

Липецкий государственный технический университет

Факультет автоматизации и информатики

Кафедра автоматизированных систем управления

ЛАБОРАТОРНАЯ РАБОТА №7

по дисциплине «Операционная система Linux»

Работа с SSH

Студент

Посаднев В.В.

Группа АС-18

Руководитель

Кургасов В.В.

Липецк 2021 г.

Цель работы

Приобрести практические навыки работы с программным обеспечением удаленного доступа к распределенным системам обработки данных.

Задание кафедры

1. Создать подключение удаленного доступа к системе обработки данных по указанному логину и паролю.

2. Выполнить подключение с использованием полноэкранного консольного оконного менеджера `screen` и запустить какую-либо долгую операцию.

3. Сформировать шифрованные ключи и произвести их обмен с удаленной системой. Продемонстрировать успешное подключение без пароля и наличие публичного ключа на удаленной системе.

4. Запустить терминал с командной оболочкой ОС и ввести команду `tmux` (терминальный мультиплексор). Комбинациями клавиш `Ctrl-B + C` создать нового окно и запустить анализатор трафика `tcpdump` с фильтром пакетов получаемых и передаваемых от узла `domen.name` с TCP-портом источника и назначения 23. С помощью команды `tee`, вывести отфильтрованные IP-пакеты на терминал и сохранить данные в файл `telnet.log`, в домашнем каталоге пользователя.

5. В первом окне терминального мультиплексора попытаться установить соединение с удаленным сервером `domen.name` по протоколу TELNET.

6. Подключившись к удаленной системе ввести пароль и выполнить команду `uname -a`, выведя тем самым информацию об удаленной системе. Для разрыва соединения использовать команду `logout`.

7. Прервать фильтрацию пакетов сетевым анализатором `tcpdump`, воспользовавшись комбинацией `Ctrl-C`. В файле `telnet.log` выделить записи установления и разрыва соединения с сервером TELNET.

8. Снова запустить анализатор сетевого трафика с фильтром пакетов получаемых и передаваемых узлу `domen.name` с TCP-портом источника и назначения 22. С помощью команды `tee`, вывести отфильтрованные IP-пакеты на терминал и сохранить данные в файл `ssh.log`, в домашнем каталоге пользователя.

9. Переключившись на первое окно терминального мультиплексора, с помощью команды `ssh -l student domen.name` попытаться установить зашифрованное соединение с удаленным сервером `domen.name`. Проследить передачу и прием пакетов между узлами в окне сетевого анализатора.

10. Подключившись к удаленной системе ввести пароль и выполнить команду `uname -a`, выведя информацию об удаленной системе.

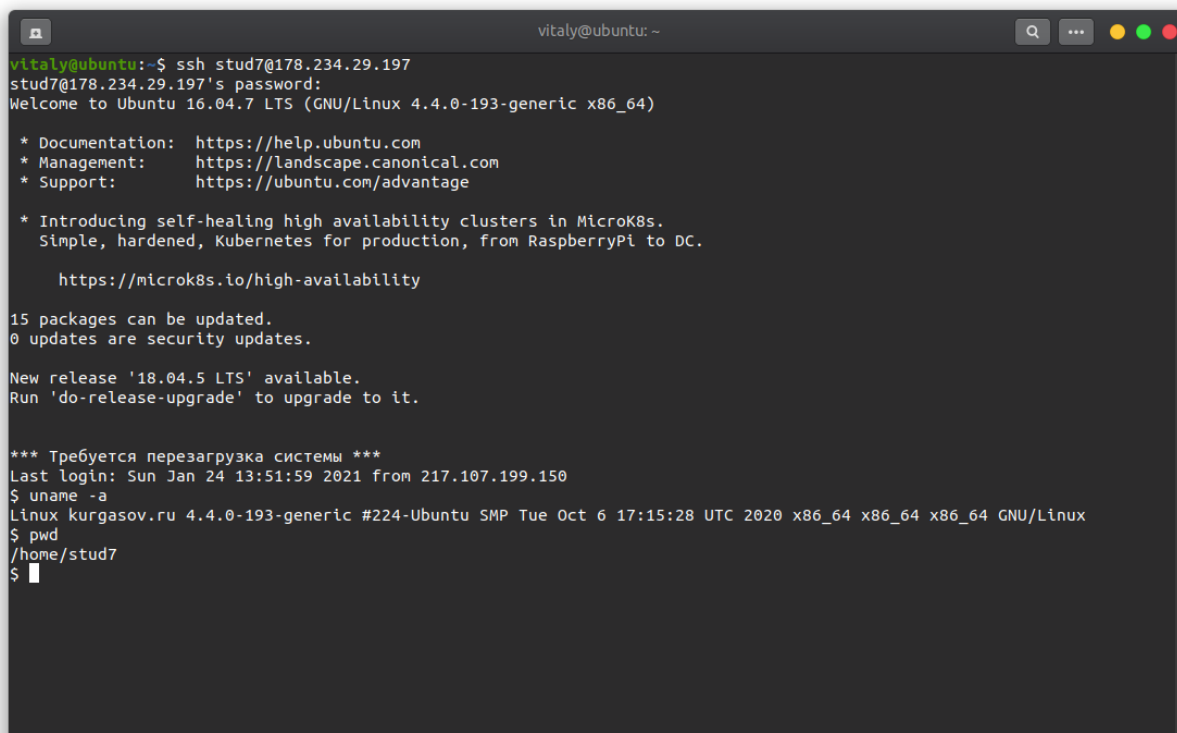
11. Создать текстовый файл с содержанием ФИО и номера лабораторной работы на локальном узле и с помощью команды `scp` передать его по зашифрованному каналу на удаленную систему. Проверить наличие копии переданного файла на удаленном узле.

12. Проверить состояние сессии `screen` и выполнение ранее указанной в ней команды.

13. Остановить анализатор сетевых пакетов, воспользовавшись комбинацией `Ctrl-C`.

