

Лабораторная работа №11

Настройка безопасного удалённого доступа по протоколу SSH

Ромицына Анастасия Романовна

Содержание

1	Цель работы	6
2	Выполнение лабораторной работы	7
3	Выводы	22
4	Ответы на контрольные вопросы:	23
	Список литературы	25

Список иллюстраций

2.1	Открытие режима суперпользователя на виртуальной машине server и создание пароля для пользователя root.	7
2.2	Запуск в дополнительном терминале мониторинга системных событий.	8
2.3	Попытка получить с клиента доступ к серверу посредством SSH-соединения через пользователя root.	8
2.4	Открытие на сервере файла /etc/ssh/sshd_config конфигурации sshd для редактирования и запрет входа на сервер пользователю root.	9
2.5	Перезапуск sshd.	9
2.6	Повторная попытка получения доступа с клиента к серверу посредством SSH-соединения через пользователя root.	10
2.7	Попытка получить с клиента доступ к серверу посредством SSH-соединения через пользователя arromichina.	10
2.8	Открытие на сервере файла /etc/ssh/sshd_config конфигурации sshd на редактирование и добавление нужной строки.	10
2.9	Перезапуск sshd.	11
2.10	Повторная попытка получения доступа с клиента к серверу посредством SSH-соединения через пользователя arromichina.	11
2.11	Внесение изменения в файле /etc/ssh/sshd_config конфигурации sshd.	11
2.12	Перезапуск sshd и повторная попытка получить доступ с клиента к серверу посредством SSH-соединения через пользователя arromichina.	12
2.13	Добавление ниже строки Port записей в файле конфигурации sshd /etc/ssh/sshd_config на сервере.	12
2.14	Перезапуск sshd и просмотр расширенного статуса работы.	13
2.15	Исправление на сервере метки SELinux к порту 2022, открытие в настройках межсетевого порта 2022 протокола TCP, повторный перезапуск sshd и просмотр расширенного статуса его работы.	13
2.16	Попытка получить с клиента доступа к серверу посредством SSH-соединения через пользователя arromichina и получение доступа root.	14
2.17	Повторная попытка получения доступа с клиента к серверу посредством SSH-соединения через пользователя arromichina, указав порт 2022. Получение доступа root.	14

2.18	Настройка параметра на сервере в конфигурационном файле /etc/ssh/sshd_config, разрешающего аутентификацию по ключу. .	14
2.19	Перезапуск sshd.	15
2.20	Формирование на клиенте SSH-ключа и копирование открытого ключа на сервер.	15
2.21	Попытка получения доступа с клиента к серверу посредством SSH-соединения.	16
2.22	Просмотр на клиенте запущенных служб с протоколом TCP и перенаправление порта 80 на server.arromichina.net на порт 8080 на локальной машине.	16
2.23	Повторный просмотр на клиенте запущенных служб с протоколом TCP.	17
2.24	Запуск на клиенте браузера и ввод в адресной строке localhost:8080. .	18
2.25	Открытие на клиенте терминала под пользователем arromichina. Просмотр имени узла сервера, списка файлов на сервере и почты на сервере.	19
2.26	Разрешение отображать на сервере в конфигурационном файле /etc/ssh/sshd_config на локальном клиентском компьютере графические интерфейсы X11.	20
2.27	Перезапуск sshd.	20
2.28	Попытка с клиента удалённо подключиться к серверу и запустить графическое приложение firefox.	20
2.29	Переход на виртуальной машине server в каталог для внесения изменений в настройки внутреннего окружения /vagrant/provision/server/, создание в нём каталога ssh, в который поместили в соответствующие подкаталоги конфигурационный файл sshd_config. Создание в каталоге /vagrant/provision/server исполняемого файла ssh.sh. .	21
2.30	Открытие файла на редактирование и написание в нём скрипта. .	21
2.31	Редактирование конфигурационного файла Vagrantfile.	21

Список таблиц

1 Цель работы

Целью данной работы является приобретение практических навыков по настройке удалённого доступа к серверу с помощью SSH.

2 Выполнение лабораторной работы

На сервере зададим пароль для пользователя root: passwd root(рис. 2.1).

```
[root@server.arromichina.net ~]# passwd root
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: password updated successfully
```

Рис. 2.1: Открытие режима суперпользователя на виртуальной машине server и создание пароля для пользователя root.

На сервере в дополнительном терминале запустим мониторинг системных событий (рис. 2.2).

```
root@server:~ - sudo -i

ead (libc.so.6 + 0x94b68)
(libc.so.6 + 0x1056bc)

libc.so.6 + 0x1034bd)
+ 0x0)
+ 0x0)
+ 0x0)
art_call_main (libc.so.6 + 0x2a30e)
art_main@@GLIBC_2.34 (libc.so.6 + 0x2a3c9)
+ 0x0)

AMD x86-64
Subject: Process 13325 (VBoxClient) dumped core
Defined-By: systemd
Support: https://wiki.rockylinux.org/rocky/support
Documentation: man:core(5)

Process 13325 (VBoxClient) crashed and dumped core.

This usually indicates a programming error in the crashing program and
should be reported to its vendor as a bug.
Nov 14 14:07:40 server.arromichina.net systemd[1]: systemd-coredump@54-13329-0.service: Deactiva
ted successfully.
Subject: Unit succeeded
Defined-By: systemd
Support: https://wiki.rockylinux.org/rocky/support

The unit systemd-coredump@54-13329-0.service has successfully entered the 'dead' state.
```

