Отчёт по лабораторной работе 11

Настройка безопасного удалённого доступа по протоколу SSH

Элсаиед Адел

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

1. На сервере для пользователя **root** был задан пароль (при необходимости) с переходом в привилегированный режим и выполнением процедуры установки пароля. Это обеспечивает возможность парольной аутентификации root в случае, если она разрешена настройками SSH-сервера.

2. В дополнительном терминале на сервере был запущен мониторинг системных событий и сообщений службы SSH для отслеживания попыток удалённого входа, а также причин успешной или неуспешной аутентификации.

3. С клиентской машины выполнена попытка подключения к серверу по SSH под пользователем **root**. В процессе аутентификации в системном журнале сервера зафиксированы сообщения о неуспешной проверке пароля и отказе во входе. Это указывает на то, что SSH-сервер принял соединение, запросил учётные данные, однако аутентификация не была успешно завершена.

```
Jan 06 08:16:04 server.elsaiedadel.net agetty[12157]: -: failed to get terminal attributes: Input/output error
Jan 06 08:16:06 server.elsaiedadel.net sshd-session[12144]: Failed password for root from 192.168.1.30 port 43482 ssh2
Jan 06 08:16:09 server.elsaiedadel.net unix_chkpwd[12159]: password check failed for user (root)
Jan 06 08:16:09 server.elsaiedadel.net kernel: traps: VBoxClient[12163] trap int3 ip:41dd1b sp:7f59f9757cd0 error:0 in
 VBoxClient[1dd1b,400000+bb000]
Jan 06 08:16:09 server.elsaiedadel.net systemd-coredump[12164]: Process 12160 (VBoxClient) of user 1001 terminated abn
ormally with signal 5/TRAP, processing...
```

Рис. 2.1: Попытка входа по SSH под root до запрета (сообщения journalctl)

4. Для запрета удалённого входа пользователю **root** на сервере был открыт

файл конфигурации SSH-сервера `/etc/ssh/sshd_config`, в котором установлен параметр `PermitRootLogin no`. Данная настройка запрещает удалённый вход под пользователем root независимо от используемого способа аутентификации.

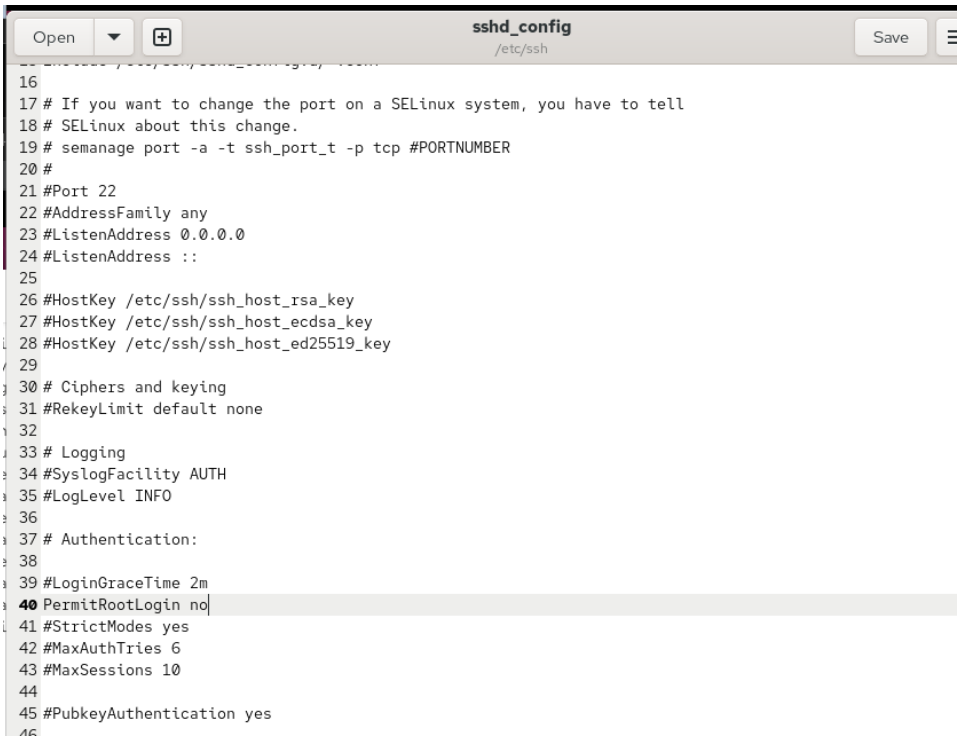

Рис. 2.2: Запрет входа root через PermitRootLogin no

5. После сохранения изменений служба SSH была перезапущена для применения новой конфигурации.

6. После запрета входа root повторно выполнена попытка подключения с клиента под пользователем **root**. В журнале сервера зафиксировано завершение соединения на этапе предварительной аутентификации и сообщения о неуспешных попытках входа. Это соответствует штатному поведению SSH-сервера при запрете удалённого входа пользователя root.

