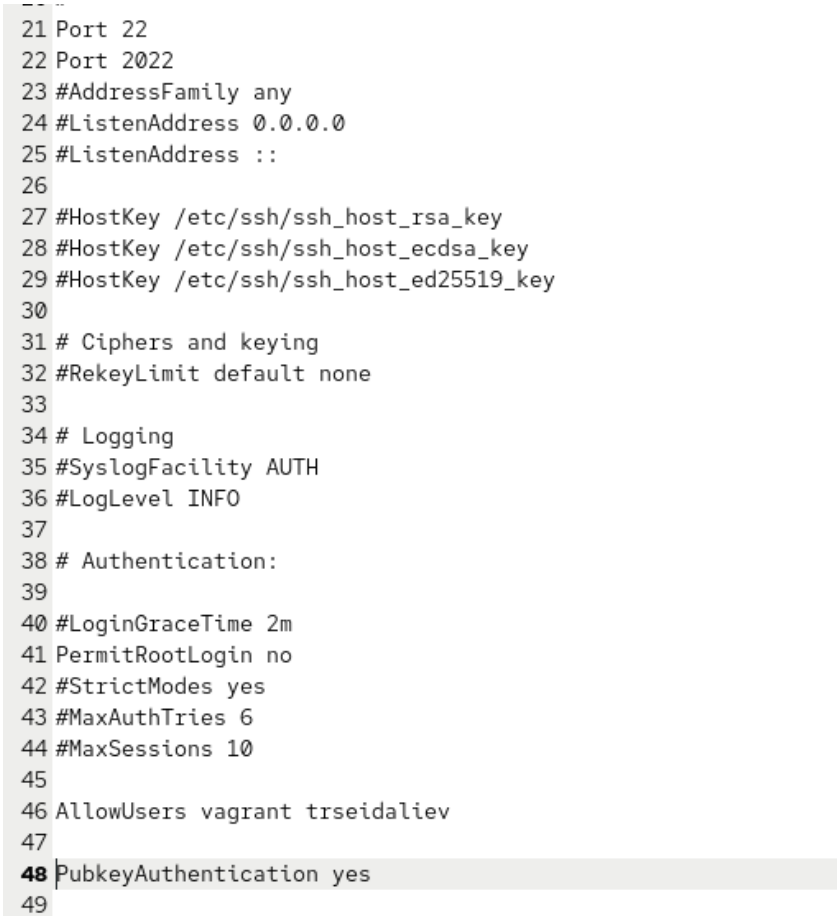

Рис. 2.13: Включение PubkeyAuthentication

После внесения изменений sshd был перезапущен.

На клиентской машине была создана пара SSH-ключей.
Закрытый ключ помещён в файл `~/.ssh/id_rsa`, открытый — в `~/.ssh/id_rsa.pub`.

Открытый ключ передан на сервер с помощью команды, которая автоматически добавляет его в файл `~/.ssh/authorized_keys` пользователя:

После успешной установки ключа система уведомляет о количестве добавленных ключей.

Далее выполнено подключение к серверу по SSH без ввода пароля:

```
|*+*.B .          |
|==+=.*           |
|.  =+=o o        |
|  ooB .o .       |
+----[SHA256]-----+
[trseidaliev@client.trseidaliev.net ~]$ ssh-copy-id trseidaliev@server.trseidaliev.net
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: ssh-add -L
/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 prompted now it is to i
nstall the new keys
trseidaliev@server.trseidaliev.net's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'trseidaliev@server.trseidaliev.net'"
and check to make sure that only the key(s) you wanted were added.

[trseidaliev@client.trseidaliev.net ~]$ ssh trseidaliev@server.trseidaliev.net
Web console: https://server.trseidaliev.net:9090/ or https://192.168.1.1:9090/

Last login: Fri Dec  5 12:48:57 2025 from 192.168.1.30
[trseidaliev@server.trseidaliev.net ~]$
logout
Connection to server.trseidaliev.net closed.
[trseidaliev@client.trseidaliev.net ~]$
```

Рис. 2.14: Успешное подключение по ключу

Аутентификация происходит автоматически — сервер использует ранее установленные ключи, что подтверждает корректную настройку.

