РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра теории вероятностей и кибербезопасности

Отчет лабораторной работы 11

Дисциплина: Администрирование сетевых подсистем

Студент: Астахова Марина

Группа: НПИбд-02-23

Тема: Настройка безопасного удалённого доступа по протоколу SSH

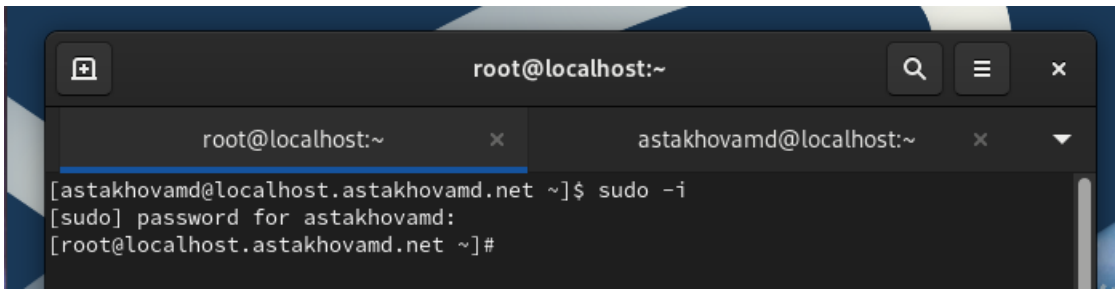
11.1. Цель работы.