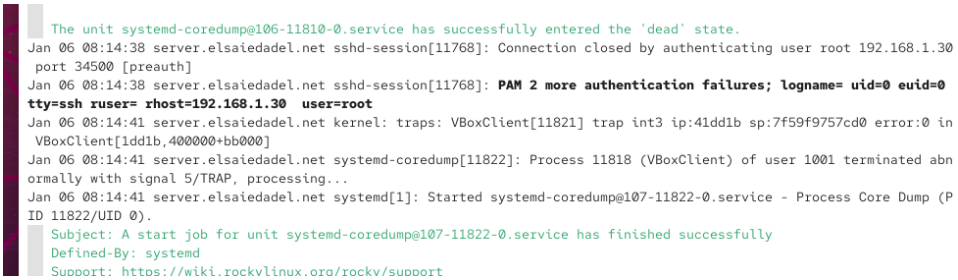

Рис. 2.3: Попытка входа root после запрета PermitRootLogin

2.2 Ограничение списка пользователей для удалённого доступа по SSH

7. С клиентской машины выполнена проверка подключения по SSH под пользователем **elsaiedadel**. Подключение прошло успешно: сервер запросил пароль, после его ввода была открыта интерактивная пользовательская сессия.

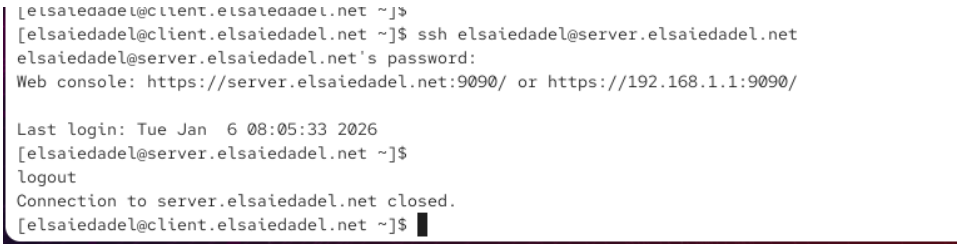

Рис. 2.4: Успешное SSH-подключение пользователя elsaiedadel до ограничения

8. На сервере в файле конфигурации `/etc/ssh/sshd_config` добавлено ограничение списка пользователей с помощью параметра `AllowUsers vagrant`. После сохранения конфигурации служба SSH была перезапущена. Данная настройка разрешает удалённый доступ только пользователю `vagrant`, а все остальные подключения отклоняются.

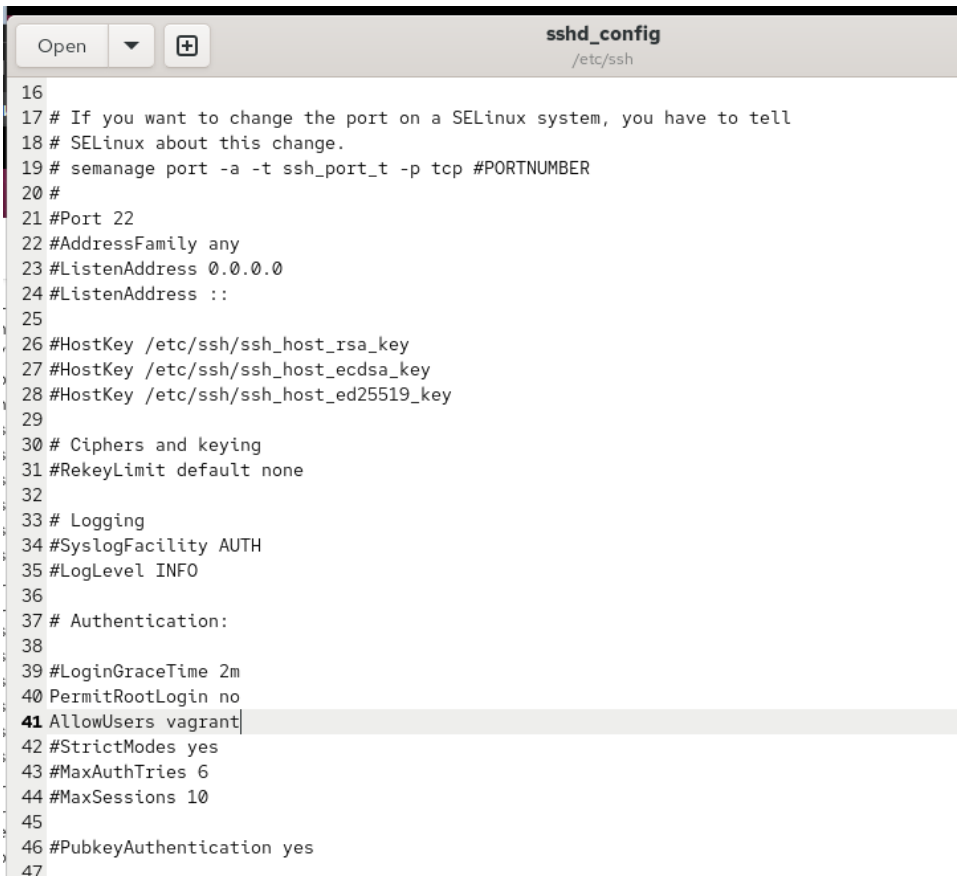

Рис. 2.5: Ограничение доступа по SSH параметром AllowUsers vagrant

9. После введения ограничения выполнена повторная попытка подключения по SSH под пользователем **elsaiedadel**. Соединение завершилось отказом с сообщением об отсутствии прав доступа, так как данный пользователь не был указан в списке разрешённых.