2.5 Организация туннелей SSH, перенаправление TCP-портов

Перед созданием туннеля выполнялась проверка открытых TCP-соединений. В момент проверки список был пуст, что означает отсутствие активных прослушивающих или установленных TCP-соединений.

Это дало возможность обращаться к веб-серверу, работающему на удалённой машине, через локальный порт.

После запуска туннеля повторная проверка TCP-соединений показала появившиеся записи:

```
[trseidaliev@client.trseidaliev.net ~]$
[trseidaliev@client.trseidaliev.net ~]$ lsof | grep TCP
[trseidaliev@client.trseidaliev.net ~]$ ssh -fNL 8080:localhost:80 trseidaliev@server.trseidaliev
.net
[trseidaliev@client.trseidaliev.net ~]$ lsof | grep TCP
ssh       11726                   trseidaliev    3u     IPv4             73679       0t0
TCP client.trseidaliev.net:39702->mail.trseidaliev.net:ssh (ESTABLISHED)
ssh       11726                   trseidaliev    4u     IPv6             73691       0t0
TCP localhost:webcache (LISTEN)
ssh       11726                   trseidaliev    5u     IPv4             73692       0t0
TCP localhost:webcache (LISTEN)
[trseidaliev@client.trseidaliev.net ~]$
```

Рис. 2.15: Отображение TCP-соединений после создания туннеля

Список содержит:

- установленное SSH-соединение между клиентом и сервером,
- процесс, слушающий порт `localhost:webcache`, соответствующий локальному перенаправлению.

Это подтверждает корректную работу SSH-туннеля.

После открытия браузера и перехода по адресу `localhost:8080` отобразилась стартовая страница, обслуживаемая сервером:

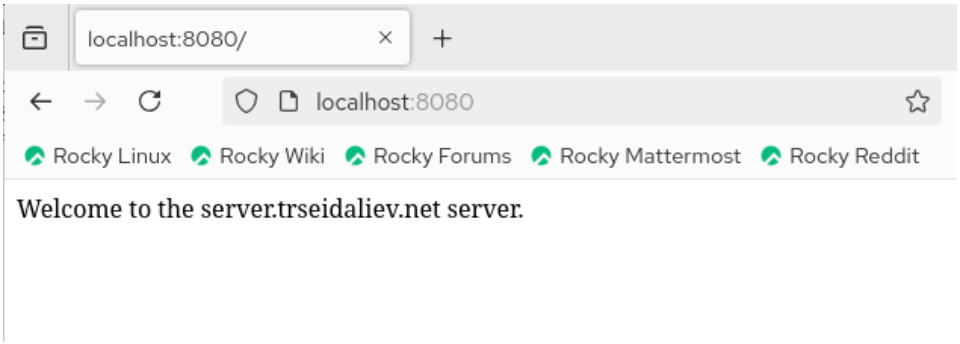

Рис. 2.16: Проверка доступа к веб-серверу через SSH-туннель

Это подтверждает, что все HTTP-запросы успешно перенаправляются через SSH-соединение.

