

Липецкий государственный технический университет

Кафедра прикладной математики

Отчет по лабораторной работе №7 «Работа с SSH. Авторизация по ключу SSH.»

Студент

подпись, дата

Комолых Т.О.
фамилия, инициалы

Группа

ПМ-18

Руководитель

доц., к.п.н. кафедры АСУ
ученая степень, ученое звание

подпись, дата

Кургасов В. В.
фамилия, инициалы

Липецк 2021 г.

Содержание

Цель работы	3
Практическое задание	3
Выполнение практического задания.	5
Вопросы для самопроверки.	12
Вывод	15
Список литературы	16

Цель работы

Лабораторная работа предназначена для целей практического ознакомления с программным обеспечением удалённого доступа к распределённым системам обработки данных.

Практическое задание

- 1) Создать подключение удаленного доступа к системе обработки данных, сформировать шифрованные ключи и произвести их обмен с удаленной системой, передать файл по шифрованному туннелю, воспользовавшись беспарольным доступом с аутентфикацией по публичным ключам.
- 2) Выполнить подключение с использованием полноэкранного консольного оконного менеджера screen.
- 3) Запустить терминал с командной оболочкой ОС и ввести команду `tmux` (терминальный мультиплексор). Комбинациями клавиш `Ctrl-b` с создать новое 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. Для этого следует воспользоваться командой `sudo tcpdump -l -v -nn tcp and src port 23 or dst port 23 | tee telnet.log`;
- 4) В первом окне терминального мультиплексора попытаться установить соединение с удаленным сервером `domen.name` по протоколу TELNET. Для авторизации следует использовать логин `student`; /при возможности организовать такой доступ инженерами кафедры АСУ ЛГТУ/
- 5) Воспользовавшись окном сетевого монитора, анализировать прохождение сетевых пакетов между узлами назначения. Отметить пакеты инициации соединения `telnet`;
- 6) Подключившись к удаленной системе ввести пароль `Password` и выполнить команду `uname -a`, выведя тем самым информацию об удаленной системе. Для разрыва соединения использовать команду `logout`;
- 7) В окне сетевого монитора отметить пакеты иницирующие разрыв сессии `telnet`. 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. Для этого следует воспользоваться командой `sudo tcpdump -l -v -nn tcp and src port 22 or dst port 22 | tee ssh.log`;
- 9) Переключившись на первое окно терминального мультиплексора, с помощью команды `ssh -l student domen.name` попытаться установить шифрованное соединени с удаленным сервером `domen.name`. Проследить передачу и прием пакетов между узлами в окне сетевого анализатора. Отметить взаимодействующие TCP-порты;
- 10) Подключившись к удаленной системе ввести пароль `Password` 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 -v -o User=student/home/student/имя_файла domen.name:/home/student/`

- передать его по зашифрованному каналу на удаленную систему. Проверить наличие копии переданного файла на удаленном узле, воспользовавшись файловым менеджером «Midnight Commander» (команда `mc` на удаленной системе);
- 12) Отключившись от удаленного узла (команда `exit`), на локальном хосте, сформировать зашифрованные ключи, воспользовавшись командой `ssh-keygen`;
 - 13) Используя команду `scp` с указанием места расположения файла (публичного ключа) на локальной системе (`/home/student/.ssh/key.pub`), произвести его передачу по зашифрованному туннелю на удаленный узел в заданный каталог `/home/student/.ssh/` под именем `authorized_keys`. Проследить процесс пересылки пакетов между удаленными узлами в окне анализатора пакетов;
 - 14) Воспользовавшись командой `ssh -l student domen.name`, снова сделать попытку подключения к удаленной системе. Отметить отличия в процедурах подключения и регистрации пользователя на удаленной системе;
 - 15) Аналогично, с помощью команды `scp`, произвести повторную передачу текстового файла на удаленный узел. Убедиться в наличии переданной копии файла на удаленном хосте. Отметить отличия в процедуре передачи файла;
 - 16) Остановить анализатор сетевых пакетов, воспользовавшись комбинацией `Ctrl-c`. Просмотреть содержимое файла `ssh.log`, отметить пакеты инициации сетевого взаимодействия и разрыва соединений TCP.

Выполнение практического задания.