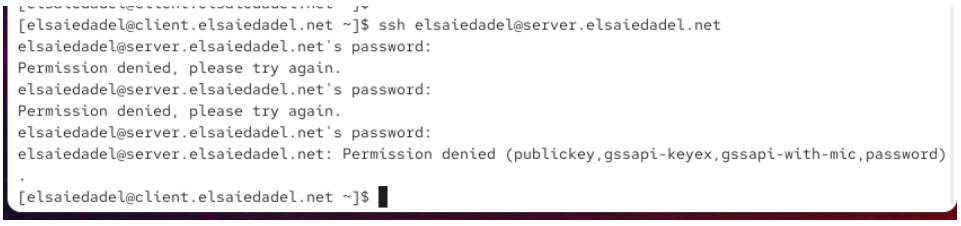

Рис. 2.6: Отказ в доступе пользователю elsaiedadel при AllowUsers vagrant

10. Для восстановления доступа пользователя **elsaiedadel** в конфигурации SSH-сервера выполнено изменение параметра `AllowUsers` с добавлением дан-

ного пользователя в список разрешённых. После сохранения файла служба SSH была перезапущена.

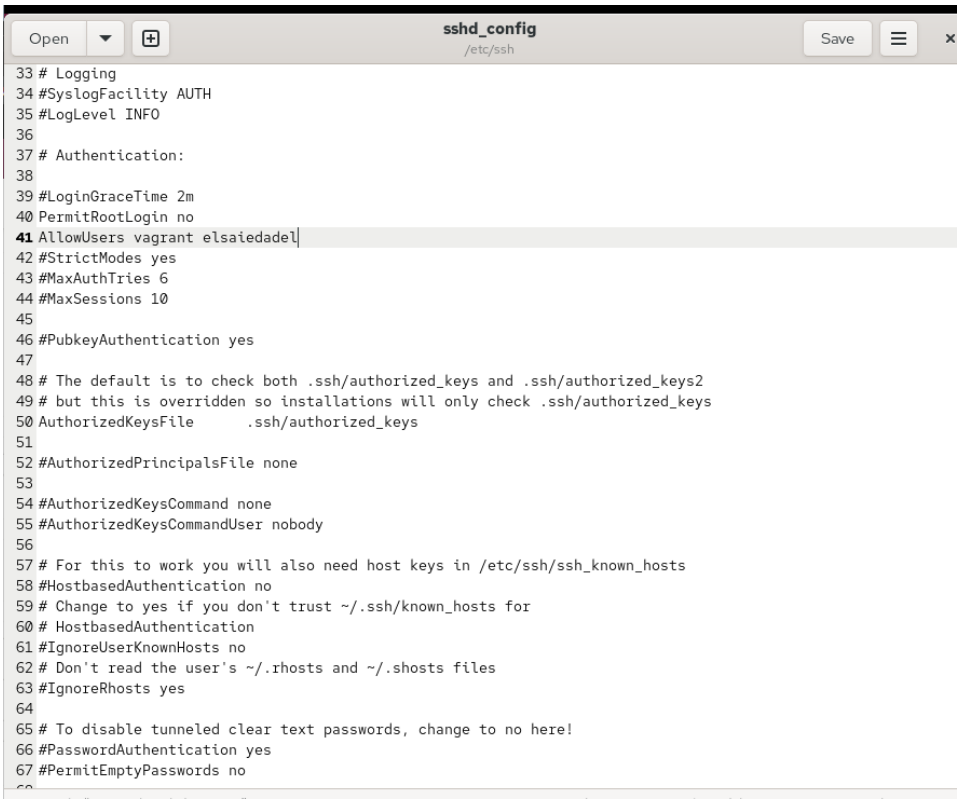

Рис. 2.7: Добавление пользователя elsaiedadel в AllowUsers

11. После обновления списка разрешённых пользователей повторно выполнена попытка подключения по SSH под пользователем **elsaiedadel**. Подключение прошло успешно, что подтверждает корректность применённых ограничений и восстановление доступа для указанного пользователя. Дополнительно отображено уведомление о количестве неудачных попыток входа с момента предыдущего успешного подключения.