2.6 Запуск консольных приложений через SSH

Удалённый вызов команды hostname корректно возвращает имя сервера:

```
[trseidaliev@client.trseidaliev.net ~]$ ssh trseidaliev@server.trseidaliev.net hostname
server.trseidaliev.net
[trseidaliev@client.trseidaliev.net ~]$ ssh trseidaliev@server.trseidaliev.net ls -Al
total 56
-rw-------.  1 trseidaliev trseidaliev  656 Dec  5 12:48 .bash_history
-rw-r--r--.  1 trseidaliev trseidaliev   18 Oct 29  2024 .bash_logout
-rw-r--r--.  1 trseidaliev trseidaliev  144 Oct 29  2024 .bash_profile
-rw-r--r--.  1 trseidaliev trseidaliev  549 Nov 19 10:19 .bashrc
drwx------. 11 trseidaliev trseidaliev 4096 Nov 19 18:22 .cache
drwx------. 12 trseidaliev trseidaliev 4096 Nov 30 11:07 .config
drwxr-xr-x.  2 trseidaliev trseidaliev    6 Nov 19 10:19 Desktop
drwxr-xr-x.  2 trseidaliev trseidaliev    6 Nov 19 10:19 Documents
drwxr-xr-x.  2 trseidaliev trseidaliev    6 Nov 19 10:19 Downloads
drwx------.  4 trseidaliev trseidaliev   32 Nov 19 10:19 .local
drwx------.  5 trseidaliev trseidaliev 4096 Nov 30 11:37 Maildir
drwxr-xr-x.  5 trseidaliev trseidaliev   54 Nov 19 18:22 .mozilla
drwxr-xr-x.  2 trseidaliev trseidaliev    6 Nov 19 10:19 Music
drwxr-xr-x.  2 trseidaliev trseidaliev    6 Nov 19 10:19 Pictures
```

Рис. 2.17: hostname через SSH

Команда удалённого просмотра каталога выводит содержимое домашнего каталога пользователя:

Удалённый запуск консольной почтовой программы показывает содержимое каталога Maildir:

```
[trseidaliev@client.trseidaliev.net ~]$
[trseidaliev@client.trseidaliev.net ~]$ ssh trseidaliev@server.trseidaliev.net MAIL=~/Maildir mai
l
s-nail version v14.9.24.  Type `?' for help
/home/trseidaliev/Maildir: 3 messages 1 unread
    1 trseidaliev           2025-11-30 10:46   18/685   "test1               "
    2 trseidaliev@client.t  2025-11-30 11:11   21/880   "LMTP TEST           "
▸U  3 trseidaliev           2025-11-30 11:37   22/861   "test3               "
q
Held 3 messages in /home/trseidaliev/Maildir
[trseidaliev@client.trseidaliev.net ~]$
```

Рис. 2.18: Просмотр почты через SSH

2.7 Разрешение X11-переадресации на сервере

В конфигурации sshd был разрешён вывод графических приложений на сторону клиента:

```
21 Port 22
22 Port 2022
23 #AddressFamily any
24 #ListenAddress 0.0.0.0
25 #ListenAddress ::
26
27 #HostKey /etc/ssh/ssh_host_rsa_key
28 #HostKey /etc/ssh/ssh_host_ecdsa_key
29 #HostKey /etc/ssh/ssh_host_ed25519_key
30
31 # Ciphers and keying
32 #RekeyLimit default none
33
34 # Logging
35 #SyslogFacility AUTH
36 #LogLevel INFO
37
38 # Authentication:
39
40 #LoginGraceTime 2m
41 PermitRootLogin no
42 #StrictModes yes
43 #MaxAuthTries 6
44 #MaxSessions 10
45
46 AllowUsers vagrant trseidaliev
47
48 PubkeyAuthentication yes
49
50 X11Forwarding yes
```

Рис. 2.19: Изменение настроек X11Forwarding

После изменения sshd был перезапущен.

При попытке запустить графическое приложение возникла ошибка:

```
[trseidaliev@client.trseidaliev.net ~]$
[trseidaliev@client.trseidaliev.net ~]$ ssh -YC trseidaliev@server.trseidaliev.net 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
[trseidaliev@client.trseidaliev.net ~]$
[trseidaliev@client.trseidaliev.net ~]$ ssh -YC trseidaliev@server.trseidaliev.net 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
[trseidaliev@client.trseidaliev.net ~]$
```

Рис. 2.20: Ошибка X11 перенаправления

Смысл ошибки:

- сервер разрешает X11-forwarding,
- но на клиенте отсутствует переменная DISPLAY,
- значит X-сервер на клиентской машине не запущен или не настроен.

Поэтому передавать графический интерфейс невозможно.

2.8 Внесение изменений в настройки внутреннего окружения виртуальной машины

В каталоге `/vagrant/provision/server/` был создан каталог `ssh/etc/ssh`, куда помещён действующий конфигурационный файл SSH-демона.