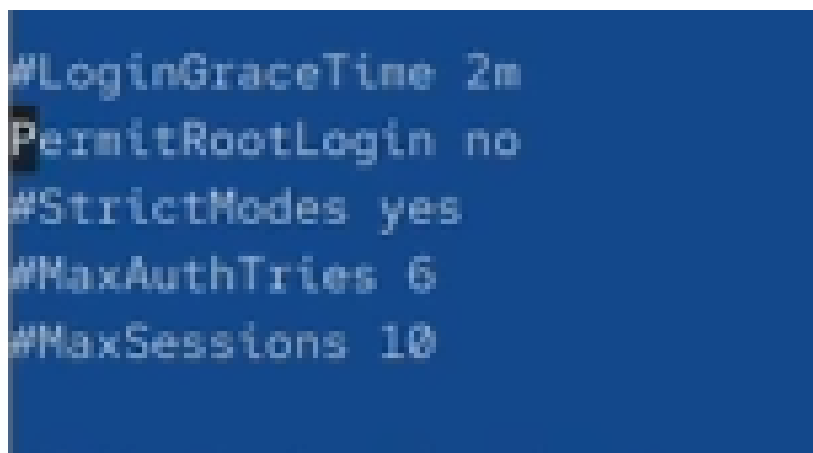
Рис. 2.2: Запуск в дополнительном терминале мониторинга системных событий.

С клиента попытаемся получить доступ к серверу посредством SSH-соединения через пользователя root(рис. 2.3).

```
arromichina@client.arromichina.net ~]$ ssh root@server.arromichina.net
The authenticity of host 'server.arromichina.net (192.168.1.1)' can't be esta
blished.
ED25519 key fingerprint is SHA256:LUBVKU3m9LeYkjemiPx1VvFkq9Eq9laMvpgqUuTCjHA
.
This host key is known by the following other names/addresses:
  ~/.ssh/known_hosts:1: [server.arromichina.net]:2022
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added 'server.arromichina.net' (ED25519) to the list of
known hosts.
root@server.arromichina.net's password:
Permission denied, please try again.
root@server.arromichina.net's password: █
```

Рис. 2.3: Попытка получить с клиента доступ к серверу посредством SSH-соединения через пользователя root.

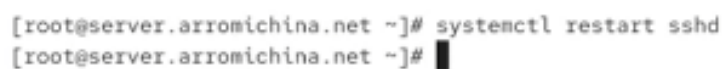
На сервере откроем файл /etc/ssh/sshd_config конфигурации sshd для редактирования и запретим вход на сервер пользователю root(рис. 2.4).

A screenshot of a terminal window with a blue background. It displays the contents of the /etc/ssh/sshd_config file. The visible lines are: #LoginGraceTime 2m, PermitRootLogin no, #StrictModes yes, #MaxAuthTries 6, and #MaxSessions 10. A cursor is visible at the end of the PermitRootLogin line.

```
#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
```

Рис. 2.4: Открытие на сервере файла /etc/ssh/sshd_config конфигурации sshd для редактирования и запрет входа на сервер пользователю root.

После сохранения изменений в файле конфигурации перезапустим sshd (рис. 2.5).

A screenshot of a terminal window showing the command 'systemctl restart sshd' being executed. The prompt is '[root@server.arromichina.net ~]#'. The output line shows the same prompt followed by a cursor.

```
[root@server.arromichina.net ~]# systemctl restart sshd
[root@server.arromichina.net ~]#
```

Рис. 2.5: Перезапуск sshd.

Повторим попытку получения доступа с клиента к серверу посредством SSH-соединения через пользователя root (рис. 2.6).

```
[arromichina@client.arromichina.net ~]$ ssh root@server
The authenticity of host 'server (192.168.1.1)' can't be established.
ED25519 key fingerprint is SHA256:LUBVKU3m9LeYkjemIPx1VkJkq9Eq9laMvpqgUuTCjHA
.
This host key is known by the following other names/addresses:
  ~/.ssh/known_hosts:1: [server.arromichina.net]:2022
  ~/.ssh/known_hosts:4: server.arromichina.net
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'server' (ED25519) to the list of known hosts.
root@server's password:
Permission denied, please try again.
root@server's password:
Permission denied, please try again.
root@server's password:

[arromichina@client.arromichina.net ~]$
```

Рис. 2.6: Повторная попытка получения доступа с клиента к серверу посредством SSH-соединения через пользователя root.

С клиента попытаемся получить доступ к серверу посредством SSH-соединения через пользователя arromichina (рис. 2.7).

```
[arromichina@client.arromichina.net ~]$ ssh arromichina@server.arromichina.net
arromichina@server.arromichina.net's password:
Web console: https://server.arromichina.net:9090/ or https://10.0.2.15:9090/

Last login: Fri Nov 14 14:02:56 2025
[arromichina@server.arromichina.net ~]$
```

Рис. 2.7: Попытка получить с клиента доступ к серверу посредством SSH-соединения через пользователя arromichina.