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

Last failed login: Tue Jan  6 08:18:48 UTC 2026 from 192.168.1.30 on ssh:notty
There were 3 failed login attempts since the last successful login.
Last login: Tue Jan  6 08:17:08 2026 from 192.168.1.30
[elsaiedadel@server.elsaiedadel.net ~]$
logout
Connection to server.elsaiedadel.net closed.
[elsaiedadel@client.elsaiedadel.net ~]$
```

Рис. 2.8: Успешное SSH-подключение после добавления в AllowUsers

2.3 Изменение порта службы SSH на 2022

12. На сервере в файле `/etc/ssh/sshd_config` выполнена попытка изменения порта службы SSH. В конфигурации была задана строка с некорректным значением порта (параметр `Port` был указан в неверном формате), что привело к ошибке применения конфигурации при перезапуске службы.

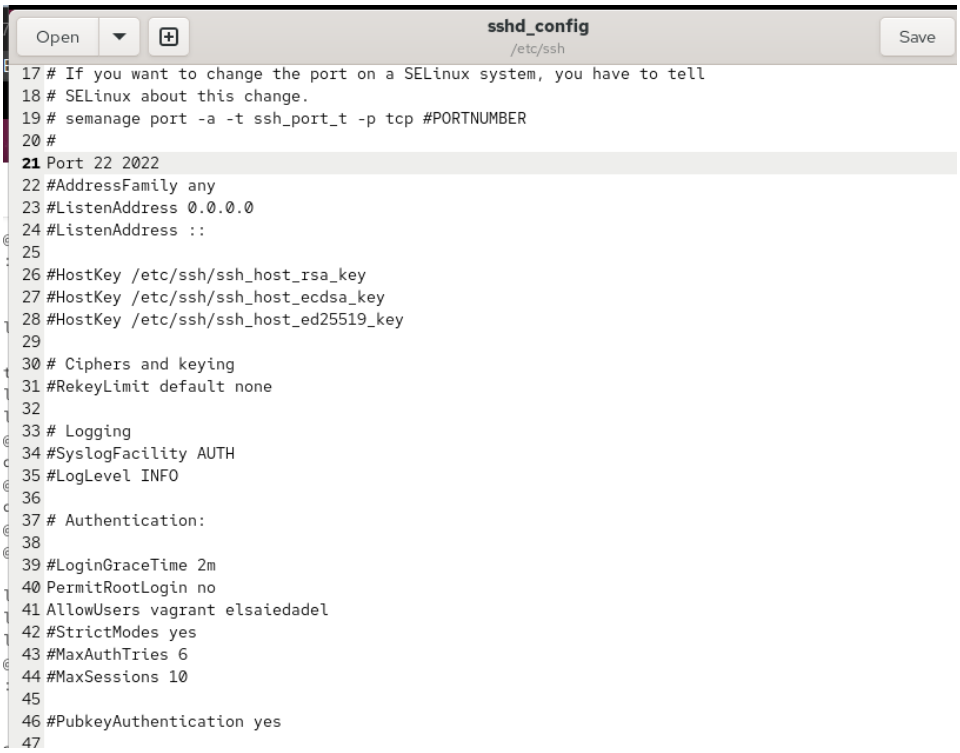

Рис. 2.9: Некорректная настройка параметра Port в sshd_config

13. После сохранения конфигурации выполнен перезапуск `sshd`, однако служба не запустилась. В выводе `systemctl` зафиксировано завершение процесса sshd с ошибкой (`status=255/EXCEPTION`), что соответствует ошибке чтения/применения конфигурации.