Ход работы

Подключимся к удаленному хосту с помощью команды `ssh stud7@178.234.29.197`. После выполнения данной команды сервер запросит пароль для указанного пользователя (stud7), при успешной авторизации мы попадаем в домашнюю директорию пользователя аналогично той, что представлена на рисунке 1.



```
vitaly@ubuntu: ~  
vitaly@ubuntu:~$ ssh stud7@178.234.29.197  
stud7@178.234.29.197's password:  
Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 4.4.0-193-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage  
  
* Introducing self-healing high availability clusters in MicroK8s.  
  Simple, hardened, Kubernetes for production, from RaspberryPi to DC.  
    https://microk8s.io/high-availability  
  
15 packages can be updated.  
0 updates are security updates.  
  
New release '18.04.5 LTS' available.  
Run 'do-release-upgrade' to upgrade to it.  
  
*** Требуется перезагрузка системы ***  
Last login: Sun Jan 24 13:51:59 2021 from 217.107.199.150  
$ uname -a  
Linux kurgasov.ru 4.4.0-193-generic #224-Ubuntu SMP Tue Oct 6 17:15:28 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux  
$ pwd  
/home/stud7  
$
```

Рисунок 1 – Успешное подключение к серверу с использованием логина и пароля

Откроем новую сессию оконного менеджера `screen` с помощью команды `screen -S ping_ya_ru`, в данной команде `ping_ya_ru` указан в виде названия новой сессии. После выполнения данной команды перед нами откроется новое пустое окно, с которым можно взаимодействовать также как и с обычным. После отключения от сервера сессия оконного менеджера останется незавершенной и будет также выполнять свою работу. Запустим в новой сессии команду `ping ya.ru`, которая будет проверять доступность соединения между сервером и сайтом `ya.ru`. Пример активной проверки соединения представлено на рисунке 2.

```
vitaly@ubuntu: ~  
$ ping ya.ru  
PING ya.ru (87.250.250.242): 56(84) bytes of data.  
64 bytes from ya.ru (87.250.250.242): icmp_seq=1 ttl=249 time=24.3 ms  
64 bytes from ya.ru (87.250.250.242): icmp_seq=2 ttl=249 time=24.3 ms  
64 bytes from ya.ru (87.250.250.242): icmp_seq=3 ttl=249 time=24.2 ms  
64 bytes from ya.ru (87.250.250.242): icmp_seq=4 ttl=249 time=24.2 ms  
64 bytes from ya.ru (87.250.250.242): icmp_seq=5 ttl=249 time=24.2 ms  
64 bytes from ya.ru (87.250.250.242): icmp_seq=6 ttl=249 time=25.1 ms  
64 bytes from ya.ru (87.250.250.242): icmp_seq=7 ttl=249 time=27.1 ms  
64 bytes from ya.ru (87.250.250.242): icmp_seq=8 ttl=249 time=24.1 ms  
64 bytes from ya.ru (87.250.250.242): icmp_seq=9 ttl=249 time=24.2 ms  
64 bytes from ya.ru (87.250.250.242): icmp_seq=10 ttl=249 time=24.1 ms  
64 bytes from ya.ru (87.250.250.242): icmp_seq=11 ttl=249 time=24.2 ms
```

Рисунок 2 – Проверка доступности сервиса ya.ru

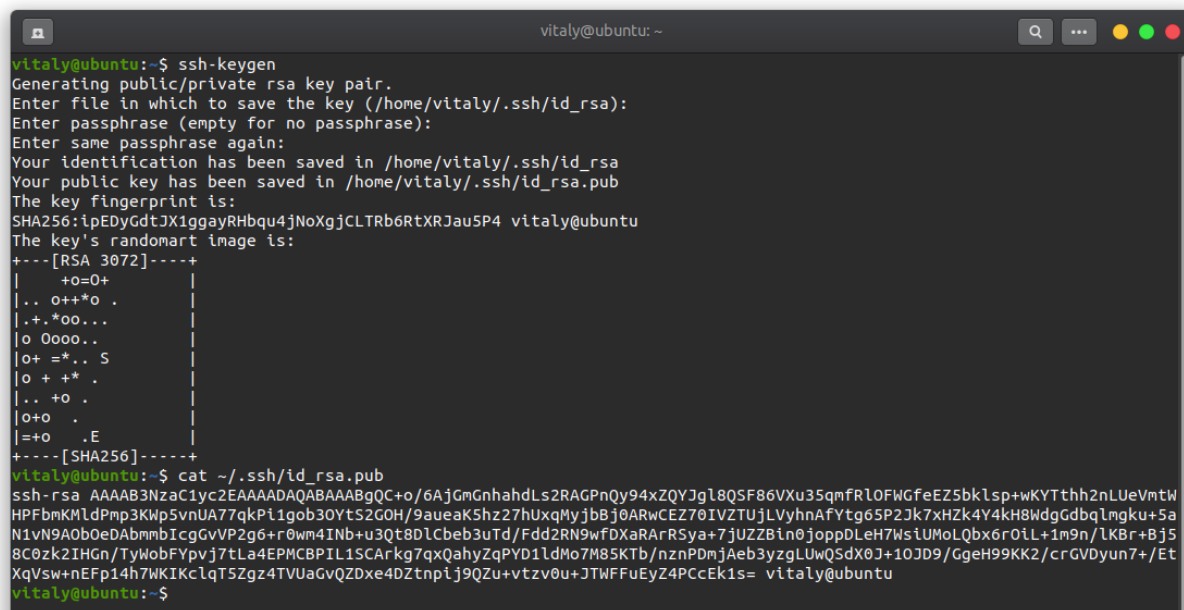
Для отключения от окна сессии `screen` необходимо воспользоваться комбинацией клавиш `Ctrl-A + D`. Для просмотра активных сессий `screen` можно воспользоваться командой `screen -li`. Пример выполнения данных действий представлено на рисунке 3.

```
vitaly@ubuntu: ~  
vitaly@ubuntu:~$ ssh stud7@178.234.29.197  
stud7@178.234.29.197's password:  
Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 4.4.0-193-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage  
  
* Introducing self-healing high availability clusters in MicroK8s.  
  Simple, hardened, Kubernetes for production, from RaspberryPi to DC.  
  https://microk8s.io/high-availability  
  
15 packages can be updated.  
0 updates are security updates.  
  
New release '18.04.5 LTS' available.  
Run 'do-release-upgrade' to upgrade to it.  
  
*** Требуется перезагрузка системы ***  
Last login: Sun Jan 24 13:52:42 2021 from 217.107.199.150  
$ screen -S ping_ya_ru  
[detached from 19302.ping_ya_ru]  
$ screen -li  
-sh: 2: ы: not found  
$ screen -li  
There is a screen on:  
19302.ping_ya_ru      (24.01.2021 13:54:13)  (Detached)  
1 Socket in /var/run/screen/S-stud7.  
$
```

Рисунок 3 – Отключение от сессии `screen` и просмотр списка активных сессий

Следующим шагом нам необходимо сгенерировать новый `ssh` ключ. Для этого воспользуемся командой `ssh-keygen` и заполним необходимые данные. После успешной генерации у нас будет два ключа: публичный и приватный. Просмотрим публичный ключ, который находится по следующему пути `~/.ssh/id_rsa.pub`, с помощью команды `cat`. Результат выполнения данных