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

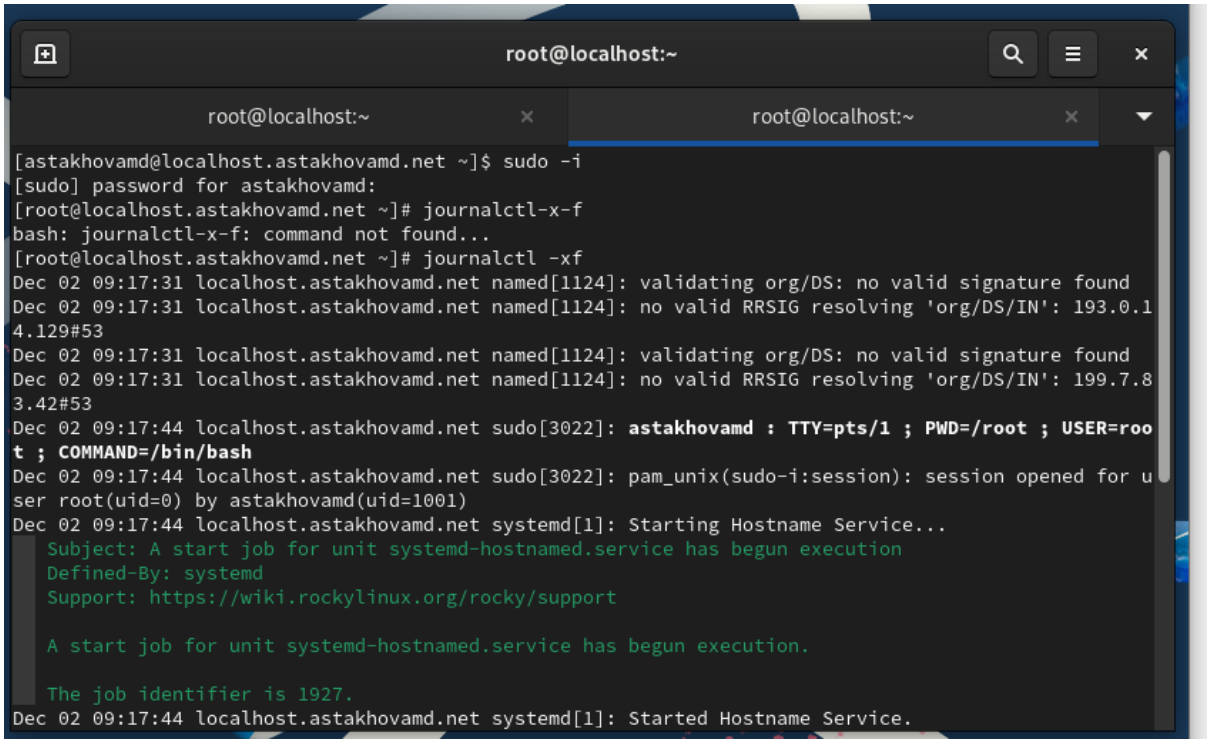
11.2. Выполнение работы

11.2.1. Запрет удалённого доступа по SSH для пользователя root

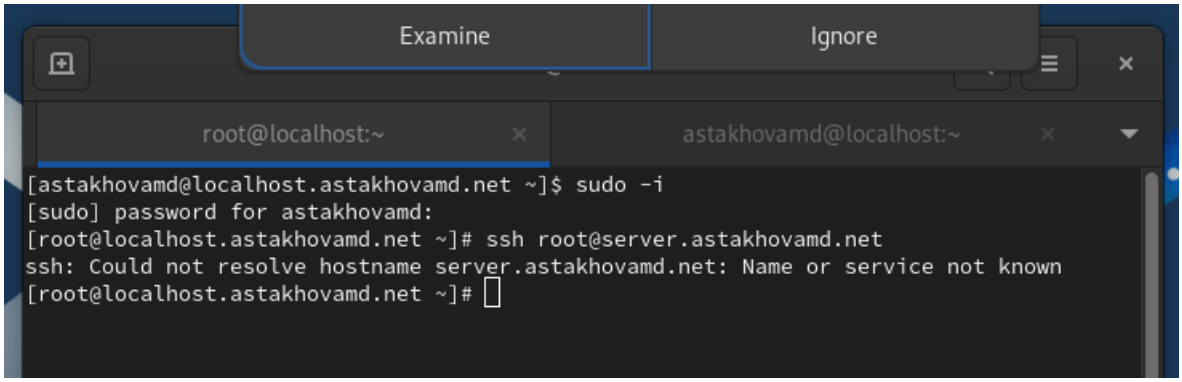
Перехожу в режим суперпользователя и ввожу пароль



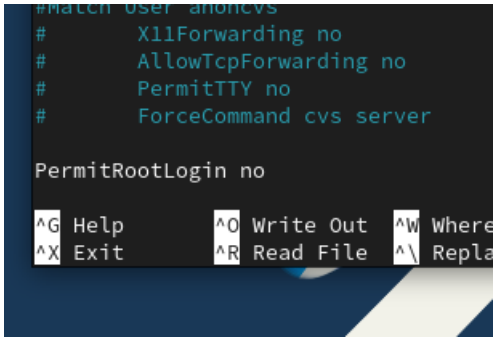
На сервере в дополнительном терминале запускаю мониторинг системных событий:



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



Так как сервис не найден, на сервере откройте файл /etc/ssh/sshd_config конфигурации sshd для редактирования и запретите вход на сервер пользователю root, установив:



Перезапускаю sshd

```
[root@localhost.astakhovamd.net ssh]# sudo nano sshd_config
[root@localhost.astakhovamd.net ssh]# systemctl restart sshd
[root@localhost.astakhovamd.net ssh]#
```

Повторяю попытку:

```
[root@localhost.astakhovamd.net ~]# ssh root@server.astakhovamd.net
ssh: Could not resolve hostname server.astakhovamd.net: Name or service not known
[root@localhost.astakhovamd.net ~]# ssh root@server
ssh: Could not resolve hostname server: Name or service not known
[root@localhost.astakhovamd.net ~]#
```

Итог: Результат такой же.

11.2.2. Ограничение списка пользователей для удалённого доступа по SSH

Опять попытаюсь - тот же итог.

```
[root@localhost.astakhovamd.net ~]# ssh astakhovamd@server.astakhovamd.net
ssh: Could not resolve hostname server.astakhovamd.net: Name or service not known
[root@localhost.astakhovamd.net ~]#
```

В файле конфигурации добавляю строку:

```
#       AllowTcpForwarding no
#       PermitTTY no
#       ForceCommand cvs server

PermitRootLogin no
AllowUsers vagrant

^G Help       ^O Write Out  ^W Where
```

11.2.3. Настройка дополнительных портов для удалённого доступа по SSH

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

```
#
Port 22
Port 2022
#AddressFamily any
#ListenAddress 0.0.
```