```
[root@server.elsaiedadel.net ~]# systemctl restart sshd
Job for sshd.service failed because the control process exited with error code.
See "systemctl status sshd.service" and "journalctl -xeu sshd.service" for details.
[root@server.elsaiedadel.net ~]# systemctl status sshd -l
● sshd.service - OpenSSH server daemon
     Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; preset: enabled)
     Active: activating (auto-restart) (Result: exit-code) since Tue 2026-01-06 08:21:55 UTC; 11s ago
 Invocation: 31cb3bff9953439aabe58b4f598567e2
       Docs: man:sshd(8)
             man:sshd_config(5)
    Process: 13126 ExecStart=/usr/sbin/sshd -D $OPTIONS (code=exited, status=255/EXCEPTION)
   Main PID: 13126 (code=exited, status=255/EXCEPTION)
   Mem peak: 1.5M
        CPU: 2ms
```

Рис. 2.10: Сбой перезапуска sshd из-за ошибки конфигурации

14. Для корректного переноса SSH на порт **2022** выполнены дополнительные настройки безопасности:

- для SELinux добавлено разрешение использования порта 2022 службой SSH через назначение контекста `ssh_port_t`;
- в `firewalld` открыт порт `2022/tcp` (временно и постоянно);
- файл `/etc/ssh/sshd_config` исправлен (параметр `Port` задан корректно отдельной строкой), после чего `sshd` успешно перезапущен. В статусе службы подтверждено состояние `active (running)`, а в журнале видно, что демон слушает порт **2022** (и также остаётся активным стандартный порт **22**, так как он указан отдельной строкой).

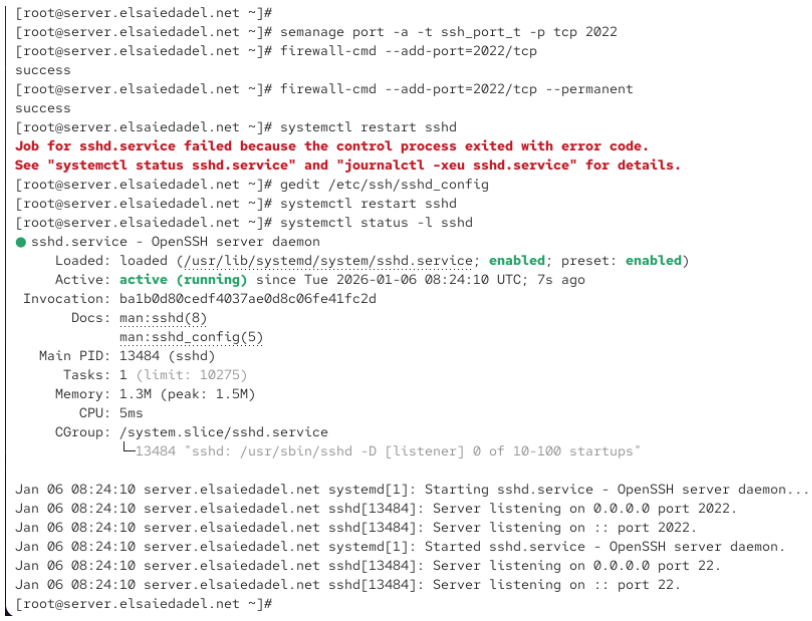

Рис. 2.11: Настройка SELinux и firewalld для порта 2022 и успешный запуск sshd

15. С клиентской машины выполнена проверка подключения:

- подключение по умолчанию (без указания порта) выполняется на стандартный порт **22** и проходит успешно;
- подключение с явным указанием порта `-p2022` также проходит успешно. Это подтверждает, что сервер принимает SSH-соединения одновременно на двух портах (22 и 2022), что соответствует конфигурации с двумя директивами `Port`.

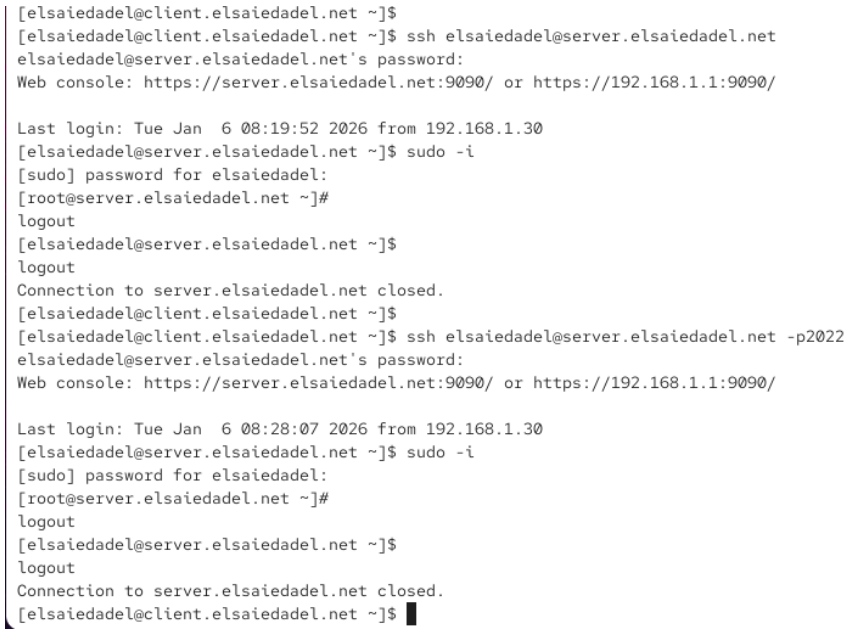

Рис. 2.12: Проверка SSH-подключения на портах 22 и 2022

2.4 Настройка удалённого доступа по SSH по ключу

16. На сервере в конфигурации `/etc/ssh/sshd_config` включена аутентификация по ключу установкой параметра `PubkeyAuthentication yes`. Также подтверждено, что ранее применённые ограничения безопасности остаются активными (запрет root-входа и ограничение списка пользователей через `AllowUsers`). После сохранения файла выполнен перезапуск `sshd` для применения настроек.

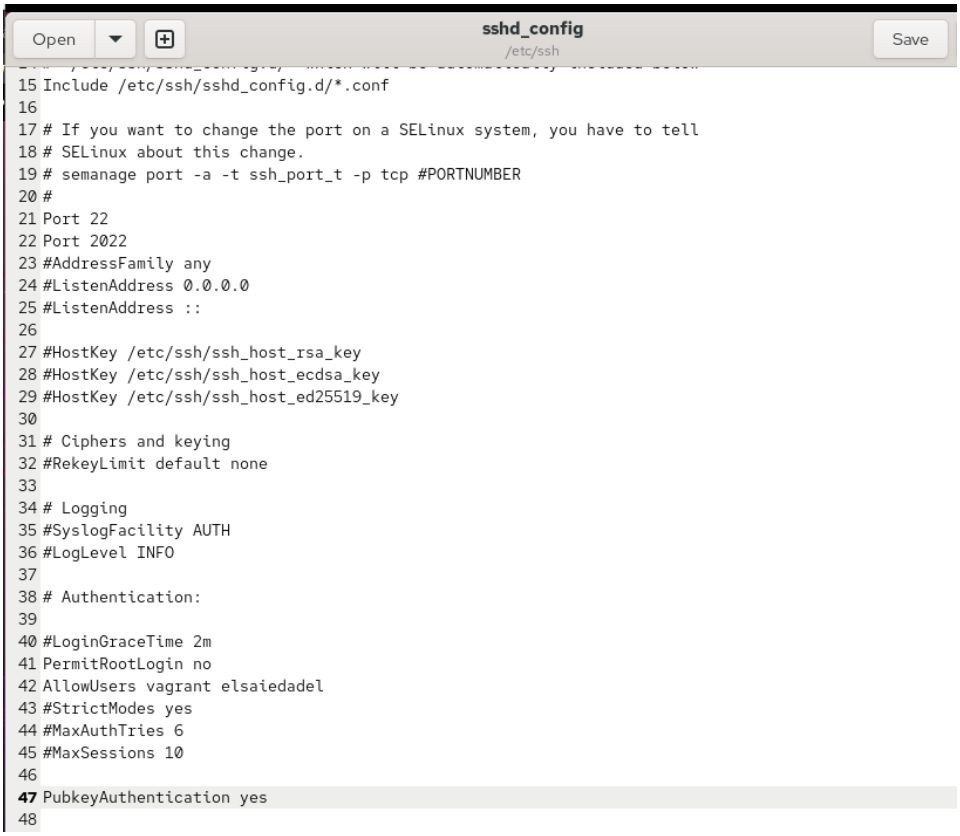

Рис. 2.13: Разрешение PubkeyAuthentication yes в sshd_config