На сервере откроем файл /etc/ssh/sshd_config конфигурации sshd на редактирование и добавим строку (рис. 2.8).

```
AllowUsers vagrant
# no default banner path
#Banner none
```

Рис. 2.8: Открытие на сервере файла /etc/ssh/sshd_config конфигурации sshd на редактирование и добавление нужной строки.

После сохранения изменений в файле конфигурации перезапустим sshd (рис.

2.9).

```
[root@server.arromichina.net ~]# systemctl restart sshd
[root@server.arromichina.net ~]#
```

Рис. 2.9: Перезапуск sshd.

Повторим попытку получения доступа с клиента к серверу посредством SSH-соединения через пользователя arromichina(рис. 2.10).

```
[arromichina@server.arromichina.net ~]$ ssh arromichina@server.arromichina.net
t
The authenticity of host 'server.arromichina.net (192.168.1.1)' can't be established.
ED25519 key fingerprint is SHA256:LUBVKU3m9LeYkjemtPx1VkFkq9Eq9laMvpgqUuTCjHA
.
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.arromichina.net' (ED25519) to the list of
known hosts.
arromichina@server.arromichina.net's password:
Permission denied, please try again.
arromichina@server.arromichina.net's password:
Permission denied, please try again.
arromichina@server.arromichina.net's password:
arromichina@server.arromichina.net: Permission denied (publickey,gssapi-keyex
,gssapi-with-mic,password).
```

Рис. 2.10: Повторная попытка получения доступа с клиента к серверу посредством SSH-соединения через пользователя arromichina.

В файле /etc/ssh/sshd_config конфигурации sshd внесём следующее изменение: AllowUsers vagrant arromichina(рис. 2.11).

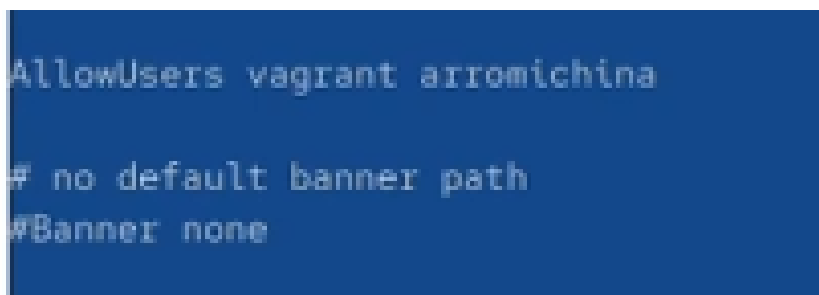
A screenshot of a terminal window with a blue background. The text is white and shows the configuration of the sshd service. The line 'AllowUsers vagrant arromichina' is highlighted. Below it, there are two commented-out lines: '# no default banner path' and '#Banner none'.

Рис. 2.11: Внесение изменения в файле /etc/ssh/sshd_config конфигурации sshd.

После сохранения изменений в файле конфигурации перезапустим sshd

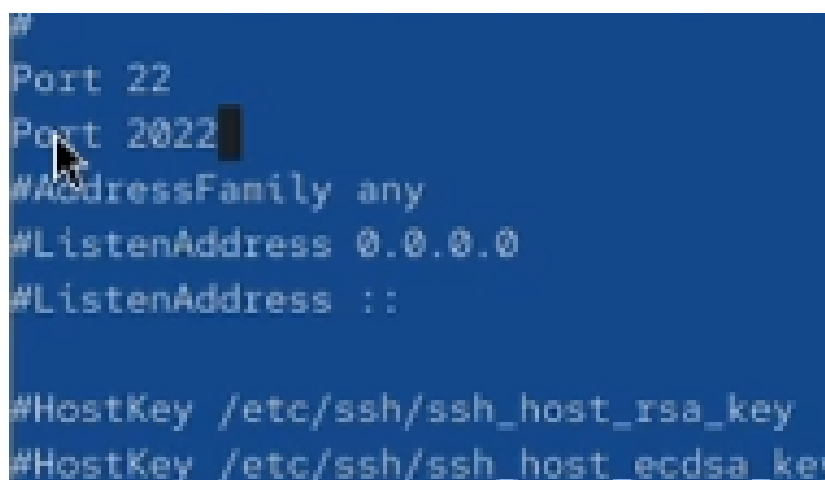
и вновь попытаемся получить доступ с клиента к серверу посредством SSH-соединения через пользователя arromichina(рис. 2.12).

```
[arromichina@client.arromichina.net ~]$ ssh arromichina@server.arromichina.net
arromichina@server.arromichina.net's password:
Web console: https://server.arromichina.net:9090/ or https://10.0.2.15:9090/

Last login: Sat Nov 15 00:58:40 2025 from 192.168.1.2
[arromichina@server.arromichina.net ~]$
```

Рис. 2.12: Перезапуск sshd и повторная попытка получить доступ с клиента к серверу посредством SSH-соединения через пользователя arromichina.

На сервере в файле конфигурации sshd /etc/ssh/sshd_config найдём строку Port и ниже этой строки добавим (рис. 2.13).



```
#
Port 22
Port 2022
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
```

Рис. 2.13: Добавление ниже строки Port записей в файле конфигурации sshd /etc/ssh/sshd_config на сервере.