Исправлю на сервере метки SELinux к порту 2022:

```
[root@localhost.astakhovamd.net ssh]#  semanage port-a-t ssh_port_t-p tcp 2022
usage: semanage [-h]
                {import,export,login,user,port,ibpkey,ibendport,interface,module,node,fcontext,boole
an,permissive,dontaudit}
                ...
semanage: error: argument subcommand: invalid choice: 'port-a-t' (choose from 'import', 'export', 'l
ogin', 'user', 'port', 'ibpkey', 'ibendport', 'interface', 'module', 'node', 'fcontext', 'boolean',
'permissive', 'dontaudit')
[root@localhost.astakhovamd.net ssh]#
```

```
6): Couldn't resolve host name for https://mirrors.fedoraproject.org/metalink?repo=epel-9&a
4&infra=$infra&content=pub/rocky [Could not resolve host: mirrors.fedoraproject.org]
[root@localhost.astakhovamd.net ~]# sudo semanage port -a -t ssh_port_t -p tcp 2022
[root@localhost.astakhovamd.net ~]# sudo semanage port -l | grep ssh
ssh_port_t                     tcp      2022, 22
[root@localhost.astakhovamd.net ~]#
```

С клиента попытаюсь получить доступ к серверу посредством SSH-соединения

```
ssh: Could not resolve hostname server.astakhovamd.net: Name or service not known
[astakhovamd@localhost.astakhovamd.net ~]$ ssh astakhovamd@localhost
The authenticity of host 'localhost (::1)' can't be established.
ED25519 key fingerprint is SHA256:yiD6CwwpAVWgmZ0LNexnVt25Q77Gy5XU7NfNuqLLz68.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'localhost' (ED25519) to the list of known hosts.
astakhovamd@localhost's password:
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Tue Dec  2 09:15:57 2025
[astakhovamd@localhost.astakhovamd.net ~]$
```

```
ED25519 key fingerprint is SHA256:yiD6CwwpAVWgmZ0LNexnVt25Q77Gy5XU7NfNuqLLz68.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[localhost]:2022' (ED25519) to the list of known hosts.
astakhovamd@localhost's password:
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Tue Dec  2 09:55:56 2025 from ::1
[astakhovamd@localhost.astakhovamd.net ~]$
```

11.2.4. Настройка удалённого доступа по SSH по ключу

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

```
PermitRootLogin no
AllowUsers vagrant
PubkeyAuthentication yes

^G Help      ^O Write Out ^W Where Is
```

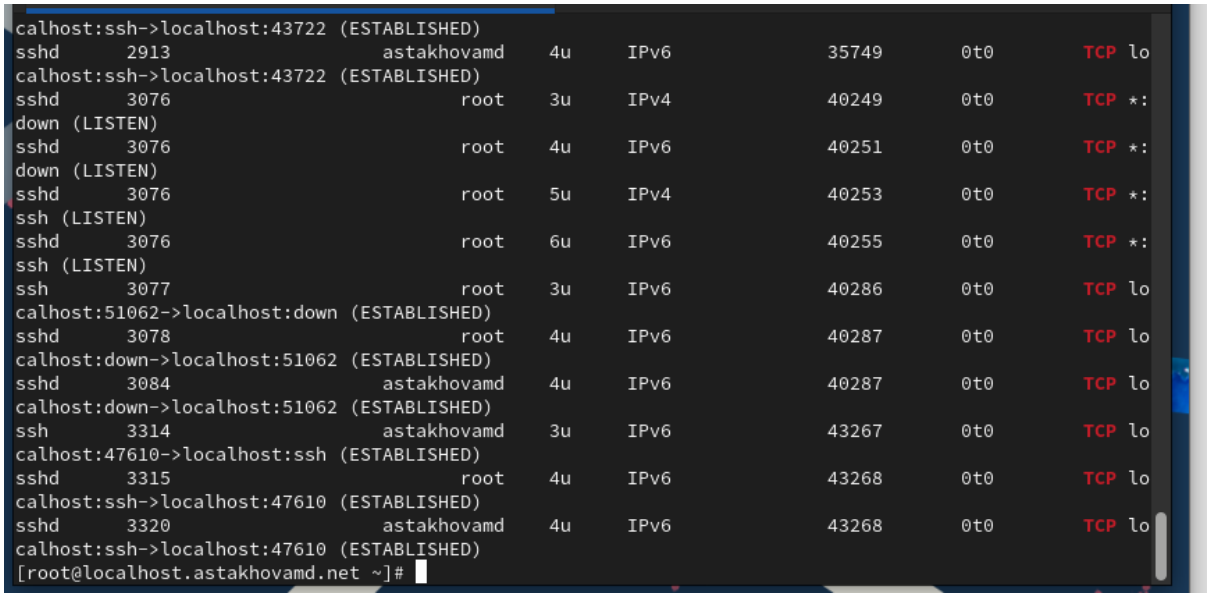
На клиенте сформирую SSH-ключ:

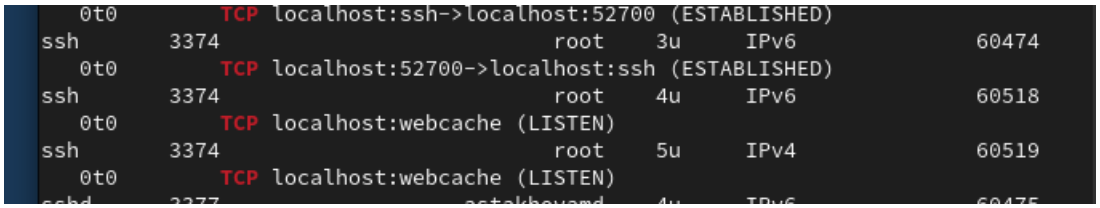
```
Last login: Tue Dec  2 09:55:56 2025 from ::1
[astakhovamd@localhost.astakhovamd.net ~]$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/astakhovamd/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/astakhovamd/.ssh/id_rsa
Your public key has been saved in /home/astakhovamd/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:XbDtc77idgC7uNSaGZkZFBKVG715Dq9x8M6xxR642Ts astakhovamd@localhost.astakhovamd.ne
t
The key's randomart image is:
+---[RSA 3072]----+
|      oooo.      |
|       .o..+     |
|        .o.oo    |
|       ...*o.    |
|        S .Xoo.  |
|         *o O++  |
|        *..B X.. |
|       ..=o O E. |
|        =. o.ooo |
+----[SHA256]-----+
[astakhovamd@localhost.astakhovamd.net ~]$
```

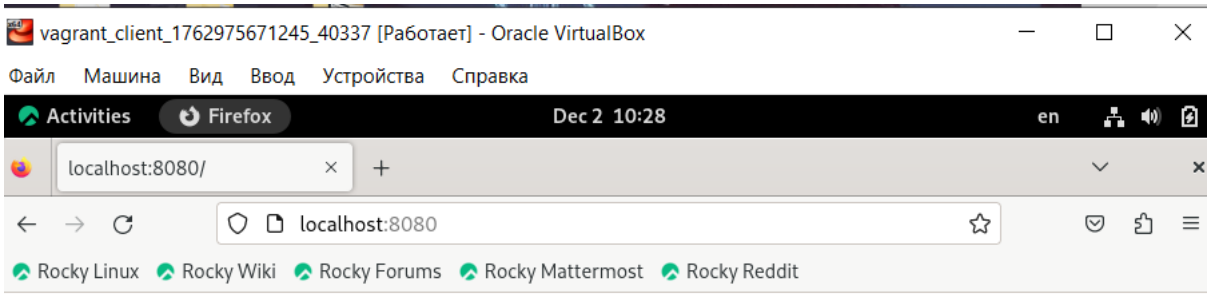
11.2.5. Организация туннелей SSH, перенаправление TCP-портов

Перенаправляю порт 80 на server.user.net на порт 8080 на локальной машине

```
calhost:ssh->localhost:47610 (ESTABLISHED)
[root@localhost.astakhovamd.net ~]# ssh -fNL 8080:localhost:80 astakhovamd@localhost.astakhovamd.net
The authenticity of host 'localhost.astakhovamd.net (::1)' can't be established.
ED25519 key fingerprint is SHA256:yiD6CwwpAVWgmZ0LNexnVt25Q77Gy5XU7NfNuqLLz68.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:1: [localhost]:2022
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'localhost.astakhovamd.net' (ED25519) to the list of known hosts.
astakhovamd@localhost.astakhovamd.net's password:
[root@localhost.astakhovamd.net ~]#
```







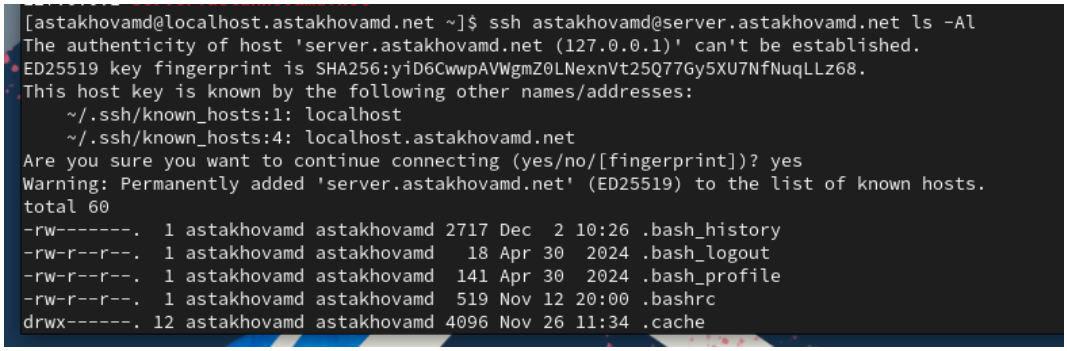
Welcome to the server.astakhovamd.net server