17. На клиентской машине сформирована пара ключей SSH, после чего открытый ключ скопирован на сервер для пользователя **elsaiedadel** с помощью `ssh-copy-id`. Далее выполнено повторное подключение к серверу: аутентификация прошла без запроса пароля, что подтверждает корректную установку ключа в `~/.ssh/authorized_keys` и работу ключевой аутентификации.

Рис. 2.14: Копирование ключа на сервер и вход без пароля

2.5 Организация SSH-туннелей и перенаправление TCP-портов

18. На клиентской машине выполнена проверка активных TCP-соединений и слушающих сокетов. Затем организовано перенаправление порта: удалённый порт **80** на сервере перенаправлен на локальный порт **8080** с использованием SSH-туннеля в фоновом режиме. После настройки туннеля повторная проверка TCP-сокетов показала:

- наличие SSH-соединения в состоянии `ESTABLISHED`;
- появление локального слушающего порта `localhost:webcache (LISTEN)`, что соответствует локальному порту **8080** (служба/алиас может отображаться как `webcache`). Таким образом подтверждено, что локальный порт 8080 открыт процессом SSH и обслуживает туннель до удалённого веб-сервиса.

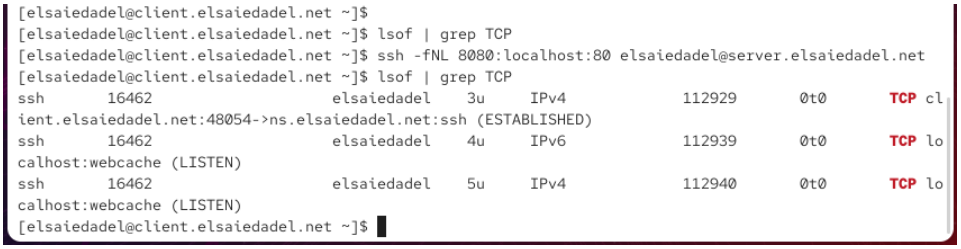

Рис. 2.15: Состояние TCP-соединений до и после создания SSH-туннеля 8080->80

19. В браузере на клиентской машине открыт адрес `localhost:8080`. Отображена страница приветствия веб-сервера, обслуживаемого на стороне сервера, что подтверждает работоспособность туннеля и корректное перенаправление TCP-порта.

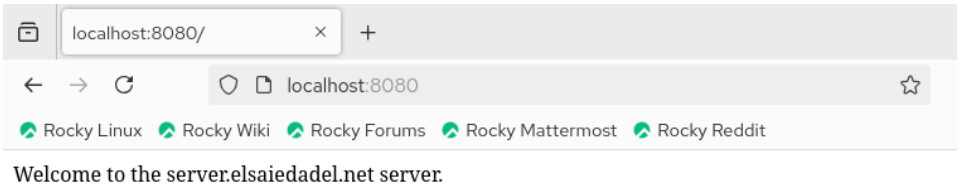

Рис. 2.16: Проверка туннеля в браузере: страница Welcome через localhost:8080

2.6 Запуск консольных приложений через SSH

20. С клиентской машины выполнен удалённый запуск консольных команд на сервере без открытия интерактивной сессии:

- получено имя узла сервера через выполнение `hostname`;
- выведен список файлов домашнего каталога пользователя командой `ls -Al`;

- выполнен запуск почтового клиента с указанием переменной окружения `MAIL=~/Maildir/`, в результате отображён список сообщений. Это демонстрирует возможность выполнять на сервере отдельные команды по SSH и получать их вывод в терминал клиента.

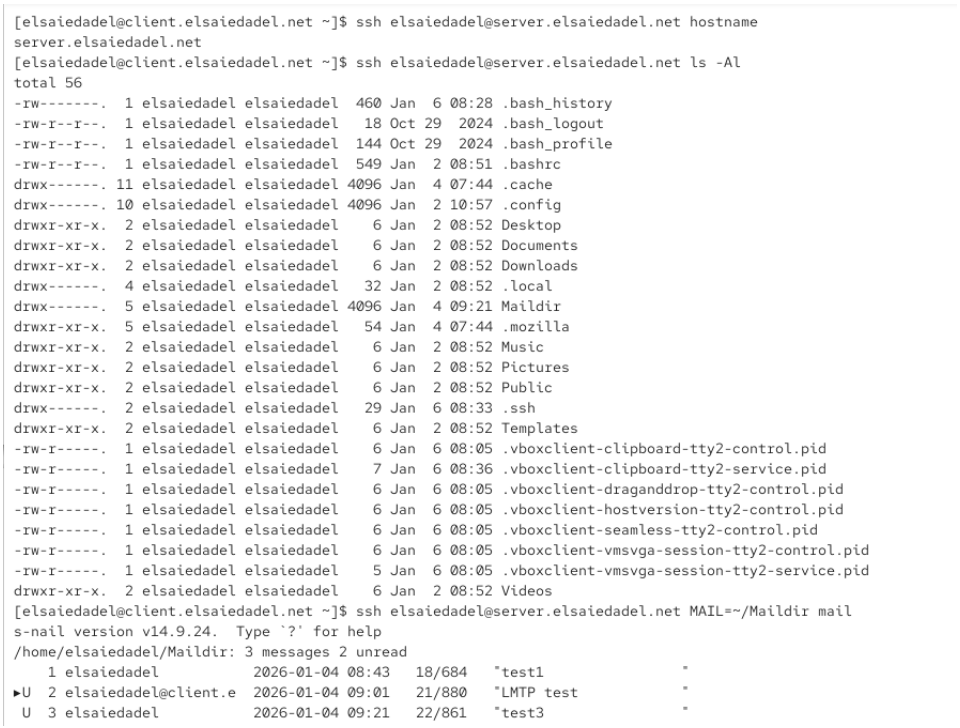

Рис. 2.17: Удалённый запуск команд hostname, ls и mail через SSH

2.7 Запуск графических приложений через SSH (X11 Forwarding)

21. На сервере в конфигурационном файле `/etc/ssh/sshd_config` была разрешена передача графического интерфейса X11 путём установки параметра `X11Forwarding yes`. Дополнительно подтверждено, что остальные параметры безопасности SSH (запрет входа root, ограничение пользователей, аутентификация по ключу, настройка портов) сохранены без изменений.

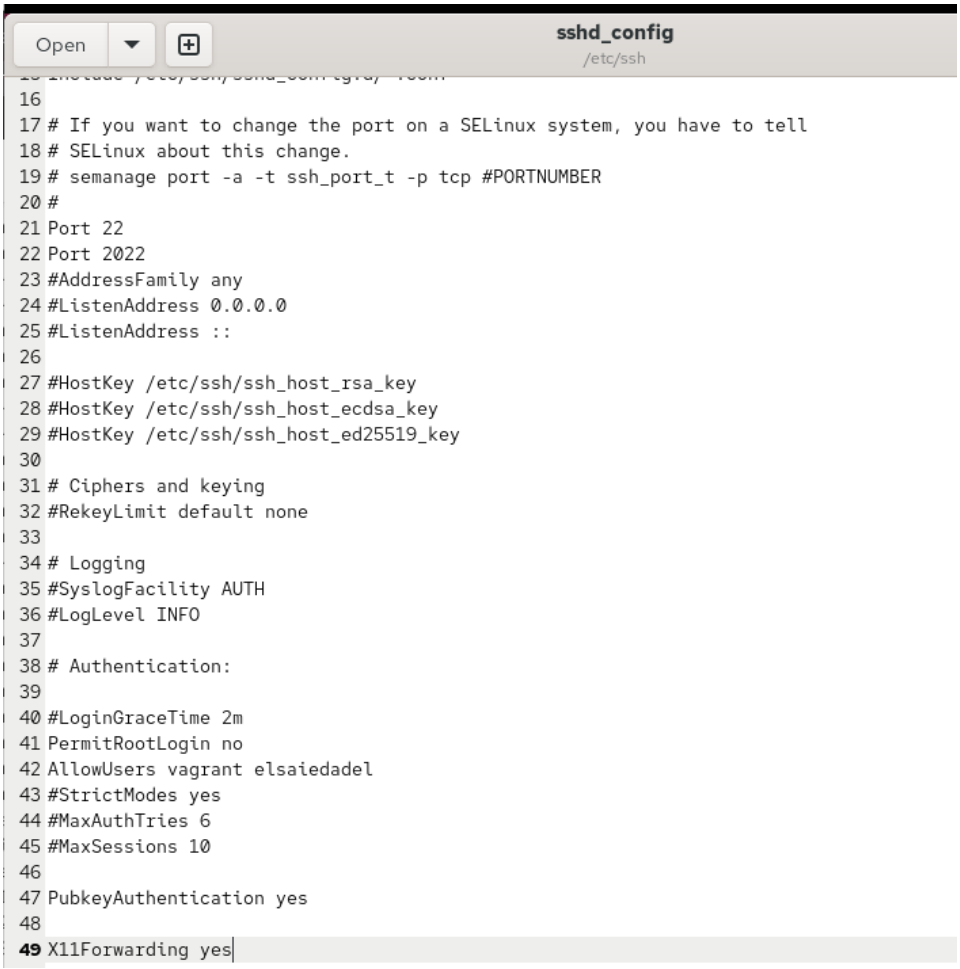

Рис. 2.18: Разрешение X11Forwarding в sshd_config

22. После сохранения изменений служба SSH была перезапущена для применения обновлённой конфигурации и активации поддержки X11 Forwarding.