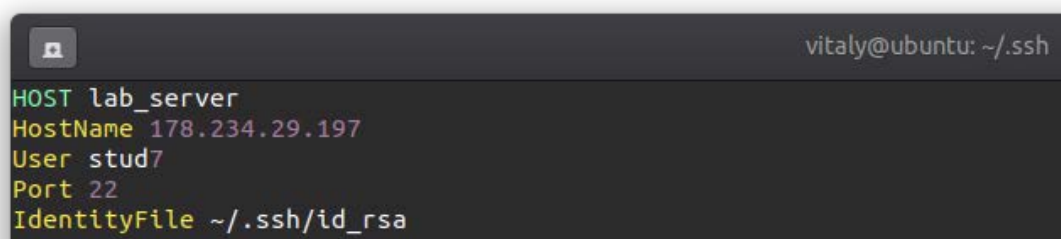
действий представлен на рисунке 4.



```
vitaly@ubuntu: ~  
vitaly@ubuntu:~$ ssh-keygen  
Generating public/private rsa key pair.  
Enter file in which to save the key (/home/vitaly/.ssh/id_rsa):  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /home/vitaly/.ssh/id_rsa  
Your public key has been saved in /home/vitaly/.ssh/id_rsa.pub  
The key fingerprint is:  
SHA256:ipEDyGdtJX1ggayRhbqu4jNoXgjCLTRb6RtXRJau5P4 vitaly@ubuntu  
The key's randomart image is:  
+---[RSA 3072]-----+  
|  
|.. o++*o .  
|..*oo...  
|o Oooo..  
|o+ =*.. S  
|o + * .  
|.. +o .  
|o+ .  
|..+o .E  
+---[SHA256]-----+  
vitaly@ubuntu:~$ cat ~/.ssh/id_rsa.pub  
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGC+o/6AjGmGnhahdLs2RAGPnQy94xZQYJg18QSF86VXu35qmFRLOFWGfeEZ5bklsp+wKYTthh2nLueVmtW  
HPFbmKMLdPmp3KwPsvnUA77qkPi1gob30YtS2G0H/9aueaK5hz27huxqMyjbbj0ARwCEZ70IVZTUjLVyhnAfYtg65P2Jk7xHZk4Y4kH8wdgGdbqlmgku+5a  
N1vN9A0b0eDAbmmbIcgGvVP2g6+r0wm4INb+u3Qt8DLCbeb3uTd/Fdd2RN9wFDXaRARSya+7jUZZBin0joppDLH7WsiUMoLQbx6r0iL+1m9n/lKBr+8j5  
8C0zk2IHGn/TyWobFYpvj7tLa4EPMCBPIL1SCArk7qxQahyZqPYD1ldMo7M8SKTb/nznPDmjAeb3yzgLUwQ5dX0J+10JD9/GgeH99KK2/crGVdyun7+/Et  
XqVsw+nEFp14h7WKIKclqT5Zgz4TVUaGvQZDxe40Ztnpij9QZu+vtzv0u+JTWFFuEYz4PccEk1s= vitaly@ubuntu  
vitaly@ubuntu:~$
```

Рисунок 4 – Пример создания и просмотра нового ключа доступа

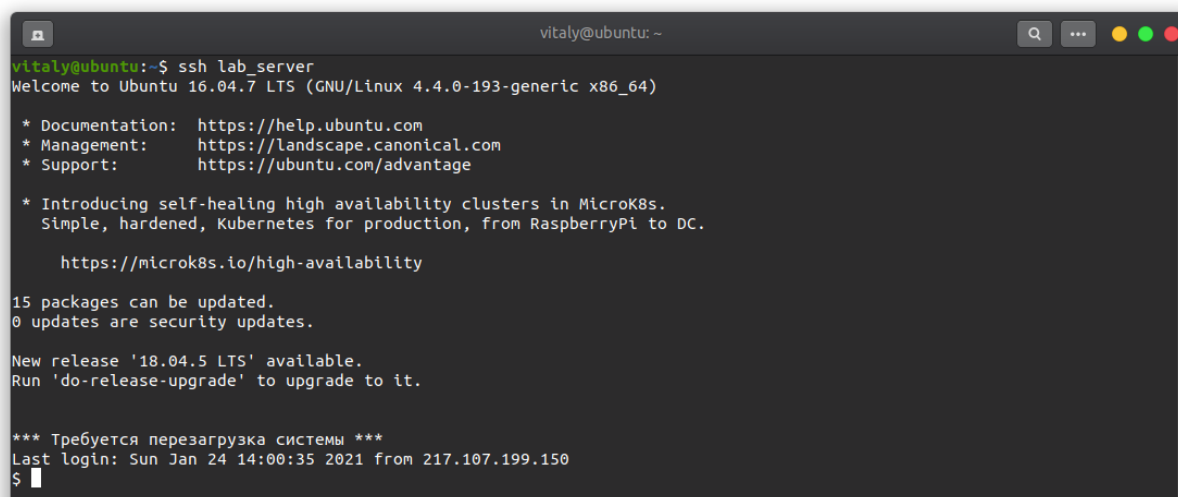
Теперь отправим полученный ключ на удаленный хост. Для этого необходимо воспользоваться следующей командой: `ssh-copy-id stud7@178.234.29.197`. После выполнения данной команды нам необходимо будет ввести пароль от указанного пользователя. При успешном копировании мы увидим сообщение, что ключ был добавлен и мы можем попробовать подключиться без пароля. Пример выполнения данных действий представлен на рисунке 5. После успешного подключения по ssh без пароля убедимся, что ключ успешно передался. Для этого необходимо перейти в директорию `.ssh` и посмотреть содержимое файла `authorized_keys`. При успешном копировании ключа содержимое данного файла должно быть похоже на то, что изображено на рисунке 6.



```
vitaly@ubuntu: ~/.ssh
HOST lab_server
HostName 178.234.29.197
User stud7
Port 22
IdentityFile ~/.ssh/id_rsa
```