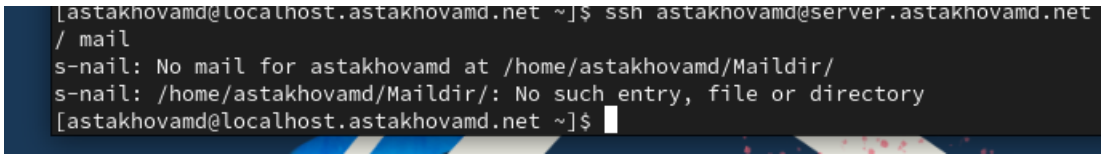
SSH tunnel is working!

Итог: туннель работает.

11.2.6. Запуск консольных приложений через SSH

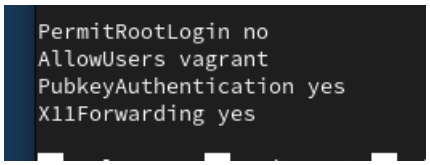
Посмотрю с клиента почту на сервере

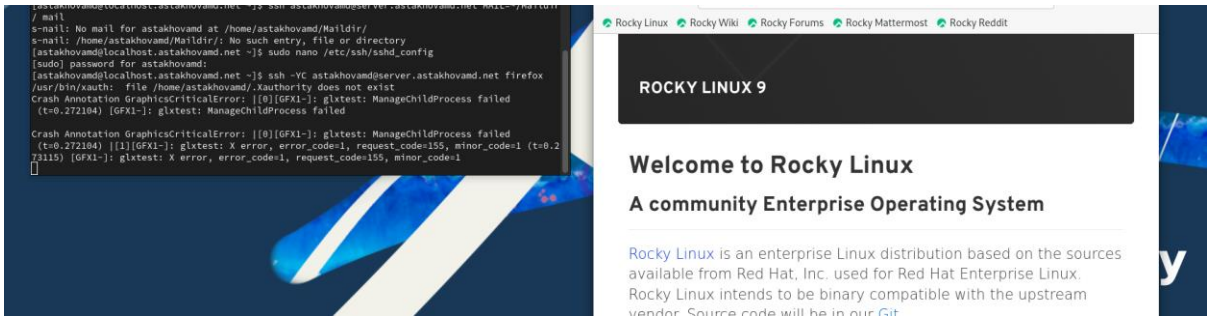




11.2.7. Запуск графических приложений через SSH (X11Forwarding)

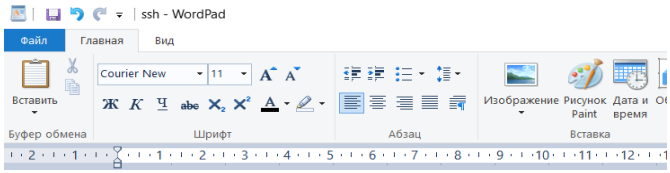
На сервере в конфигурационном файле /etc/ssh/sshd_config разрешу отображать на локальном клиентском компьютере графические интерфейсы X11:





11.2.8. Внесение изменений в настройки внутреннего окружения виртуальной машины

Вношу изменения в vagrantfile и завожу файл 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


                        path: "provision/server/mail.sh"