23. С клиентской машины выполнена попытка удалённого запуска графического приложения Firefox на сервере с использованием SSH-подключения с поддержкой X11. При попытке запуска в терминале клиента были выведены сообщения об ошибке, указывающие на отсутствие переменной окружения DISPLAY и невозможность установить X11-соединение. Это свидетельствует о том, что на клиентской стороне отсутствует или не запущен X-сервер, либо не настроено окружение для приёма X11-перенаправления.

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

Рис. 2.19: Ошибка запуска графического приложения по SSH из-за отсутствия DISPLAY

2.8 Внесение изменений в настройки внутреннего окружения виртуальной машины

24. На виртуальной машине **server** выполнен переход в каталог `/vagrant/provision/server`, предназначенный для хранения файлов автоматической инициализации. Внутри него создана иерархия каталогов `ssh/etc/ssh`, в которую скопирован актуальный конфигурационный файл `sshd_config`. Это обеспечивает хранение эталонной версии конфигурации SSH в проекте Vagrant.

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

Рис. 2.20: Копирование sshd_config в каталог provision/server/ssh

25. В каталоге `/vagrant/provision/server` создан исполняемый файл `ssh.sh`, предназначенный для автоматической настройки SSH-сервера при запуске виртуальной машины. Файлу были назначены права на выполнение, после чего он открыт на редактирование.

26. В файл `ssh.sh` внесён provisioning-скрипт, выполняющий следующие действия при загрузке виртуальной машины:

- копирование конфигурационных файлов SSH из каталога provision в системный каталог `/etc`;

- восстановление контекстов безопасности SELinux;
- настройку firewall для разрешения TCP-порта 2022;
- добавление порта 2022 в контексты SELinux для службы SSH;
- перезапуск службы sshd для применения всех изменений.

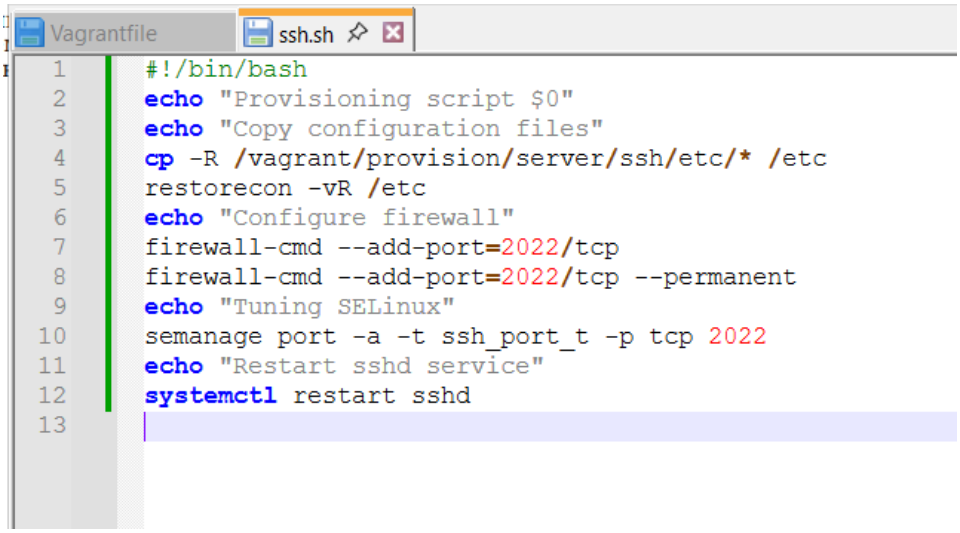

Рис. 2.21: Provisioning-скрипт ssh.sh для автоматической настройки SSH

27. Для автоматического выполнения созданного provisioning-скрипта во время загрузки виртуальной машины **server** в конфигурационный файл `Vagrantfile` был добавлен shell-provisioner с указанием пути к файлу `provision/server/ssh.sh` и сохранением порядка выполнения. Это гарантирует, что все настройки SSH, firewall и SELinux применяются автоматически при инициализации окружения без ручного вмешательства.

3 Вывод