1. SSH использует асимметричное шифрование, суть которого заключается в наличии двух ключей: закрытого и открытого. Для генерации пары ключей используется программа `ssh-keygen` (рисунок 1).

```
root@arianrod:~# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:0kgES0Dmgbs7h8oTIjy8puh0+k6pMZGIafygZpLMHHg root@arianrod
The key's randomart image is:
+----[RSA 3072]-----+
|+o.
|+ .
|. + .
|Boo.
|OOE . S
|OO=... .
|B@+=. o
|X=0 .
|XB+o
+----[SHA256]-----+
```

Рисунок 1.

В результате будет создано два файла: `id_rsa` и `id_rsa.pub`. Открытый ключ хранится в файле `/root/.ssh/id_rsa.pub`, закрытый — `/root/.ssh/id_rsa` (рисунок 2).

```
root@arianrod:~# ls /root/.ssh
authorized_keys id_rsa id_rsa.pub known_hosts
```

Рисунок 2.

Скопируем содержимое файла `id_rsa.pub` на удалённую машину в файл `/.ssh/authorized_keys` (рисунок 3).

```
root@arianrod:~# ssh-copy-id stud3@178.234.29.197
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install all the new keys
stud3@178.234.29.197's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'stud3@178.234.29.197'"
and check to make sure that only the key(s) you wanted were added.
```

Рисунок 3.

Теперь выполним подключение с помощью клиента SSH (рисунок 4).

```

root@arianrod:~# ssh 'stud3@178.234.29.197'
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: Mon Jan 11 17:26:25 2021 from 178.234.129.89
$

```

Рисунок 4.

Передадим файл file по зашифрованному туннелю, воспользовавшись беспарольным доступом с аутентификацией по публичному ключу (рисунок 5).

```

root@arianrod:~# scp ./file stud3@178.234.29.197:
file                                                                    100% 10 2.8KB/s 00:00

```

Рисунок 5.

2. Теперь, подключившись к удалённому серверу, используем полноэкранный консольный оконный менеджер screen (рисунок 6).

```

Last login: Thu Jan 28 17:16:38 2021 from 100.112.201.247
$ screen_

```

Рисунок 6.

3. Создадим новое окно и запустим анализатор трафика tcpdump, предварительно запустив терминальный мультиплексор tmux, с фильтром пакетов, получаемых и передаваемых от узла domain.name с TCP-портом источника и назначения 23. С помощью команды tee, выводим отфильтрованные IP-пакеты на терминал и сохраняем данные в файл telnet.log, в домашнем каталоге пользователя. Для этого следует воспользоваться командой `sudo tcpdump -l -v -nn tcp and src port 23 or dst port 23 | tee telnet.log`;

4-7. telnet – утилита, предназначенная для создания интерактивного соединения между удаленными компьютерами. Она работает по протоколу TELNET, но этот протокол поддерживается многими сервисами, поэтому её можно использовать для управления ими. Протокол работает на основе TCP, и позволяет передавать обычные строковые команды на другое устройство.

Попытка установить связь с удалённым сервером по протоколу TELNET оказалась неудачной, так как отсутствует связь с сервером по 23 порту. Об этом говорит ошибка, полученная в результате долгого ожидания ответа сервера (рисунок 7).

```

tatyana@arianrod:~$ sudo telnet 178.234.29.197 23
[sudo] password for tatyana:
Trying 178.234.29.197...
telnet: Unable to connect to remote host: Connection timed out

```

Рисунок 7.

8. Снова запустим анализатор сетевого трафика с фильтром пакетов, получаемых и передаваемых узлу `domen.name` с TCP-портом источника и назначения 22. Выводим отфильтрованные IP-пакеты на терминал и сохраняем данные в файл `ssh.log`. Для этого следует воспользоваться командой `sudo tcpdump -l -v -nn tcp and src port 22 or dst port 22 | tee ssh.log`;

9. Переключившись на первое окно с запущенным терминальным мультиплексором, попытаемся установить зашифрованное соединение с удалённым сервером (рисунок 8).