Рисунок 7 – Создание нового псевдонима для сервера

Теперь после выполнения команды `ssh lab_server` мы успешно подключимся по ключу к удаленному серверу не вводя данные пользователя. Пример выполнения подключения через псевдоним без ввода данных представлен на рисунке 8.



```
vitaly@ubuntu: ~
vitaly@ubuntu:~$ ssh lab_server
Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 4.4.0-193-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:   https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

 * Introducing self-healing high availability clusters in MicroK8s.
   Simple, hardened, Kubernetes for production, from RaspberryPi to DC.

   https://microk8s.io/high-availability

15 packages can be updated.
0 updates are security updates.

New release '18.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

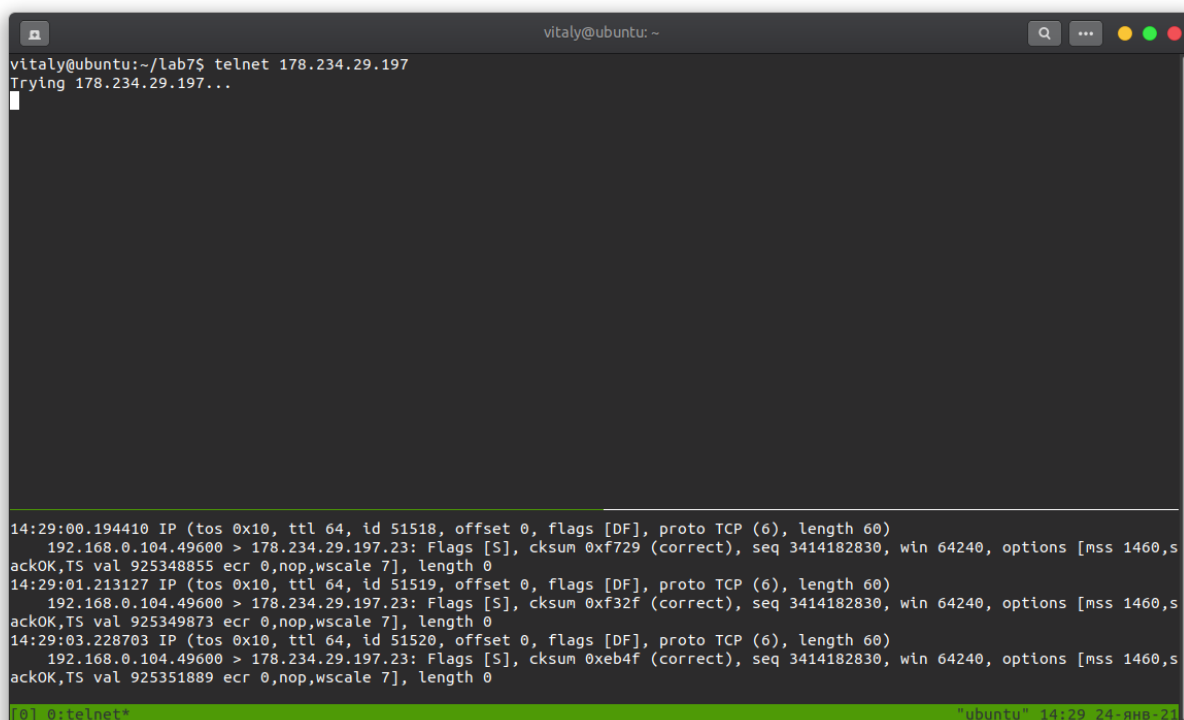
*** Требуется перезагрузка системы ***
Last login: Sun Jan 24 14:00:35 2021 from 217.107.199.150
$
```

Рисунок 8 – Успешное подключение через псевдоним без ввода данных пользователя

Следующим пунктом выполнения данной лабораторной работы является работа с терминальным мультиплексором `tmux` и анализатором трафика `tcpdump`. Создадим в одном окне два терминала: в верхнем будем пытаться подключаться к удаленному серверу и производить необходимые манипуляции, а в нижнем производить анализ трафика. Сначала запустим два терминала, для этого выполним следующие действия: запустим `tmux` и выполним комбинацию клавиш `Ctrl-B` и `%`, тем самым создадим два терминала разделенных горизонтальной линией. Для разделения терминалов вертикальной линией необходимо воспользоваться комбинацией клавиш `Ctrl-B` и `“`. В нижнем терминале выполним следующую команду: `sudo tcpdump -l -v`

`-nn tcp and src port 23 or dst port 23 / tee telnet.log`. Данной командой мы будем фильтровать все пакеты получаемые и передаваемые от узла с TCP-портом источника и приемника 23 (стандартный порт TELNET`а) и записывать отфильтрованные пакеты в файл `telnet.log`.

В верхней консоли попробуем подключиться через TELNET к удаленному хосту с помощью команды `telnet 178.234.29.197`. Пример подключения и анализ передаваемого и принимаемого трафика представлен на рисунке 9.



```
vitaly@ubuntu: ~/lab7$ telnet 178.234.29.197
Trying 178.234.29.197...

14:29:00.194410 IP (tos 0x10, ttl 64, id 51518, offset 0, flags [DF], proto TCP (6), length 60)
    192.168.0.104.49600 > 178.234.29.197.23: Flags [S], cksum 0xf729 (correct), seq 3414182830, win 64240, options [mss 1460,s
ackOK,TS val 925348855 ecr 0,nop,wscale 7], length 0
14:29:01.213127 IP (tos 0x10, ttl 64, id 51519, offset 0, flags [DF], proto TCP (6), length 60)
    192.168.0.104.49600 > 178.234.29.197.23: Flags [S], cksum 0xf32f (correct), seq 3414182830, win 64240, options [mss 1460,s
ackOK,TS val 925349873 ecr 0,nop,wscale 7], length 0
14:29:03.228703 IP (tos 0x10, ttl 64, id 51520, offset 0, flags [DF], proto TCP (6), length 60)
    192.168.0.104.49600 > 178.234.29.197.23: Flags [S], cksum 0xeb4f (correct), seq 3414182830, win 64240, options [mss 1460,s
ackOK,TS val 925351889 ecr 0,nop,wscale 7], length 0

[0] 0:telnet*
```

Рисунок 9 – Подключение и анализ передаваемого и принимаемого трафика через TELNET