После сохранения изменений в файле конфигурации перезапустим sshd: `systemctl restart sshd` И посмотрим расширенный статус работы: `systemctl status -l sshd` Система сообщила нам об отказе в работе sshd через порт 2022(рис. 2.14).

```

[root@server.arromichina.net ~]# systemctl restart sshd
[root@server.arromichina.net ~]# systemctl status -l sshd
● sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; preset: enabled)
   Active: active (running) since Fri 2025-11-14 14:29:51 UTC; 23s ago
  Invocation: 34c701f9b49c4efc8c2a60588deb249c
     Docs: man:sshd(8)
           man:sshd_config(5)
   Main PID: 16477 (sshd)
      Tasks: 1 (limit: 10398)
     Memory: 1M (peak: 1.2M)
        CPU: 28ms
   CGroup: /system.slice/ssh.service
           └─16477 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Nov 14 14:29:51 server.arromichina.net systemd[1]: Starting sshd.service - OpenSSH server daemon...
Nov 14 14:29:51 server.arromichina.net (sshd)[16477]: sshd.service: Referenced but unset environme
Nov 14 14:29:51 server.arromichina.net sshd[16477]: error: Bind to port 2022 on 0.0.0.0 failed:
Nov 14 14:29:51 server.arromichina.net sshd[16477]: error: Bind to port 2022 on :: failed: Permi
Nov 14 14:29:51 server.arromichina.net sshd[16477]: Server listening on 0.0.0.0 port 22.
Nov 14 14:29:51 server.arromichina.net systemd[1]: Started sshd.service - OpenSSH server daemon.
Nov 14 14:29:51 server.arromichina.net sshd[16477]: Server listening on :: port 22.
lines 1-20/20 (END)

```

Рис. 2.14: Перезапуск sshd и просмотр расширенного статуса работы.

Исправим на сервере метки SELinux к порту 2022: `semanage port -a -t ssh_port_t -p tcp 2022` В настройках межсетевого экрана откроем порт 2022 протокола TCP: `firewall-cmd --add-port=2022/tcp` `firewall-cmd --add-port=2022/tcp --permanent` Вновь перезапустим sshd и посмотрим расширенный статус его работы (статус показывает, что процесс sshd теперь прослушивает два порта) (рис. 2.15).

```

[root@server.arromichina.net ~]# semanage port -a -t ssh_port_t -p tcp 2022
[root@server.arromichina.net ~]# firewall-cmd --add-port=2022/tcp
success
[root@server.arromichina.net ~]# firewall-cmd --add-port=2022/tcp --permanent
success
[root@server.arromichina.net ~]# systemctl restart sshd
[root@server.arromichina.net ~]# systemctl status -l sshd
● sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; preset: enabled)
   Active: active (running) since Fri 2025-11-14 14:32:03 UTC; 16s ago
  Invocation: 65f3d1038c7a4d9a8076ca6ffcf5a6c5
     Docs: man:sshd(8)
           man:sshd_config(5)
   Main PID: 16828 (sshd)
      Tasks: 1 (limit: 10398)
     Memory: 1.3M (peak: 1.5M)
        CPU: 34ms
   CGroup: /system.slice/ssh.service
           └─16828 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Nov 14 14:32:03 server.arromichina.net systemd[1]: Starting sshd.service - OpenSSH server daemon...
Nov 14 14:32:03 server.arromichina.net (sshd)[16828]: sshd.service: Referenced but unset environment variable evaluated
Nov 14 14:32:03 server.arromichina.net sshd[16828]: Server listening on 0.0.0.0 port 2022.
Nov 14 14:32:03 server.arromichina.net sshd[16828]: Server listening on :: port 2022.
Nov 14 14:32:03 server.arromichina.net systemd[1]: Started sshd.service - OpenSSH server daemon.
Nov 14 14:32:03 server.arromichina.net sshd[16828]: Server listening on 0.0.0.0 port 22.
Nov 14 14:32:03 server.arromichina.net sshd[16828]: Server listening on :: port 22.
lines 1-20/20 (END)

```

Рис. 2.15: Исправление на сервере метки SELinux к порту 2022, открытие в настройках межсетевого порта 2022 протокола TCP, повторный перезапуск sshd и просмотр расширенного статуса его работы.

С клиента попытаемся получить доступ к серверу посредством SSH-

соединения через пользователя `arromichina`: `ssh arromichina@server.arromichina.net`
После открытия оболочки пользователя введём `sudo -i` для получения доступа `root` (рис. 2.16).

```
[arromichina@client.arromichina.net ~]$ ssh arromichina@server.arromichina.net
arromichina@server.arromichina.net's password:
Web console: https://server.arromichina.net:9090/ or https://10.0.2.15:9090/

Last login: Sat Nov 15 00:58:40 2025 from 192.168.1.2
[arromichina@server.arromichina.net ~]$ sudo -i
[sudo] password for arromichina:
[root@server.arromichina.net ~]#
```

Рис. 2.16: Попытка получить с клиента доступа к серверу посредством SSH-соединения через пользователя `arromichina` и получение доступа `root`.