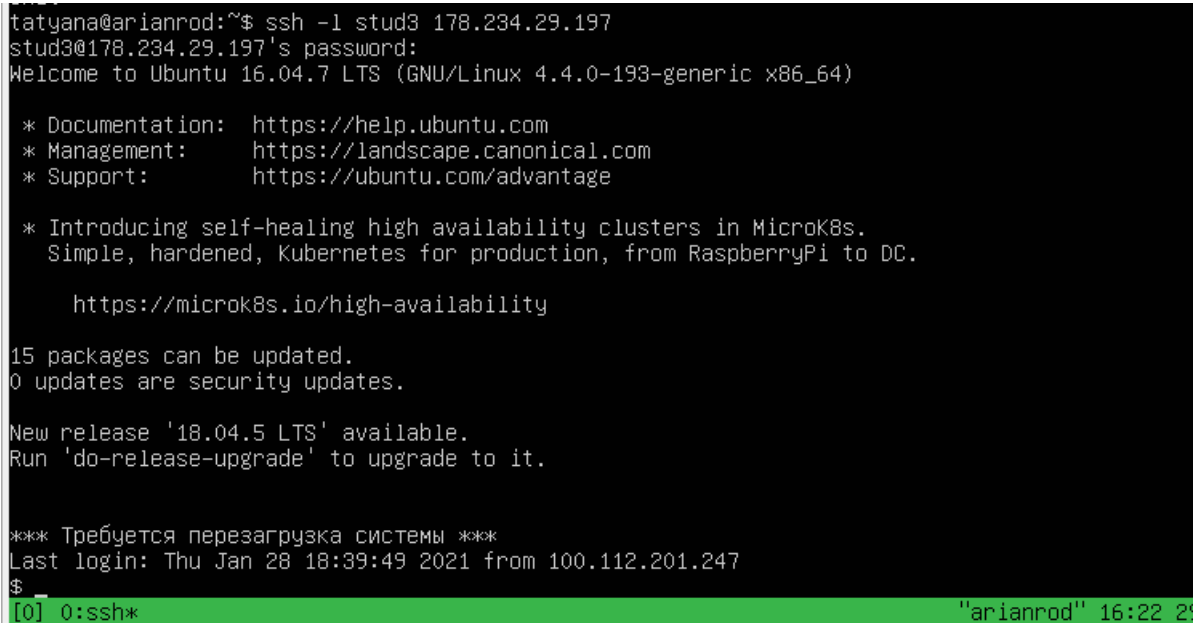
A screenshot of a terminal window with a black background and white text. The terminal shows an SSH session initiated from a host named 'tatyana@arianrod' to a remote host 'stud3@178.234.29.197'. The user 'stud3' provides a password, and the remote system is identified as 'Ubuntu 16.04.7 LTS (GNU/Linux 4.4.0-193-generic x86_64)'. The terminal displays various system messages, including links to documentation, management, and support, as well as information about available updates. At the bottom, a green status bar shows '[0] 0:ssh*' and the host name 'arianrod'.

Рисунок 8.