Как видно из рисунка 9 подключиться к удаленному серверу через TELNET не удалось, возможно ошибка связана с неправильной настройкой серверной части `telnetd`. Остановим попытки подключения к серверу через TELNET и перехват пакетов с порта 23. Посмотрим на содержимое полученных пакетов с помощью команды `more telnet.log`. Пример выполнения данных действий представлен на рисунке 10.

```
vitaly@ubuntu:~/lab7$ telnet 178.234.29.197
Trying 178.234.29.197...
^C
vitaly@ubuntu:~/lab7$ less telnet.log
vitaly@ubuntu:~/lab7$ more telnet.log
14:29:00.194410 IP (tos 0x10, ttl 64, id 51518, offset 0, flags [DF], proto TCP (6), length 60)
192.168.0.104.49600 > 178.234.29.197.23: Flags [S], cksum 0xf729 (correct), seq 3414182830, win 64240, options [nss 1460,s
ackOK,TS val 925348855 ecr 0,nop,wscale 7], length 0
14:29:01.213127 IP (tos 0x10, ttl 64, id 51519, offset 0, flags [DF], proto TCP (6), length 60)
192.168.0.104.49600 > 178.234.29.197.23: Flags [S], cksum 0xf32f (correct), seq 3414182830, win 64240, options [nss 1460,s
ackOK,TS val 925349873 ecr 0,nop,wscale 7], length 0
14:29:03.228703 IP (tos 0x10, ttl 64, id 51520, offset 0, flags [DF], proto TCP (6), length 60)
192.168.0.104.49600 > 178.234.29.197.23: Flags [S], cksum 0xeb4f (correct), seq 3414182830, win 64240, options [nss 1460,s
ackOK,TS val 925351889 ecr 0,nop,wscale 7], length 0
14:29:07.356549 IP (tos 0x10, ttl 64, id 51521, offset 0, flags [DF], proto TCP (6), length 60)
192.168.0.104.49600 > 178.234.29.197.23: Flags [S], cksum 0xdb2f (correct), seq 3414182830, win 64240, options [nss 1460,s
ackOK,TS val 925356017 ecr 0,nop,wscale 7], length 0
14:29:15.549133 IP (tos 0x10, ttl 64, id 51522, offset 0, flags [DF], proto TCP (6), length 60)
192.168.0.104.49600 > 178.234.29.197.23: Flags [S], cksum 0xbb2f (correct), seq 3414182830, win 64240, options [nss 1460,s
ackOK,TS val 925364209 ecr 0,nop,wscale 7], length 0
vitaly@ubuntu:~/lab7$

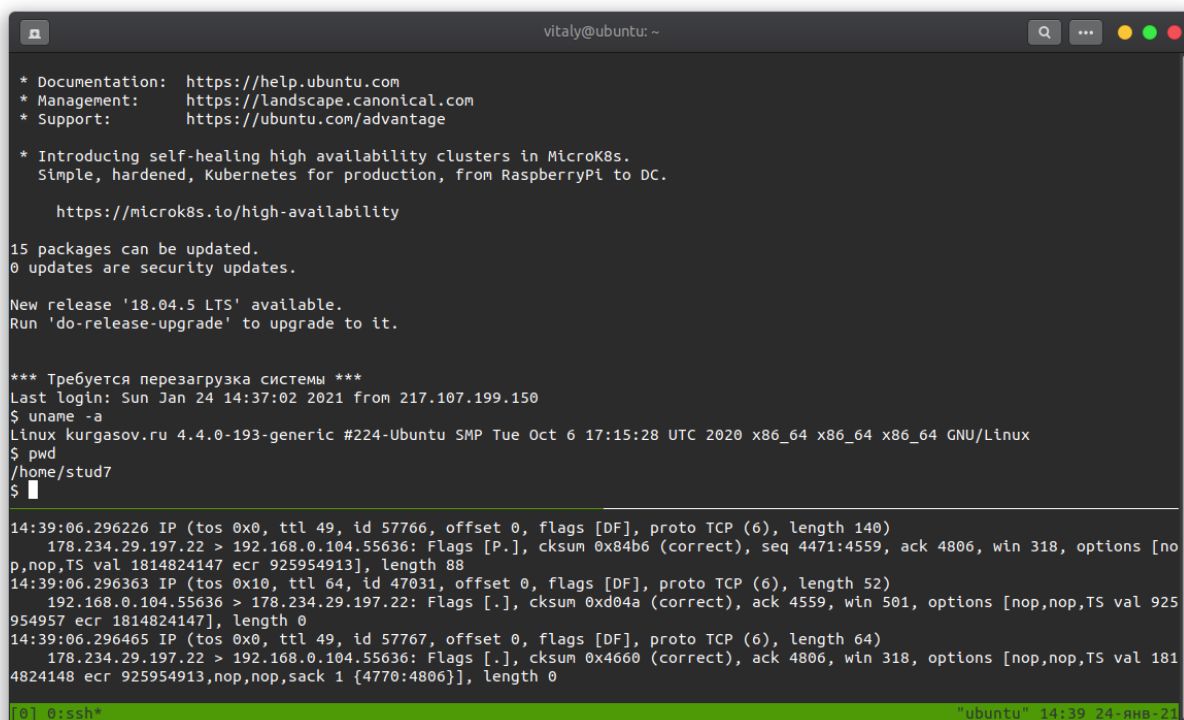
14:29:15.549133 IP (tos 0x10, ttl 64, id 51522, offset 0, flags [DF], proto TCP (6), length 60)
192.168.0.104.49600 > 178.234.29.197.23: Flags [S], cksum 0xbb2f (correct), seq 3414182830, win 64240, options [nss 1460,s
ackOK,TS val 925364209 ecr 0,nop,wscale 7], length 0
^C5 packets captured
5 packets received by filter
0 packets dropped by kernel

vitaly@ubuntu:~/lab7$ ls
telnet.log
vitaly@ubuntu:~/lab7$
[0] 0:bash*
```

Рисунок 10 – Просмотр полученных пакетов

Как видно из рисунка 10, были отправлены пакеты на удаленный сервер по порту 23 с попыткой подключения. Но ответа не было дано.

Попробуем проделать аналогичную манипуляцию только через подключение по SSH. Для этого воспользуемся следующей командой: *sudo tcpdump -l -v -nn tcp and src port 22 or dst port 22 | tee ssh.log*. Данная команда аналогична предыдущей кроме прослушиваемого порта. Для подключения по SSH по умолчанию используется 22 порт. Пример выполнения авторизации при анализе трафика представлен на рисунке 11.



```
vitaly@ubuntu: ~  
* Documentation: https://help.ubuntu.com  
* Management: https://landscape.canonical.com  
* Support: https://ubuntu.com/advantage  
  
* Introducing self-healing high availability clusters in MicroK8s.  
Simple, hardened, Kubernetes for production, from RaspberryPi to DC.  
  
https://microk8s.io/high-availability  
  
15 packages can be updated.  
0 updates are security updates.  
  
New release '18.04.5 LTS' available.  
Run 'do-release-upgrade' to upgrade to it.  
  
*** Требуется перезагрузка системы ***  
Last login: Sun Jan 24 14:37:02 2021 from 217.107.199.150  
$ uname -a  
Linux kurgasov.ru 4.4.0-193-generic #224-Ubuntu SMP Tue Oct 6 17:15:28 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux  
$ pwd  
/home/stud7  
$  
  
14:39:06.296226 IP (tos 0x0, ttl 49, id 57766, offset 0, flags [DF], proto TCP (6), length 140)  
178.234.29.197.22 > 192.168.0.104.55636: Flags [P.], cksum 0x84b6 (correct), seq 4471:4559, ack 4806, win 318, options [nop,nop,TS val 1814824147 ecr 925954913], length 88  
14:39:06.296363 IP (tos 0x10, ttl 64, id 47031, offset 0, flags [DF], proto TCP (6), length 52)  
192.168.0.104.55636 > 178.234.29.197.22: Flags [.] , cksum 0xd04a (correct), ack 4559, win 501, options [nop,nop,TS val 925954957 ecr 1814824147], length 0  
14:39:06.296465 IP (tos 0x0, ttl 49, id 57767, offset 0, flags [DF], proto TCP (6), length 64)  
178.234.29.197.22 > 192.168.0.104.55636: Flags [S.], cksum 0x4660 (correct), ack 4806, win 318, options [nop,nop,TS val 1814824148 ecr 925954913,nop,nop,sack 1 {4770:4806}], length 0  
  
[0] 0:ssh* "ubuntu" 14:39 24-RH8-21
```

Рисунок 11 – Успешная авторизации на сервере через SSH при анализе трафика

Как видно из рисунка 11, подключение на сервере произошло успешно. Также это можно наблюдать в файле `ssh.log`, а именно в следующих строчках:

```
14:38:57.421991 IP (tos 0x0, ttl 64, id 46983, offset 0, flags [DF], proto TCP (6), length 60)  
192.168.0.104.55636 > 178.234.29.197.22: Flags [S], cksum 0x422b (correct), seq 2065177742, win 64240, options [mss 1460,sackOK,TS val 925946082 ecr 0,nop,wscale 7], length 0  
14:38:57.461086 IP (tos 0x0, ttl 49, id 0, offset 0, flags [DF], proto TCP (6), length 60)  
178.234.29.197.22 > 192.168.0.104.55636: Flags [S.], cksum 0x8244 (correct), seq 790409789, ack 2065177743, win 28960, options [mss 1440,sackOK,TS val 1814821940 ecr 925946082,nop,wscale 7], length 0  
14:38:57.461293 IP (tos 0x0, ttl 64, id 46984, offset 0, flags [DF], proto TCP (6), length 52)
```

В сравнении с подключением через TELNET мы видим, что ответ от сервера был получен в кратчайшие сроки первым же пакетом.