Теперь повторим попытку получения доступа с клиента к серверу посредством SSH-соединения через пользователя `arromichina`, указав порт 2022: `ssh -p2022 arromichina@server.arromichina.net` После открытия оболочки пользователя введём `sudo -i` для получения доступа `root`(рис. 2.17).

```
[arromichina@client.arromichina.net ~]$ ssh -p2022 arromichina@server.arromichina.net
arromichina@server.arromichina.net's password:
Web console: https://server.arromichina.net:9090/ or https://10.0.2.15:9090/

Last login: Sat Nov 15 01:08:12 2025 from 192.168.1.2
[arromichina@server.arromichina.net ~]$ sudo -i
[sudo] password for arromichina:
[root@server.arromichina.net ~]#
```

Рис. 2.17: Повторная попытка получения доступа с клиента к серверу посредством SSH-соединения через пользователя `arromichina`, указав порт 2022. Получение доступа `root`.

На сервере в конфигурационном файле `/etc/ssh/sshd_config` зададим параметр, разрешающий аутентификацию по ключу: `PubkeyAuthentication yes` (рис. 2.18).



```
PubkeyAuthentication yes
```

Рис. 2.18: Настройка параметра на сервере в конфигурационном файле `/etc/ssh/sshd_config`, разрешающего аутентификацию по ключу.

После сохранения изменений в файле конфигурации перезапустим sshd (рис. 2.19).

```
[root@server.arromichina.net ~]# systemctl restart sshd
[root@server.arromichina.net ~]# █
```

Рис. 2.19: Перезапуск sshd.

На клиенте сформируем SSH-ключ, введя в терминале под пользователем arromichina ssh-keygen Далее скопируем открытый ключ на сервер, введя на клиенте (рис. 2.20).

```
[arromichina@client.arromichina.net ~]$ ssh-keygen
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/arromichina/.ssh/id_ed25519):
Enter passphrase for "/home/arromichina/.ssh/id_ed25519" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/arromichina/.ssh/id_ed25519
Your public key has been saved in /home/arromichina/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:K/3cUZMa2av0RZnU9EeMnSvCrz7svg4cT6xtUDdHzFE arromichina@client.arromichina.net
The key's randomart image is:
+--[ED25519 256]--+
|                +*E|
|                .oBo|
|                .. oo.+|
|                oo.=o=.|
|                So o= X |
|                ...B  * o |
|                . o+.++ o |
|                . oo* +  |
|                BBB  |
+-----[SHA256]-----+
[arromichina@client.arromichina.net ~]$ ssh-copy-id arromichina@server.arromichina.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 n
ow it is to install the new keys
arromichina@server.arromichina.net's password:

Number of key(s) added: 1

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

[arromichina@client.arromichina.net ~]$
```

Рис. 2.20: Формирование на клиенте SSH-ключа и копирование открытого ключа на сервер.

Попробуем получить доступ с клиента к серверу посредством SSH-соединения: ssh arromichina@server.arromichina.net Теперь мы проходим аутентификацию без ввода пароля для учётной записи удалённого пользователя (рис. 2.21).

```

[arromichina@client.arromichina.net ~]$ ssh arromichina@server.arromichina.net
Web console: https://server.arromichina.net:9090/ or https://10.0.2.15:9090/

Last login: Sat Nov 15 01:08:58 2025 from 192.168.1.2
[arromichina@server.arromichina.net ~]$

```

Рис. 2.21: Попытка получения доступа с клиента к серверу посредством SSH-соединения.

На клиенте посмотрим, запущены ли какие-то службы с протоколом TCP: `lsotf | grep TCP` После чего перенаправим порт 80 на `server.arromichina.net` на порт 8080 на локальной машине (рис. 2.22).

The screenshot shows a terminal window titled 'arromichina@client:~'. It displays the output of the `lsotf` command, which lists active network connections. The output is as follows:

Application	Local Address	Local Port	Peer Address	Peer Port	Protocol	State
firefox	1071	10111	sqladb:f~l	arromichina	109u	IPv4
0t0	TCP	client.arromichina.net:42384->199.232.41.91:https	(ESTABLISHED)			
firefox	9071	89761	BgIOThr~o	arromichina	47u	IPv4
0t0	TCP	client.arromichina.net:32816->93.243.107.34.bc.googleusercontent.c	(ESTABLISHED)			
m:https	(ESTABLISHED)					
firefox	9071	89761	BgIOThr~o	arromichina	94u	IPv4
0t0	TCP	client.arromichina.net:32830->93.243.107.34.bc.googleusercontent.c	(ESTABLISHED)			
m:https	(ESTABLISHED)					
firefox	9071	89761	BgIOThr~o	arromichina	109u	IPv4
0t0	TCP	client.arromichina.net:42384->199.232.41.91:https	(ESTABLISHED)			
firefox	9071	91717	DNS\x20Re	arromichina	47u	IPv4
0t0	TCP	client.arromichina.net:32816->93.243.107.34.bc.googleusercontent.c	(ESTABLISHED)			
m:https	(ESTABLISHED)					
firefox	9071	91717	DNS\x20Re	arromichina	94u	IPv4
0t0	TCP	client.arromichina.net:32830->93.243.107.34.bc.googleusercontent.c	(ESTABLISHED)			
m:https	(ESTABLISHED)					
firefox	9071	91717	DNS\x20Re	arromichina	109u	IPv4
0t0	TCP	client.arromichina.net:42384->199.232.41.91:https	(ESTABLISHED)			
firefox	9071	91720	DNS\x20Re	arromichina	47u	IPv4
0t0	TCP	client.arromichina.net:32816->93.243.107.34.bc.googleusercontent.c	(ESTABLISHED)			
m:https	(ESTABLISHED)					
firefox	9071	91720	DNS\x20Re	arromichina	94u	IPv4
0t0	TCP	client.arromichina.net:32830->93.243.107.34.bc.googleusercontent.c	(ESTABLISHED)			
m:https	(ESTABLISHED)					
firefox	9071	91720	DNS\x20Re	arromichina	109u	IPv4
0t0	TCP	client.arromichina.net:42384->199.232.41.91:https	(ESTABLISHED)			
firefox	9071	92551	Backgro~o	arromichina	47u	IPv4
0t0	TCP	client.arromichina.net:32816->93.243.107.34.bc.googleusercontent.c	(ESTABLISHED)			
m:https	(ESTABLISHED)					
firefox	9071	92551	Backgro~o	arromichina	94u	IPv4
0t0	TCP	client.arromichina.net:32830->93.243.107.34.bc.googleusercontent.c	(ESTABLISHED)			
m:https	(ESTABLISHED)					
firefox	9071	92551	Backgro~o	arromichina	109u	IPv4
0t0	TCP	client.arromichina.net:42384->199.232.41.91:https	(ESTABLISHED)			

Below the table, the terminal shows the execution of the `ssh -fNL 8080:localhost:80 arromichina@server.arromichina.net` command, which successfully establishes a port forwarding connection. The prompt returns to `[arromichina@client.arromichina.net ~]$`.

Рис. 2.22: Просмотр на клиенте запущенных служб с протоколом TCP и перенаправление порта 80 на `server.arromichina.net` на порт 8080 на локальной машине.

Вновь на клиенте посмотрим, запущены ли какие-то службы с протоколом

TCP (рис. 2.23).

```
arromichina@client:~  
+  
firefox 9071 89761 BgIOThr~o arromichina 94u IPv4 1290662  
0t0 TCP client.arromichina.net:32830->93.243.107.34.bc.googleusercontent.co  
m:https (ESTABLISHED)  
firefox 9071 89761 BgIOThr~o arromichina 109u IPv4 1290709  
0t0 TCP client.arromichina.net:42384->199.232.41.91:https (ESTABLISHED)  
firefox 9071 91717 DNS\x20Re arromichina 47u IPv4 1290068  
0t0 TCP client.arromichina.net:32816->93.243.107.34.bc.googleusercontent.co  
m:https (ESTABLISHED)  
firefox 9071 91717 DNS\x20Re arromichina 94u IPv4 1290662  
0t0 TCP client.arromichina.net:32830->93.243.107.34.bc.googleusercontent.co  
m:https (ESTABLISHED)  
firefox 9071 91717 DNS\x20Re arromichina 109u IPv4 1290709  
0t0 TCP client.arromichina.net:42384->199.232.41.91:https (ESTABLISHED)  
firefox 9071 91720 DNS\x20Re arromichina 47u IPv4 1290068  
0t0 TCP client.arromichina.net:32816->93.243.107.34.bc.googleusercontent.co  
m:https (ESTABLISHED)  
firefox 9071 91720 DNS\x20Re arromichina 94u IPv4 1290662  
0t0 TCP client.arromichina.net:32830->93.243.107.34.bc.googleusercontent.co  
m:https (ESTABLISHED)  
firefox 9071 91720 DNS\x20Re arromichina 109u IPv4 1290709  
0t0 TCP client.arromichina.net:42384->199.232.41.91:https (ESTABLISHED)  
firefox 9071 92551 Backgro~o arromichina 47u IPv4 1290068  
0t0 TCP client.arromichina.net:32816->93.243.107.34.bc.googleusercontent.co  
m:https (ESTABLISHED)  
firefox 9071 92551 Backgro~o arromichina 94u IPv4 1290662  
0t0 TCP client.arromichina.net:32830->93.243.107.34.bc.googleusercontent.co  
m:https (ESTABLISHED)  
firefox 9071 92551 Backgro~o arromichina 109u IPv4 1290709  
0t0 TCP client.arromichina.net:42384->199.232.41.91:https (ESTABLISHED)  
ssh 92812 arromichina 3u IPv4 1341197  
0t0 TCP client.arromichina.net:48574->server.arromichina.net:ssh (ESTABLISH  
ED)  
ssh 92812 arromichina 4u IPv6 1341201  
0t0 TCP localhost:webcache (LISTEN)  
ssh 92812 arromichina 5u IPv4 1341202  
0t0 TCP localhost:webcache (LISTEN)  
[arromichina@client.arromichina.net ~]$
```

Рис. 2.23: Повторный просмотр на клиенте запущенных служб с протоколом TCP.

На клиенте запустим браузер и в адресной строке введём localhost:8080. Убедимся, что отобразилась страница с приветствием «Welcome to the server.arromichina.net server» (рис. 2.24).