Вернувшись к окну с запущенным анализатором сетевого трафика, заметим, что в процессе соединения с удалённым сервером была осуществлена передача 71-ого IP-пакета между сервером по 22-ому порту и клиентом по 39468-ому (рисунок 9).

```

10.0.2.15.39468 > 178.234.29.197.22: Flags [.] , cksum 0xdcd8 (incorrect -> 0x9d11), ack 2447, win
n 63440, length 0
16:40:25.154418 IP (tos 0x0, ttl 64, id 57418, offset 0, flags [none], proto TCP (6), length 84)
178.234.29.197.22 > 10.0.2.15.39468: Flags [P.] , cksum 0x86e6 (correct), seq 2447:2491, ack 1990
, win 65535, length 44
16:40:25.154566 IP (tos 0x0, ttl 64, id 53179, offset 0, flags [DF], proto TCP (6), length 40)
10.0.2.15.39468 > 178.234.29.197.22: Flags [.] , cksum 0xdcd8 (incorrect -> 0x9ce5), ack 2491, wi
n 63440, length 0
16:40:25.154863 IP (tos 0x10, ttl 64, id 53180, offset 0, flags [DF], proto TCP (6), length 484)
10.0.2.15.39468 > 178.234.29.197.22: Flags [P.] , cksum 0xde94 (incorrect -> 0x81e7), seq 1990:24
34, ack 2491, win 63440, length 444
16:40:25.155062 IP (tos 0x0, ttl 64, id 57419, offset 0, flags [none], proto TCP (6), length 40)
178.234.29.197.22 > 10.0.2.15.39468: Flags [.] , cksum 0x92fa (correct), ack 2434, win 65535, len
gth 0
16:40:25.161238 IP (tos 0x0, ttl 64, id 57420, offset 0, flags [none], proto TCP (6), length 696)
178.234.29.197.22 > 10.0.2.15.39468: Flags [P.] , cksum 0xd0f3 (correct), seq 2491:3147, ack 2434
, win 65535, length 656
16:40:25.161238 IP (tos 0x0, ttl 64, id 57421, offset 0, flags [none], proto TCP (6), length 252)
178.234.29.197.22 > 10.0.2.15.39468: Flags [P.] , cksum 0xde75 (correct), seq 3147:3359, ack 2434
, win 65535, length 212
16:40:25.161259 IP (tos 0x10, ttl 64, id 53181, offset 0, flags [DF], proto TCP (6), length 40)
10.0.2.15.39468 > 178.234.29.197.22: Flags [.] , cksum 0xdcd8 (incorrect -> 0x9899), ack 3147, wi
n 63440, length 0
16:40:25.161287 IP (tos 0x10, ttl 64, id 53182, offset 0, flags [DF], proto TCP (6), length 40)
10.0.2.15.39468 > 178.234.29.197.22: Flags [.] , cksum 0xdcd8 (incorrect -> 0x97c5), ack 3359, wi
n 63440, length 0
16:40:25.164212 IP (tos 0x0, ttl 64, id 57422, offset 0, flags [none], proto TCP (6), length 76)
178.234.29.197.22 > 10.0.2.15.39468: Flags [P.] , cksum 0xc47c (correct), seq 3359:3395, ack 2434
, win 65535, length 36
16:40:25.164224 IP (tos 0x10, ttl 64, id 53183, offset 0, flags [DF], proto TCP (6), length 40)
10.0.2.15.39468 > 178.234.29.197.22: Flags [.] , cksum 0xdcd8 (incorrect -> 0x97a1), ack 3395, wi
n 63440, length 0
^C71 packets captured
71 packets received by filter
0 packets dropped by kernel

```

Рисунок 9.

10. После подключения к удаленной системе и ввода пароля выводим полную информацию об удалённой системе (рисунок 10).

```

$ 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

```

Рисунок 10.

11. Теперь создадим на локальном узле текстовый файл stud_LR7.txt, содержащий ФИО и номер лабораторной работы, и с помощью команды `scp -v -o User=stud3 ./stud_LR7.txt 178.234.29.197:` передадим его по зашифрованному каналу на удалённую систему (рисунок 11).

```

Sink: C0644 90 stud_LR7.txt
stud_LR7.txt                                100%  90   13.0KB/s   00:00
debug1: client_input_channel_req: channel 0 rtype exit-status reply 0
debug1: channel 0: free: client-session, nchannels 1
debug1: fd 0 clearing O_NONBLOCK
Transferred: sent 2200, received 2580 bytes, in 0.3 seconds
Bytes per second: sent 6941.1, received 8140.1
debug1: Exit status 0

```

Рисунок 11.

Проверим наличие копии переданного файла на удаленном узле, воспользовавшись файловым менеджером "Midnight Commander"(рисунок 12, 13)

Левая панель				Правая панель			
Файл	Команда	Настройки	Правки	Файл	Команда	Настройки	Правки
.и	Имя	Размер	Время	.и	Имя	Размер	Время
/..	-ВВЕРХ-	январь 8 08:53		/..	-ВВЕРХ-	январь 8 08:53	
/.cache	4096	январь 29 20:03		/.cache	4096	январь 29 20:03	
/.config	4096	январь 29 20:03		/.config	4096	январь 29 20:03	
/.local	4096	январь 29 20:03		/.local	4096	январь 29 20:03	
/.ssh	4096	декабрь 13 2019		/.ssh	4096	декабрь 13 2019	
/conf	4096	декабрь 2 2019		/conf	4096	декабрь 2 2019	
/mail	4096	декабрь 2 2019		/mail	4096	декабрь 2 2019	
/tmp	4096	декабрь 2 2019		/tmp	4096	декабрь 2 2019	
/web	4096	декабрь 2 2019		/web	4096	декабрь 2 2019	
.bash_logout	220	сентябрь 1 2015		.bash_logout	220	сентябрь 1 2015	
.bashrc	3771	сентябрь 1 2015		.bashrc	3771	сентябрь 1 2015	
.profile	655	июнь 24 2016		.profile	655	июнь 24 2016	
file	10	январь 28 17:44		file	10	январь 28 17:44	
stud_LR7.txt	90	январь 29 20:02		stud_LR7.txt	90	январь 29 20:02	

Рисунок 12.