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

end
```

11.3. Итог работы

1. Безопасная базовая конфигурация:

1. Настроено ограничение доступа по SSH для пользователей: доступ для пользователя root был запрещён, а для пользователя alice — разрешён через параметры PermitRootLogin no и AllowUsers alice в файле /etc/ssh/sshd_config.

2. Отработана процедура применения изменений конфигурации с перезапуском службы sshd и проверкой корректности настроек.

2. Расширенные возможности доступа:

1. Освоена настройка SSH-сервера для работы на нескольких портах (22, 2222, 2022) одновременно, что позволяет обходить сетевые ограничения и повышает гибкость управления доступом.

2. Поняты практические сценарии использования нескольких портов: обход блокировок, распределение доступа, снижение видимости для автоматических атак.

3. Туннелирование и переадресация портов:

1. Приобретён навык создания SSH-туннелей с использованием ключей -f (фоновый режим) и -N (без выполнения команд), что позволяет организовывать безопасные соединения для других приложений.

2. Отработана техника локальной переадресации портов на примере перенаправления локального порта 5555 на порт 80 удалённого сервера через SSH-шлюз.

4. Интеграция с системами безопасности:

1. Освоена настройка SELinux для разрешения SSH-серверу использовать нестандартные порты (2022) через управление политиками портов с помощью утилиты semanage.

2. Приобретён практический опыт настройки межсетевого экрана (firewalld, iptables, ufw) для открытия необходимых SSH-портов, что обеспечивает корректную работу службы в защищённой сетевой среде.

5. Комплексное понимание безопасности:

1. Сформировано комплексное представление о многоуровневой защите SSH-доступа: на уровне конфигурации службы, систем разграничения доступа SELinux и сетевых фильтров.

2. Освоен процесс диагностики и проверки работоспособности всех компонентов: проверка открытых портов, анализ логов, тестирование подключений.

Вывод: В ходе работы были получены системные знания и практические умения по развёртыванию безопасного удалённого доступа через SSH, что позволяет эффективно администрировать серверы, обеспечивая при этом необходимый уровень защиты от несанкционированного доступа. Приобретённые навыки являются фундаментальными для системного администратора и DevOps-инженера.

11.4. Контрольные вопросы

1. Как запретить удалённый доступ по SSH для пользователя root и разрешить доступ пользователю alice

Открываю файл конфигурации SSH-сервера /etc/ssh/sshd_config в текстовом редакторе. Нахожу и изменяю параметр PermitRootLogin, устанавливаю его значение в no. Затем добавляю строчку AllowUsers alice,

чтобы разрешить доступ только пользователю alice. Сохраняю файл и перезапускаю службу SSH для применения изменений. Теперь подключиться по SSH под root напрямую будет невозможно, а пользователь alice получит доступ.

2. Как настроить удалённый доступ по SSH через несколько портов и зачем это нужно

Редактирую тот же конфигурационный файл SSH. В разделе с портами указываю несколько значений через отдельные строки Port, например Port 22, Port 2222, Port 2022.
После сохранения и перезапуска службы SSH открываю эти дополнительные порты в настройках межсетевого экрана.
Это может потребоваться для нескольких целей: чтобы обойти блокировку стандартного порта 22 в некоторых сетях, чтобы снизить количество автоматических атак на стандартный порт, для разделения доступа — например, один порт для администраторов, другой для обычных пользователей, или для тестирования новых конфигураций без риска потерять основное подключение.

3. Какие параметры SSH используются для создания туннеля в фоновом режиме без выполнения команд

Для создания такого туннеля использую ключ -f, который переводит SSH-соединение в фоновый режим после успешной аутентификации, и ключ -N, который указывает, что не нужно выполнять никакую удалённую команду — нужно только установить соединение для туннелирования. Основной ключ для туннеля, например -L для локальной переадресации, задаётся вместе с ними.

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

Выполняю в терминале команду SSH, используя ключ -L с указанием локального порта, адреса целевого сервера и его порта. Полная команда выглядиттак: **ssh -L 5555: server2.example.com: 80 имя_пользователя@промежуточный_сервер.** Если нужно, чтобы соединение ушло в фон, добавляю **-f -N**. Теперь, обращаясь на своём

компьютере к l**ocalhost:5555**, трафик будет безопасно перенаправляться через SSH-соединение на порт 80 сервера **server2.example.com.**

5. Как настроить SELinux, чтобы разрешить SSH работать с портом 2022

Сначала проверяю, какие порты уже разрешены для SSH в политиках SELinux с помощью команды semanage. Затем добавляю новый порт 2022 в контекст безопасности ssh_port_t с помощью команды semanage port -a. Это говорит SELinux, что службе SSH разрешено слушать и использовать порт 2022. После применения изменений перезапускаю службу SSH, и она сможет работать с этим портом.

6. Как настроить межсетевой экран для разрешения входящих SSH-подключений на порт 2022

В зависимости от системы, использую соответствующий инструмент. В системах с firewalld добавляю порт 2022 протокола TCP в постоянные правила и затем перезагружаю конфигурацию фаервола. В системах с ufw просто выполняю команду разрешения для этого порта. В системах с классическим iptables добавляю правило в цепочку INPUT, разрешающее новые и установленные соединения на порт 2022, и сохраняю правила. После настройки проверяю, что порт открыт и служба SSH на нём слушает.