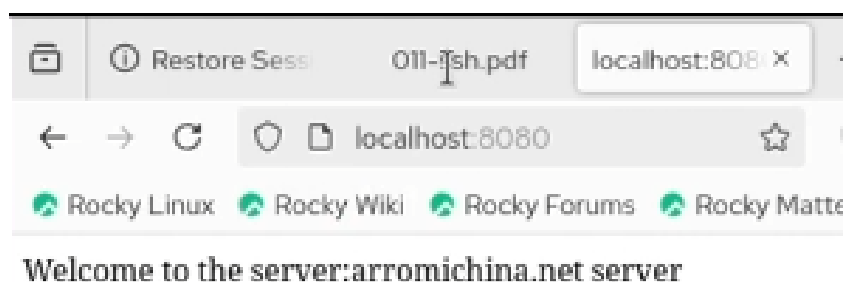


Рис. 2.24: Запуск на клиенте браузера и ввод в адресной строке localhost:8080.

На клиенте откроем терминал под пользователем arromichina и посмотрим с клиента имя узла сервера: `ssh arromichina@server.arromichina.net hostname` Посмотрим с клиента список файлов на сервере: `ssh arromichina@server.arromichina.net ls -Al` Посмотрим с клиента почту на сервере (рис. 2.25).

```
arromichina@client:~ -- ssh arromichina@server.arromichina.net M...
-rw-r--r--. 1 arromichina arromichina 144 Oct 29 2024 .bash_profile
-rw-r--r--. 1 arromichina arromichina 549 Sep 13 20:30 .bashrc
drwx-----, 14 arromichina arromichina 4096 Oct 3 18:28 .cache
drwx-----, 11 arromichina arromichina 4096 Oct 3 18:40 .config
drwxr-xr-x. 2 arromichina arromichina 6 Sep 22 12:09 Desktop
drwxr-xr-x. 2 arromichina arromichina 6 Sep 22 12:09 Documents
drwxr-xr-x. 2 arromichina arromichina 6 Sep 22 12:09 Downloads
drwx-----, 4 arromichina arromichina 32 Sep 22 12:10 .local
drwx-----, 5 arromichina arromichina 4096 Nov 5 19:31 Maildir
drwxr-xr-x. 5 arromichina arromichina 54 Sep 22 12:31 .mozilla
drwxr-xr-x. 2 arromichina arromichina 6 Sep 22 12:09 Music
drwxr-xr-x. 2 arromichina arromichina 6 Sep 22 12:09 Pictures
drwxr-xr-x. 2 arromichina arromichina 6 Sep 22 12:09 Public
drwx-----, 2 arromichina arromichina 48 Nov 15 01:15 .ssh
drwxr-xr-x. 2 arromichina arromichina 6 Sep 22 12:09 Templates
-rw-r-----, 1 arromichina arromichina 6 Nov 14 14:03 .vboxclient-clipboard-tty2-
control.pid
-rw-r-----, 1 arromichina arromichina 6 Nov 15 01:02 .vboxclient-clipboard-tty2-
service.pid
-rw-r-----, 1 arromichina arromichina 6 Nov 14 14:03 .vboxclient-draganddrop-tty
2-control.pid
-rw-r-----, 1 arromichina arromichina 6 Nov 14 14:03 .vboxclient-hostversion-tty
2-control.pid
-rw-r-----, 1 arromichina arromichina 6 Nov 14 14:03 .vboxclient-seamless-tty2-c
ontrol.pid
-rw-r-----, 1 arromichina arromichina 6 Nov 14 14:03 .vboxclient-vmvga-session-
tty2-control.pid
-rw-r-----, 1 arromichina arromichina 6 Nov 14 14:03 .vboxclient-vmvga-session-
tty2-service.pid
drwxr-xr-x. 2 arromichina arromichina 6 Sep 22 12:09 Videos
[arromichina@client.arromichina.net ~]$ ssh arromichina@server.arromichina.net MAIL=
~/Maildir/ mail
s-nail version v14.9.24. Type '?' for help
/home/arromichina/Maildir: 2 messages
▶ 1 An Romitsina 2025-10-28 11:40 18/674 "test1"
  2 An Romitsina 2025-10-28 11:43 18/673 "test3"
```

Рис. 2.25: Открытие на клиенте терминала под пользователем arromichina. Просмотр имени узла сервера, списка файлов на сервере и почты на сервере.

На сервере в конфигурационном файле /etc/ssh/sshd_config разрешим отображать на локальном клиентском компьютере графические интерфейсы X11 (рис. 2.26).

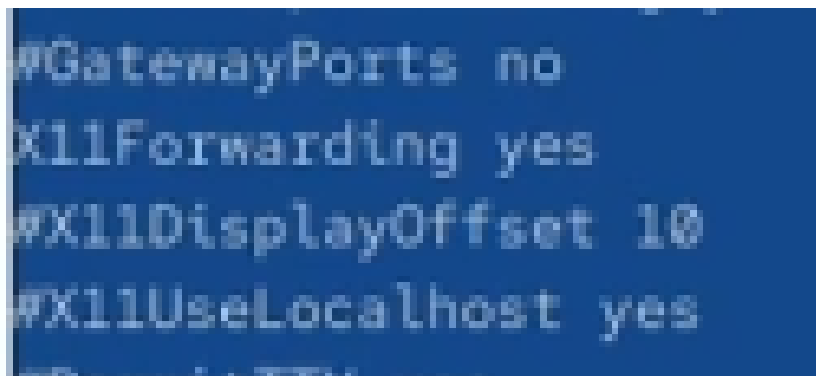


Рис. 2.26: Разрешение отображать на сервере в конфигурационном файле /etc/ssh/sshd_config на локальном клиентском компьютере графические интерфейсы X11.