/home/stud3/stud_LR7.txt	90/90	100%
Комольих Татьяна Олеговна		
Лабораторная работа №7		

Рисунок 13.

12. Отключившись от удалённого узла, вновь сгенерируем зашифрованные ssh-ключи key_rsa и поместим их в /home/tatyana/.ssh (рисунок 14).

```
tatyana@arianrod:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/tatyana/.ssh/id_rsa): /home/tatyana/.ssh/key_rsa
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/tatyana/.ssh/key_rsa
Your public key has been saved in /home/tatyana/.ssh/key_rsa.pub
The key fingerprint is:
SHA256:2btC3+aWtpcYufeq5HxCz9fsqrjGosgGCwPm8bWcw80 tatyana@arianrod
The key's randomart image is:
+---[RSA 3072]-----+
|
|.o . . o
|+ o + = S . .
|o... * E. .+
| o o .. o.o.B o.
| .... o +*X B +
| .o .. ++B0o*o
+---[SHA256]-----+
tatyana@arianrod:~$
```

Рисунок 14.

13. Используя команду scp с указанием места расположения файла key_rsa.pub на локальной системе ./ssh/key_rsa.pub, произведём его передачу по зашифрованному туннелю на удалённый узел в заданный каталог /home/stud3/.ssh/authorized_keys (рисунок 15).

```

tatyana@arianrod:~$ scp ./ssh/key_rsa.pub stud3@178.234.29.197:/home/stud3/.ssh/authorized_keys
stud3@178.234.29.197's password:
key_rsa.pub                                100% 570    79.0KB/s  00:00

```

Рисунок 15.

В окне с запущенным анализатором трафика заметим, что в процессе передачи файла key_rsa.pub с публичным ключём происходили пересылки IP-пакетов между удалёнными узлами (рисунок 16).

```

10.0.2.15.39480 > 178.234.29.197.22: Flags [P.], cksum 0xdcfc (incorrect -> 0xf04c), seq 3306:3342, ack 2723, win 63440, length 36
17:41:29.376231 IP (tos 0x0, ttl 64, id 58251, offset 0, flags [none], proto TCP (6), length 40)
178.234.29.197.22 > 10.0.2.15.39480: Flags [P.], cksum 0x3fec (correct), ack 3342, win 65535, length 0
17:41:29.379612 IP (tos 0x0, ttl 64, id 58252, offset 0, flags [none], proto TCP (6), length 164)
178.234.29.197.22 > 10.0.2.15.39480: Flags [P.], cksum 0x31e9 (correct), seq 2723:2847, ack 3342, win 65535, length 124
17:41:29.379659 IP (tos 0x8, ttl 64, id 22997, offset 0, flags [DF], proto TCP (6), length 76)
10.0.2.15.39480 > 178.234.29.197.22: Flags [P.], cksum 0xdcfc (incorrect -> 0x16d9), seq 3342:3378, ack 2847, win 63440, length 36
17:41:29.379672 IP (tos 0x8, ttl 64, id 22998, offset 0, flags [DF], proto TCP (6), length 100)
10.0.2.15.39480 > 178.234.29.197.22: Flags [P.], cksum 0xdd14 (incorrect -> 0x832f), seq 3378:3438, ack 2847, win 63440, length 60
17:41:29.379703 IP (tos 0x8, ttl 64, id 22999, offset 0, flags [DF], proto TCP (6), length 40)
10.0.2.15.39480 > 178.234.29.197.22: Flags [F.], cksum 0xcdc8 (incorrect -> 0x473e), seq 3438, ack 2847, win 63440, length 0
17:41:29.379846 IP (tos 0x0, ttl 64, id 58253, offset 0, flags [none], proto TCP (6), length 40)
178.234.29.197.22 > 10.0.2.15.39480: Flags [P.], cksum 0x3f4c (correct), ack 3378, win 65535, length 0
17:41:29.379846 IP (tos 0x0, ttl 64, id 58254, offset 0, flags [none], proto TCP (6), length 40)
178.234.29.197.22 > 10.0.2.15.39480: Flags [P.], cksum 0x3f10 (correct), ack 3438, win 65535, length 0
17:41:29.379846 IP (tos 0x0, ttl 64, id 58255, offset 0, flags [none], proto TCP (6), length 40)
178.234.29.197.22 > 10.0.2.15.39480: Flags [P.], cksum 0x3f0f (correct), ack 3439, win 65535, length 0
17:41:29.386248 IP (tos 0x0, ttl 64, id 58256, offset 0, flags [none], proto TCP (6), length 40)
178.234.29.197.22 > 10.0.2.15.39480: Flags [F.], cksum 0x3f0e (correct), seq 2847, ack 3439, win 65535, length 0
17:41:29.386266 IP (tos 0x8, ttl 64, id 0, offset 0, flags [DF], proto TCP (6), length 40)
10.0.2.15.39480 > 178.234.29.197.22: Flags [P.], cksum 0x473d (correct), ack 2848, win 63440, length 0
^C95 packets captured
95 packets received by filter
0 packets dropped by kernel