Теперь попробуем передать файл на сервер по SSH. Для этого сначала создадим файл `info.txt` с содержимым: «Lab7, Posadnev Vitaly». Для передачи

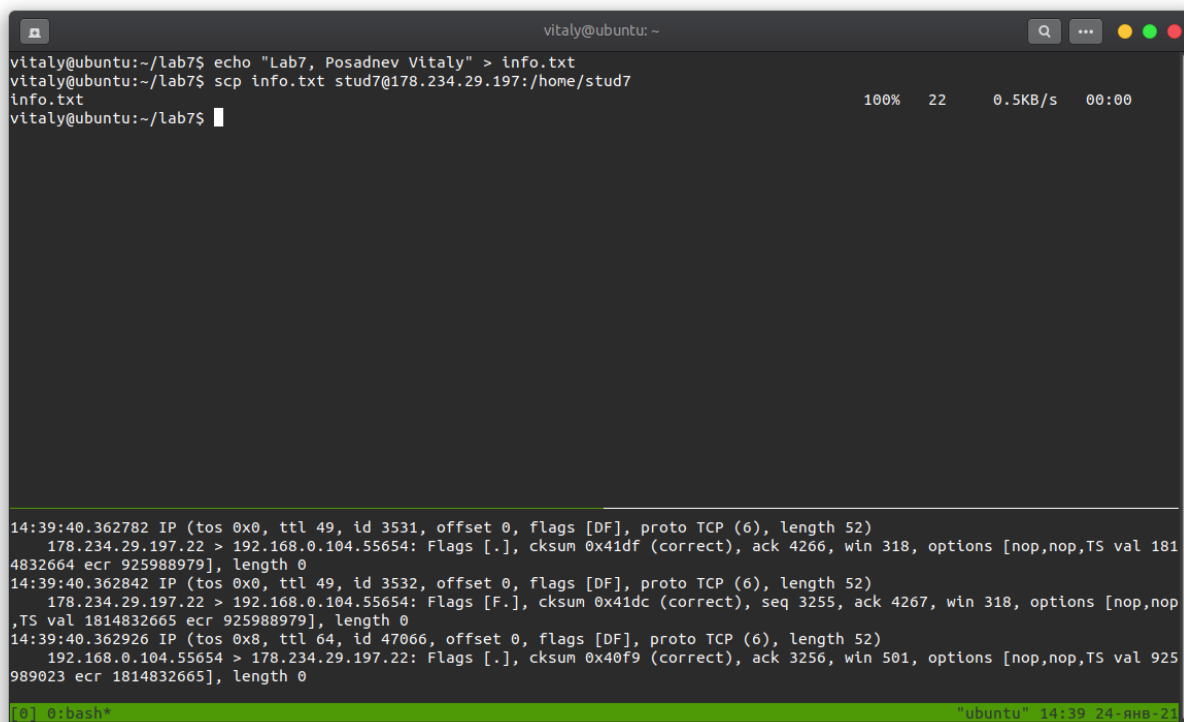
файла по SSH необходимо выполнить следующую команду: `scp info.txt stud7@178.234.29.197:/home/stud7`. Рассмотрим данную команду подробнее:

- `info.txt` – название файла который необходимо отправить по SSH. Так как данный файл находится в данном каталоге, то необходимо прописать только его название, в ином случае – полный путь до файла.

- `stud7@178.234.29.197` – имя пользователя и сервер на который будет осуществляться передача файла.

- `/home/stud7` – путь по которому будет помещен передаваемый файл. Также можно указать вместе с новым названием файла.

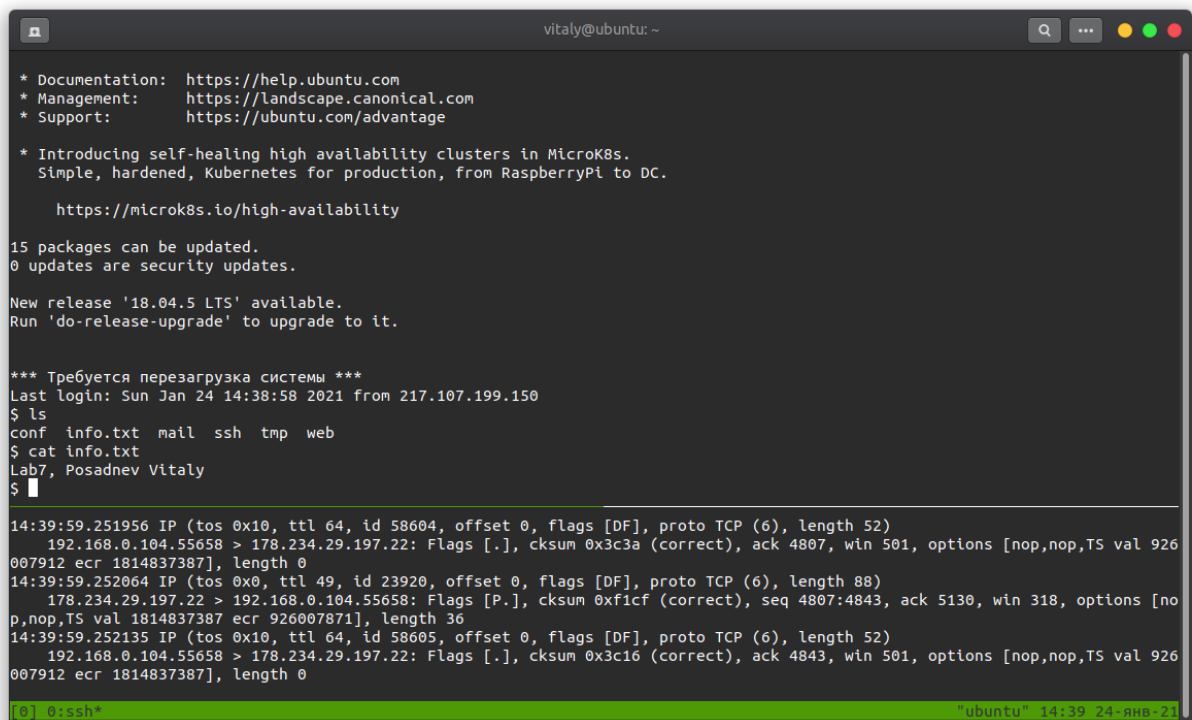
Пример успешной передачи файла на сервер по SSH представлен на рисунке 12.



```
vitaly@ubuntu: ~  
vitaly@ubuntu:~/lab7$ echo "Lab7, Posadnev Vitaly" > info.txt  
vitaly@ubuntu:~/lab7$ scp info.txt stud7@178.234.29.197:/home/stud7  
info.txt 100% 22 0.5KB/s 00:00  
vitaly@ubuntu:~/lab7$  
  
14:39:40.362782 IP (tos 0x0, ttl 49, id 3531, offset 0, flags [DF], proto TCP (6), length 52)  
178.234.29.197.22 > 192.168.0.104.55654: Flags [.], cksum 0x41df (correct), ack 4266, win 318, options [nop,nop,TS val 1814832664 ecr 925988979], length 0  
14:39:40.362842 IP (tos 0x0, ttl 49, id 3532, offset 0, flags [DF], proto TCP (6), length 52)  
178.234.29.197.22 > 192.168.0.104.55654: Flags [F.], cksum 0x41dc (correct), seq 3255, ack 4267, win 318, options [nop,nop,TS val 1814832665 ecr 925988979], length 0  
14:39:40.362926 IP (tos 0x8, ttl 64, id 47066, offset 0, flags [DF], proto TCP (6), length 52)  
192.168.0.104.55654 > 178.234.29.197.22: Flags [.], cksum 0x40f9 (correct), ack 3256, win 501, options [nop,nop,TS val 925989023 ecr 1814832665], length 0  
[0] 0:bash* "ubuntu" 14:39 24-янв-21
```