В ходе выполнения лабораторной работы была выполнена комплексная настройка службы SSH на сервере Rocky Linux. Реализованы меры по повышению безопасности удалённого доступа: запрещён вход по SSH для пользователя root, ограничен список пользователей, разрешённых для подключения, настроена аутентификация по ключам, изменён стандартный порт SSH и обеспечена его корректная работа с учётом требований SELinux и межсетевого экрана. Дополнительно были рассмотрены возможности SSH по организации туннелей, перенаправлению TCP-портов, запуску консольных и графических приложений, а также автоматизации конфигурации SSH с использованием provisioning-скриптов в среде Vagrant. Проведённые проверки подтвердили корректную работу всех настроек и практическую применимость SSH как универсального инструмента администрирования удалённых систем.

4 Контрольные вопросы

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

Для запрета удалённого доступа пользователю root в конфигурационном файле `/etc/ssh/sshd_config` необходимо установить параметр `PermitRootLogin no`. Для разрешения доступа пользователю alice следует либо явно указать его в параметре `AllowUsers alice`, либо убедиться, что данный параметр отсутствует или не ограничивает список пользователей. После внесения изменений требуется перезапустить службу sshd для применения новой конфигурации.

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

Для настройки доступа по SSH через несколько портов в файле `sshd_config` необходимо указать несколько директив `Port`, каждая из которых задаёт отдельный порт, например стандартный 22 и дополнительный 2022. Такая настройка может потребоваться для повышения безопасности, обхода сетевых ограничений, организации плавного перехода на новый порт без прерывания доступа или для разграничения типов подключений.

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

Для создания такого туннеля используются параметры `-f`, `-N` и `-L`. Ключ `-f` переводит SSH-соединение в фоновый режим, `-N` указывает не выполнять удалённую команду, а `-L` задаёт правило локальной переадресации портов. Совместное

использование этих параметров позволяет организовать туннель без интерактивной сессии.

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

Локальная переадресация настраивается указанием правила перенаправления, в котором локальный порт 5555 связывается с портом 80 сервера server2.example.com. В результате все обращения к локальному порту 5555 на клиентской машине будут передаваться по SSH-туннелю на веб-сервер, работающий на удалённом узле.

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

Для этого необходимо добавить порт 2022 в контекст безопасности `ssh_port_t`, используемый службой SSH. Таким образом SELinux будет считать данный порт допустимым для работы sshd и не будет блокировать соединения на нём.

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

В конфигурации межсетевого экрана требуется разрешить входящие TCP-соединения на порт 2022. Для этого порт добавляется в список разрешённых как временно, так и на постоянной основе, после чего межсетевой экран начинает пропускать SSH-подключения через указанный порт.