```

Рисунок 16.

14. Вновь подключаемся к удалённой системе. Процедуры подключения по ssh-ключу не отличаются друг от друга; процесс регистрации пользователя на удалённой системе осуществляется посредством разных команд, но в целом не имеет отличий.

15. Аналогично, с помощью команды scp произведём повторную передачу текстового файла new_text.txt на удалённый узел (рисунок 17).

```

tatyana@arianrod:~$ scp ./new_text.txt stud3@178.234.29.197:
stud3@178.234.29.197's password:
new_text.txt                                100% 10     2.6KB/s  00:00
tatyana@arianrod:~$

```

Рисунок 17.

Убедимся в наличии переданной копии файла на удалённом хосте (рисунок 18). Отличий в процедуре передачи файла не было обнаружено.

Левая панель				Правая панель			
Файл	Команда	Настройки		Файл	Команда	Настройки	
Имя	Размер	Время правки		Имя	Размер	Время правки	
./..	-ВВЕРХ-	янв 8 08:53		./..	-ВВЕРХ-	янв 8 08:53	
./.cache	4096	янв 29 20:03		./.cache	4096	янв 29 20:03	
./.config	4096	янв 29 20:03		./.config	4096	янв 29 20:03	
./.local	4096	янв 29 20:03		./.local	4096	янв 29 20:03	
./.ssh	4096	дек 13 2019		./.ssh	4096	дек 13 2019	
/conf	4096	дек 2 2019		/conf	4096	дек 2 2019	
/mail	4096	дек 2 2019		/mail	4096	дек 2 2019	
/tmp	4096	дек 2 2019		/tmp	4096	дек 2 2019	
/web	4096	дек 2 2019		/web	4096	дек 2 2019	
.bash_logout	220	сен 1 2015		.bash_logout	220	сен 1 2015	
.bashrc	3771	сен 1 2015		.bashrc	3771	сен 1 2015	
.profile	655	июн 24 2016		.profile	655	июн 24 2016	
file	10	янв 28 17:44		file	10	янв 28 17:44	
new_text.txt	10	янв 29 20:51		new_text.txt	10	янв 29 20:51	
stud_LR7.txt	90	янв 29 20:02		stud_LR7.txt	90	янв 29 20:02	

Рисунок 18.