Рисунок 12 – Успешная отправка файла на сервер

Теперь авторизуемся на сервере по SSH с помощью команды `ssh lab_server` и посмотрим на содержимое папки `/home/stud7`. В данном каталоге будет находиться ранее переданный файл, содержимое которого можно наблюдать на рисунке 13.



```
vitaly@ubuntu: ~  
* Documentation: https://help.ubuntu.com  
* Management: https://landscape.canonical.com  
* Support: https://ubuntu.com/advantage  
  
* Introducing self-healing high availability clusters in MicroK8s.  
Simple, hardened, Kubernetes for production, from RaspberryPi to DC.  
  
https://microk8s.io/high-availability  
  
15 packages can be updated.  
0 updates are security updates.  
  
New release '18.04.5 LTS' available.  
Run 'do-release-upgrade' to upgrade to it.  
  
*** Требуется перезагрузка системы ***  
Last login: Sun Jan 24 14:38:58 2021 from 217.107.199.150  
$ ls  
conf info.txt mail ssh tmp web  
$ cat info.txt  
Lab7, Posadnev Vitaly  
$  
  
14:39:59.251956 IP (tos 0x10, ttl 64, id 58604, offset 0, flags [DF], proto TCP (6), length 52)  
192.168.0.104.55658 > 178.234.29.197.22: Flags [.], cksum 0x3c3a (correct), ack 4807, win 501, options [nop,nop,TS val 926  
007912 ecr 1814837387], length 0  
14:39:59.252064 IP (tos 0x0, ttl 49, id 23920, offset 0, flags [DF], proto TCP (6), length 88)  
178.234.29.197.22 > 192.168.0.104.55658: Flags [P.], cksum 0xf1cf (correct), seq 4807:4843, ack 5130, win 318, options [no  
p,nop,TS val 1814837387 ecr 926007871], length 36  
14:39:59.252135 IP (tos 0x10, ttl 64, id 58605, offset 0, flags [DF], proto TCP (6), length 52)  
192.168.0.104.55658 > 178.234.29.197.22: Flags [.], cksum 0x3c16 (correct), ack 4843, win 501, options [nop,nop,TS val 926  
007912 ecr 1814837387], length 0  
  
[0] 0:ssh* "ubuntu" 14:39 24-января 2021
```

Рисунок 13 – Успешное получение файла по SSH

Последним шагом нам необходимо удостовериться, что ранее запущенная сессия `screen` всё еще активна и выполняется. Для просмотра списка активных сессий `screen` необходимо выполнить команду `screen -li`. Для подключения к выбранной сессии `screen` необходимо выполнить команду `screen -r id_сессии` или ранее указанное имя. В нашем случае, необходимо воспользоваться командой `screen -r 19302`. Пример просмотра активных сессий `screen` представлен на рисунке 14. После успешного подключения к сессии посмотрим мы увидим, что последний отправленный пакет находится под номером 2752. Это означает, что данная сессия выполнялась всё время пока делали предыдущие пункты лабораторной работы. Изображение состояния сессии `screen` представлено на рисунке 15.

```
vitaly@ubuntu: ~  
  
* Introducing self-healing high availability clusters in MicroK8s.  
Simple, hardened, Kubernetes for production, from RaspberryPi to DC.  
  
https://microk8s.io/high-availability  
  
15 packages can be updated.  
0 updates are security updates.  
  
New release '18.04.5 LTS' available.  
Run 'do-release-upgrade' to upgrade to it.  
  
*** Требуется перезагрузка системы ***  
Last login: Sun Jan 24 14:38:58 2021 from 217.107.199.150  
$ ls  
conf info.txt mail ssh tmp web  
$ cat info.txt  
Lab7, Posadnev Vitaly  
$ screen -tl  
There is a screen on:  
19302.ping_ya_ru (24.01.2021 13:54:13) (Detached)  
1 Socket in /var/run/screen/S-stud7.  
$  
  
14:40:09.459544 IP (tos 0x10, ttl 64, id 58628, offset 0, flags [DF], proto TCP (6), length 52)  
192.168.0.104.55658 > 178.234.29.197.22: Flags [.], cksum 0x068b (correct), ack 5395, win 500, options [nop,nop,TS val 926  
018120 ecr 1814839939], length 0  
14:40:09.459644 IP (tos 0x0, ttl 49, id 23935, offset 0, flags [DF], proto TCP (6), length 88)  
178.234.29.197.22 > 192.168.0.104.55658: Flags [P.], cksum 0xa6ee (correct), seq 5395:5431, ack 5526, win 318, options [no  
p,nop,TS val 1814839939 ecr 926018073], length 36  
14:40:09.459699 IP (tos 0x10, ttl 64, id 58629, offset 0, flags [DF], proto TCP (6), length 52)  
192.168.0.104.55658 > 178.234.29.197.22: Flags [.], cksum 0x0667 (correct), ack 5431, win 500, options [nop,nop,TS val 926  
018120 ecr 1814839939], length 0  
  
[0] 0:ssh* "ubuntu" 14:40 24-Янв-21
```