После создания файла `ssh.sh` ему были назначены права на исполнение. Внутри файла содержится автоматизация всех изменений, сделанных вручную:

```
1  #!/bin/bash
2  echo "Provisioning script $0"
3  echo "Copy configuration files"
4  cp -R /vagrant/provision/server/ssh/etc/* /etc
5  restorecon -vR /etc
6  echo "Configure firewall"
7  firewall-cmd --add-port=2022/tcp
8  firewall-cmd --add-port=2022/tcp --permanent
9  echo "Tuning SELinux"
10 semanage port -a -t ssh_port_t -p tcp 2022
11 echo "Restart sshd service"
12 systemctl restart sshd
13
```

Рис. 2.21: Содержимое файла ssh.sh

3 Заключение

В ходе выполнения работы:

- настроены ограничения доступа по SSH, включая запрет входа для root и разрешение соединений только определённым пользователям;
- реализована работа SSH-сервера через два порта, устранены ограничения SELinux и настроен межсетевой экран для дополнительного порта 2022;
- проверена возможность подключения по SSH как по стандартному, так и по дополнительному порту;
- выполнена настройка аутентификации по открытым ключам и подтверждена успешная работа входа без пароля;
- опробовано создание SSH-туннелей и локальной переадресации портов, обеспечив доступ к веб-сервису через защищённое соединение;
- выполнен удалённый запуск консольных приложений и просмотр почты через SSH, изучены ограничения при использовании X11-перенаправления;
- подготовлены изменения во внутреннем окружении виртуальной машины, включая создание провижининг-скрипта для автоматической настройки SSH.

Результаты лабораторной работы подтвердили корректность настройки SSH и связанных механизмов безопасности, а также продемонстрировали практическое применение туннелирования и аутентификации по ключам.

4 Контрольные вопросы

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

Чтобы запретить вход по SSH для root и разрешить его пользователю alice, необходимо изменить файл конфигурации sshd на сервере.

В конфигурацию `/etc/ssh/sshd_config` вносятся строки:

- `PermitRootLogin no` — запрещает вход root-пользователя;
- `AllowUsers alice` — разрешает вход только пользователю alice.

После изменения файл сохраняется, и служба SSH перезапускается.
В результате удалённый вход root становится недоступным, а пользователь alice получает полный доступ по SSH.

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

Чтобы настроить SSH на работу через несколько портов, в конфигурации sshd последовательно добавляются строки вида:

```
Port 22
Port 2022
```

После перезапуска SSH-сервера он начинает прослушивать указанные порты одновременно.

Такое решение применяется:

- для обеспечения резервного доступа, если основной порт становится недоступен;
- для тестирования изменений конфигурации перед их окончательным применением;
- для обхода ограничений межсетевых экранов, блокирующих нестандартные порты;
- для повышения безопасности путём изменения стандартного порта 22.

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

Для создания фонового SSH-туннеля применяются параметры:

- `-f` — переводит SSH в фоновый режим после аутентификации;
- `-N` — отключает выполнение удалённой команды (SSH создаёт только туннель);
- `-L` — задаёт локальную переадресацию портов.

Фоновое соединение не открывает удалённую оболочку и служит только каналом для передачи данных через SSH-туннель.

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

Для создания локального туннеля используется конструкция перенаправления портов:

```
-L 5555:localhost:80
```

Она связывает локальный порт 5555 с портом 80 удалённого сервера. После установления SSH-соединения браузер, открывающий `localhost:5555`, фактически обращается к веб-серверу server2.example.com.

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

Чтобы разрешить sshd открывать нестандартный порт 2022, необходимо назначить ему корректную SELinux-метку:

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

После выполнения команды SELinux разрешает службе SSH слушать этот порт. Только после этого sshd сможет успешно запуститься, если в конфигурации указан порт 2022.

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

Для открытия порта 2022 в firewall выполняются две команды:

- временное открытие:

  `firewall-cmd --add-port=2022/tcp`
- постоянное открытие (после перезагрузки сохраняется):

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

После изменения конфигурации firewall перезагружается, и порт становится доступен для входящих SSH-подключений.