16. Остановим анализатор сетевых пакетов и просмотрим содержимое файла ssh.log. Утилита tcpdump позволяет проверять заголовки пакетов TCP/IP и выводить одну строку для каждого из пакетов. Флаги TCP указывают на состояние соединения и могут содержать более одного значения. Рассмотрим на следующем примере: (рисунок 19). Флаг [P.] устанавливается при передаче пользовательских данных между удалёнными узлами(клиентом и сервером), флаг [F.] - устанавливается при нормальном закрытии соединения.

```

10.0.2.15.39480 > 178.234.29.197.22: Flags [P.], cksum 0xdcfc (incorrect -> 0xf04c), seq 3306:3342, ack 2723, win 63440, length 36
17:41:29.376231 IP (tos 0x0, ttl 64, id 58251, offset 0, flags [none], proto TCP (6), length 40)
178.234.29.197.22 > 10.0.2.15.39480: Flags [.], cksum 0x3fec (correct), ack 3342, win 65535, length 0
17:41:29.379612 IP (tos 0x0, ttl 64, id 58252, offset 0, flags [none], proto TCP (6), length 164)
178.234.29.197.22 > 10.0.2.15.39480: Flags [P.], cksum 0x31e9 (correct), seq 2723:2847, ack 3342, win 65535, length 124
17:41:29.379659 IP (tos 0x8, ttl 64, id 22997, offset 0, flags [DF], proto TCP (6), length 76)
10.0.2.15.39480 > 178.234.29.197.22: Flags [P.], cksum 0xdcfc (incorrect -> 0x16d9), seq 3342:3378, ack 2847, win 63440, length 36
17:41:29.379672 IP (tos 0x8, ttl 64, id 22998, offset 0, flags [DF], proto TCP (6), length 100)
10.0.2.15.39480 > 178.234.29.197.22: Flags [P.], cksum 0xdd14 (incorrect -> 0x832f), seq 3378:3438, ack 2847, win 63440, length 60
17:41:29.379703 IP (tos 0x8, ttl 64, id 22999, offset 0, flags [DF], proto TCP (6), length 40)
10.0.2.15.39480 > 178.234.29.197.22: Flags [F.], cksum 0xdcd8 (incorrect -> 0x473e), seq 3438, ack 2847, win 63440, length 0
17:41:29.379846 IP (tos 0x0, ttl 64, id 58253, offset 0, flags [none], proto TCP (6), length 40)
178.234.29.197.22 > 10.0.2.15.39480: Flags [.], cksum 0x3f4c (correct), ack 3378, win 65535, length 0
17:41:29.379846 IP (tos 0x0, ttl 64, id 58254, offset 0, flags [none], proto TCP (6), length 40)
178.234.29.197.22 > 10.0.2.15.39480: Flags [.], cksum 0x3f10 (correct), ack 3438, win 65535, length 0
17:41:29.379846 IP (tos 0x0, ttl 64, id 58255, offset 0, flags [none], proto TCP (6), length 40)
178.234.29.197.22 > 10.0.2.15.39480: Flags [.], cksum 0x3f0f (correct), ack 3439, win 65535, length 0
17:41:29.386248 IP (tos 0x0, ttl 64, id 58256, offset 0, flags [none], proto TCP (6), length 40)
178.234.29.197.22 > 10.0.2.15.39480: Flags [F.], cksum 0x3f0e (correct), seq 2847, ack 3439, win 65535, length 0
17:41:29.386266 IP (tos 0x8, ttl 64, id 0, offset 0, flags [DF], proto TCP (6), length 40)
10.0.2.15.39480 > 178.234.29.197.22: Flags [.], cksum 0x473d (correct), ack 2848, win 63440, length 0
^C95 packets captured
95 packets received by filter
0 packets dropped by kernel

```

Рисунок 19.

Вопросы для самопроверки.

Что такое ключ SSH? В чём преимущество их использования?

SSH сервер может выполнять аутентификацию пользователей с помощью различных алгоритмов. Хотя самым популярным и является аутентификация по паролю, так как она достаточно проста, но более безопасным и надёжным является аутентификация по ключу SSH.

SSH-ключи используются для идентификации клиента при подключении к серверу по SSH-протоколу. SSH-ключи представляют собой пару — закрытый и открытый ключ. Закрытый должен храниться в закрытом доступе у клиента, открытый отправляется на сервер и размещается в файле `authorized_keys`.

Как сгенерировать ключи ssh в разных ОС?

Генерация ключей ssh в linux.

Пару ключей для использования ssh можно сгенерировать с помощью команды `ssh-keygen`. Если не задавать других параметров, ключи сохраняются по пути `/.ssh/` в формате `<name>` для секретного ключа и `<name>.pub` для открытого. Ключи также можно защитить паролем при создании. Если указать пароль пустым, он не будет использоваться. Сменить используемый пароль можно с помощью `ssh-keygen -p`.

Генерация ключей ssh в Windows.

В ОС Windows подключение к удалённым серверам по SSH возможно с помощью клиента Putty. Для генерации ssh-ключа необходимо запустить файл `puttygen.exe`. Выбрать тип ключа SSH-2 RSA и длину 2048 бит, а затем сгенерировать (кнопка Generate) и сохранить пару ключей на локальной машине (кнопки Save public key и Save private key).

Возможно ли из "секретного"ключа сгенерировать "публичный"и/или наоборот?

Из "секретного"ключа можно получить в результате генерации "публичный но обратный процесс произвести нельзя.