Рисунок 14 – Просмотр активных сессий screen

```
vitaly@ubuntu: ~  
  
64 bytes from ya.ru (87.250.250.242): icmp_seq=2730 ttl=249 time=24.1 ms  
64 bytes from ya.ru (87.250.250.242): icmp_seq=2731 ttl=249 time=24.1 ms  
64 bytes from ya.ru (87.250.250.242): icmp_seq=2732 ttl=249 time=24.3 ms  
64 bytes from ya.ru (87.250.250.242): icmp_seq=2733 ttl=249 time=24.1 ms  
64 bytes from ya.ru (87.250.250.242): icmp_seq=2734 ttl=249 time=24.2 ms  
64 bytes from ya.ru (87.250.250.242): icmp_seq=2735 ttl=249 time=24.2 ms  
64 bytes from ya.ru (87.250.250.242): icmp_seq=2736 ttl=249 time=24.3 ms  
64 bytes from ya.ru (87.250.250.242): icmp_seq=2737 ttl=249 time=24.2 ms  
64 bytes from ya.ru (87.250.250.242): icmp_seq=2738 ttl=249 time=24.4 ms  
64 bytes from ya.ru (87.250.250.242): icmp_seq=2739 ttl=249 time=24.1 ms  
64 bytes from ya.ru (87.250.250.242): icmp_seq=2740 ttl=249 time=24.2 ms  
64 bytes from ya.ru (87.250.250.242): icmp_seq=2741 ttl=249 time=24.2 ms  
64 bytes from ya.ru (87.250.250.242): icmp_seq=2742 ttl=249 time=24.2 ms  
64 bytes from ya.ru (87.250.250.242): icmp_seq=2743 ttl=249 time=25.7 ms  
64 bytes from ya.ru (87.250.250.242): icmp_seq=2744 ttl=249 time=24.2 ms  
64 bytes from ya.ru (87.250.250.242): icmp_seq=2745 ttl=249 time=24.3 ms  
64 bytes from ya.ru (87.250.250.242): icmp_seq=2746 ttl=249 time=24.2 ms  
64 bytes from ya.ru (87.250.250.242): icmp_seq=2747 ttl=249 time=24.3 ms  
64 bytes from ya.ru (87.250.250.242): icmp_seq=2748 ttl=249 time=24.2 ms  
64 bytes from ya.ru (87.250.250.242): icmp_seq=2749 ttl=249 time=24.3 ms  
64 bytes from ya.ru (87.250.250.242): icmp_seq=2750 ttl=249 time=24.3 ms  
64 bytes from ya.ru (87.250.250.242): icmp_seq=2751 ttl=249 time=24.4 ms  
64 bytes from ya.ru (87.250.250.242): icmp_seq=2752 ttl=249 time=24.2 ms  
  
14:40:21.036181 IP (tos 0x10, ttl 64, id 58667, offset 0, flags [DF], proto TCP (6), length 52)  
192.168.0.104.55658 > 178.234.29.197.22: Flags [.], cksum 0xc16d (correct), ack 8047, win 495, options [nop,nop,TS val 926  
029696 ecr 1814842833], length 0  
14:40:22.037497 IP (tos 0x0, ttl 49, id 23959, offset 0, flags [DF], proto TCP (6), length 160)  
178.234.29.197.22 > 192.168.0.104.55658: Flags [P.], cksum 0x5ff8 (correct), seq 8047:8155, ack 6102, win 318, options [no  
p,nop,TS val 1814843084 ecr 926029696], length 108  
14:40:22.037633 IP (tos 0x10, ttl 64, id 58668, offset 0, flags [DF], proto TCP (6), length 52)  
192.168.0.104.55658 > 178.234.29.197.22: Flags [.], cksum 0xb31c (correct), ack 8155, win 495, options [nop,nop,TS val 926  
030698 ecr 1814843084], length 0  
  
[0] 0:ssh* "ubuntu" 14:40 24-Янв-21
```

Рисунок 15 – Просмотр состояния сессии screen

Вывод

При выполнении данной лабораторной работы были приобретены практические навыки работы с программным обеспечением удаленного доступа к распределенным системам обработки данных.

Контрольные вопросы

1. Определите основные цели и задачи решаемые с помощью ПО удаленного доступа?

Удаленный доступ – это система, в которой пользователь может удаленно подключиться и управлять определенным компьютером так, как если бы он находился прямо перед ним.

2. Выделите отличительные особенности между режимами работы удаленного доступа по протоколам TELNET и SSH?

TELNET очень часто использовался раньше, для удаленного управления компьютером с Linux, но потом ему на замену пришел защищенный протокол SSH. Но telnet все еще используется, например, для тестирования сети, проверки портов, а также для взаимодействия с различными IoT устройствами и роутерами. Поскольку ключ SSH представляет собой зашифрованное значение, которое должно быть расшифровано, то из-за этого можно сделать вывод, что он медленнее чем TELNET. Однако, является более безопасным.

3. Как сгенерировать ключи SSH в разных ОС?

В Unix подобных системах для генерации ключа достаточно использовать утилиту ssh-keygen. Она позволяет генерировать ключи разными алгоритмами и с дополнительными параметрами.

В современных Windows имеются стандартные средства генерации ssh ключей. Некоторая часть команд Linux доступна в Windows, в данном списке имеется ssh-keygen. Но также можно использовать программу Putty, но в этом случае Linux-сервер не примет ключ, сгенерированный данной программой и придется вручную доделывать SSH подключение.

4. Основываясь на заданиях лабораторной работы, приведите практический пример использования систем удаленного доступа?

Допустим мы решили создать веб-сайт, который был бы доступен по всему интернету. Мы можем купить удаленную систему, которая может находиться в другой стране. Для её настройки и подключения мы можем использовать систему удаленного доступа.

5. Перечислите распространенные сетевые службы, основанные на использовании шифрованного соединения по протоколу SSH? Приведите пример использования службы передачи файлов по безопасному туннелю?

GitHub, GitLab. На данных сервисах SSH используется для аутентификации на серверах без использования имени пользователя и пароля каждый раз.

6. Возможно ли из «секретного» ключа сгенерировать «публичный» и/или наоборот?

Возможно, но на это уйдет слишком много времени. Так как ключ шифруется с помощью алгоритма SHA256, то рассчитаем сколько времени на это потребуется. 36 символов в наборе, 64 количество символов всего. Если взять за условность, что в секунду мы будем перебирать 1 миллион комбинаций, что около 31.536.000 секунд в год. То мы получим достаточно большую цифру вычисляемую по формуле: $\left(\frac{36^{64}}{1000000}\right) / 31536000 = 1.27 * 10^{86}$ лет.

7. Будут ли отличаться пары ключей, сгенерированные на одном ПК несколько раз с исходными условиями (наличие/отсутствие пароля на «секретный» ключ и т.п.)?

Да, будут отличаться. Это обусловлено использованием ГСЧ.

8. Перечислите доступные ключи для ssh-keygen.exe
RSA, DSA, ECDSA, ED25519.

9. Можно ли использовать один «секретный» ключ доступа с разных ОС, установленных на одном ПК/на разных ПК?

Можно, но не является безопасным.

10. Возможно ли организовать подключение «по ключу» SSH к системе с ОС Windows, в которой запущен OpenSSH сервер?

Да, возможно. Для этого можно использовать программу Putty.