После сохранения изменения в конфигурационном файле перезапустим sshd (рис. 2.27).

```
[root@server.arromichina.net ~]# systemctl restart sshd  
[root@server.arromichina.net ~]#
```

Рис. 2.27: Перезапуск sshd.

Попробуем с клиента удалённо подключиться к серверу и запустить графическое приложение firefox (рис. 2.28).

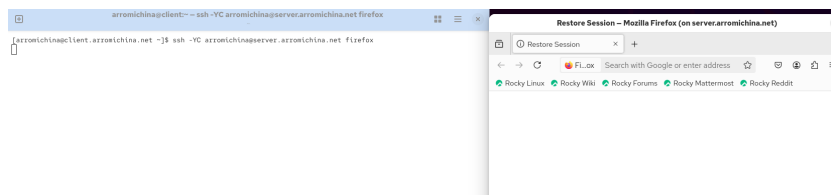


Рис. 2.28: Попытка с клиента удалённо подключиться к серверу и запустить графическое приложение firefox.

На виртуальной машине server перейдём 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. В каталоге /vagrant/provision/server создадим исполняемый файл ssh.sh (рис. 2.29).

```
[root@server.arrowichina.net ~]# cd /vagrant/provision/server
[root@server.arrowichina.net server]# mkdir -p /vagrant/provision/server/ssh/etc/ssh
[root@server.arrowichina.net server]# cp -R /etc/ssh/sshd_config /vagrant/provision/server/ssh/etc/ssh/
[root@server.arrowichina.net server]# cd /vagrant/provision/server
bash: d: command not found...
[root@server.arrowichina.net server]# cd /vagrant/provision/server
[root@server.arrowichina.net server]# touch ssh.sh
[root@server.arrowichina.net server]# chmod +x ssh.sh
[root@server.arrowichina.net server]#
```

Рис. 2.29: Переход на виртуальной машине server в каталог для внесения изменений в настройки внутреннего окружения /vagrant/provision/server/, создание в нём каталога ssh, в который поместили в соответствующие подкаталоги конфигурационный файл sshd_config. Создание в каталоге /vagrant/provision/server исполняемого файла ssh.sh.

Открыв его на редактирование, пропишем в нём скрипт из лабораторной работы (рис. 2.30).

```
ssh.sh [-M--] 22 L:[ 1*11 12/ 12] *(360 / 360b) <EOF>
#!/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. 2.30: Открытие файла на редактирование и написание в нём скрипта.

Для отработки созданного скрипта во время загрузки виртуальной машины server в конфигурационном файле Vagrantfile добавим в разделе конфигурации для сервера (рис. 2.31).

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

Рис. 2.31: Редактирование конфигурационного файла Vagrantfile.

3 Выводы

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

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 конфигурационном файле SSH `/etc/ssh/sshd_config`: # Запрет удалённого доступа пользователю root `PermitRootLogin no` # Разрешение доступа пользователю alice `AllowUsers alice` После внесения изменений, необходимо перезапустить службу SSH: `sudo service ssh restart`
2. Как настроить удалённый доступ по SSH через несколько портов? Для чего это может потребоваться? – В конфигурационном файле `/etc/ssh/sshd_config` добавьте строки: # Первый порт (по умолчанию 22) `Port 22` # Второй порт `Port 2222` После изменений перезапустите службу SSH. Это может быть полезно для повышения безопасности, а также для избежания конфликтов с другими службами, использующими порт 22.
3. Какие параметры используются для создания туннеля SSH, когда команда `ssh` устанавливает фоновое соединение и не ожидает какой-либо конкретной команды? – `ssh -N -f -L local_port:destination_host:remote_port user@ssh_server` -N: Не выполнять команду на удаленном хосте. -f: Перевести ssh в фоновый режим после установки туннеля.
4. Как настроить локальную переадресацию с локального порта 5555 на порт 80 сервера `server2.example.com`? – `ssh -L 5555:server2.example.com:80 user@ssh_server` Теперь, при подключении к локальному порту 5555, трафик будет перенаправляться через SSH к порту 80 на сервере

server2.example.com.

5. Как настроить SELinux, чтобы позволить SSH связываться с портом 2022?
– `sudo semanage port -a -t ssh_port_t -p tcp 2022` Данная команда добавляет правило SELinux, разрешая использование порта 2022 для сервиса ssh.
6. Как настроить межсетевой экран на сервере, чтобы разрешить входящие подключения по SSH через порт 2022? – `sudo firewall-cmd --permanent --add-port=2022/tcp` `sudo firewall-cmd --reload` Эти команды добавляют правило в межсетевой экран для разрешения входящих подключений по SSH через порт 2022 и перезагружают конфигурацию межсетевого экрана.

Список литературы