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

Генерируемые ssh-ключи являются уникальными и не могут совпадать при их повторном создании. Дополнительная возможность указания пароля на "секретный"ключ является мерой для большей устойчивости ко взлому и на уникальность ключей не влияет.

Перечислите доступные ключи для ssh-keygen.exe

Программа `ssh-keygen` может генерировать четыре типа ключей:

- 1) dsa;
- 2) ecdsa;
- 3) ed25519;
- 4) rsa;

Чтобы выбрать любой из этих типов, используется опция `-t`. Тип `rsa` подразумевается по умолчанию (то есть генерацию ключей можно запустить без опции `-t`).

Можно ли использовать один "секретный" ключ доступа с разных ОС, установленных на одном ПК/на разных ПК?

Можно использовать один "секретный" ключ доступа с разных ОС, установленных на одном или на разных ПК.

Возможно ли организовать подключение "по ключу" ssh к системе с ОС Windows, в которой запущен OpenSSH сервер?

Организовать подключение "по ключу" ssh к системе с ОС Windows, в которой запущен OpenSSH сервер, возможно.

Какие известные Вам сервисы сети Интернет позволяют организовать доступ к ресурсам посредством SSH ключей?

WWW-серверы, FTP-серверы, почтовики, шлюзы

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

С помощью ssh можно производить удаленное управление хостом и туннелирование TCP-соединений. В отличие от telnet и rlogin весь трафик, передаваемый по ssh, шифруется. SSH позволяет безопасно передавать в незащищённой среде практически любой другой сетевой протокол. Таким образом, можно не только удалённо работать на компьютере через командную оболочку, но и передавать по шифрованному каналу звуковой поток или видео.

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

Для удаленного доступа к Linux используются два протокола **telnet** и **SSH**.

Telnet - протокол линии передачи данных Интернет, который даёт возможность компьютеру функционировать как терминал, работающий под управлением удалённого компьютера.

Нежелательно использование протокола telnet в системах, для которых важна безопасность, таких как общественный Интернет. Сессии telnet не поддерживают шифрование данных. Это означает, что любой, кто имеет доступ к любому маршрутизатору, коммутатору или шлюзу в сети между двумя удалёнными компьютерами, соединёнными сессией связи по протоколу telnet, может перехватить проходящие пакеты и легко получить логин и пароль для доступа в систему (или завладеть любой другой информацией, которой обмениваются эти компьютеры).

SSH - (Secure Shell) — сетевой протокол, позволяющий производить удалённое управление компьютером и передачу файлов. Сходен по функциональности с протоколом telnet, однако использует алгоритмы шифрования передаваемой информации.

Недостатки telnet привели к очень быстрому отказу от использования этого протокола в пользу более безопасного и функционального протокола SSH. SSH предоставляет все те функциональные возможности, которые представлялись в telnet, с добавлением эффективного кодирования с целью предотвращения перехвата таких данных, как логины и пароли. Введенная в протоколе SSH система аутентификации с использованием публичного ключа гарантирует, что удаленный компьютер действительно является тем, за кого себя выдает.

Опишите способы установления соединения при использовании протокола SSH? Охарактеризуйте положительные и отрицательные аспекты приведенных методов.

В ssh существует несколько способов авторизации. Можно каждый раз вводить пароль пользователя или использовать безопасный и надёжный способ - ssh-ключи.

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

Область применения протокола SSH практически неограничена. Исходя из его основной функции - удаленного входа в операционную систему, протокол используют:

- 1) системные администраторы для удаленной настройки компьютеров локальной сети;
- 2) для настройки почтовых служб (повышает безопасность данных);
- 3) для скрытого обмена внутри сети массивными файлами;
- 4) для интернет-игр;

Вывод

В ходе лабораторной работы было изучено программное обеспечение удалённого доступа к распределённым системам обработки данных.

Список литературы

- [1] Львовский, С.М. Набор и верстка в системе ЛАТ_ΕX [Текст] / С.М. Львовский. М.: МЦНМО, 2006. — 448 с.
- [2] Хабр. Используем tcpdump для анализа и перехвата сетевого трафика: <https://habr.com/ru/company/alexhost/blog/531170/> (дата обращения: 30.01.2021). - Текст: электронный.
- [3] Losst. Удалённое управление telnet: https://losst.ru/kak-polzovatsya-telnet6_Удалённое_управление_telnet (дата обращения: 30.01.2021). - Текст: электронный.