

Липецкий государственный технический университет

Факультет автоматизации и информатики

Кафедра Автоматизированных систем управления

ЛАБОРАТОРНАЯ РАБОТА №7

По дисциплине «ОС Linux»

Работа с SSH

Студент

Печенкин Д.В.

Группа ПИ-18

Руководитель

Доцент

Кургасов В.В.

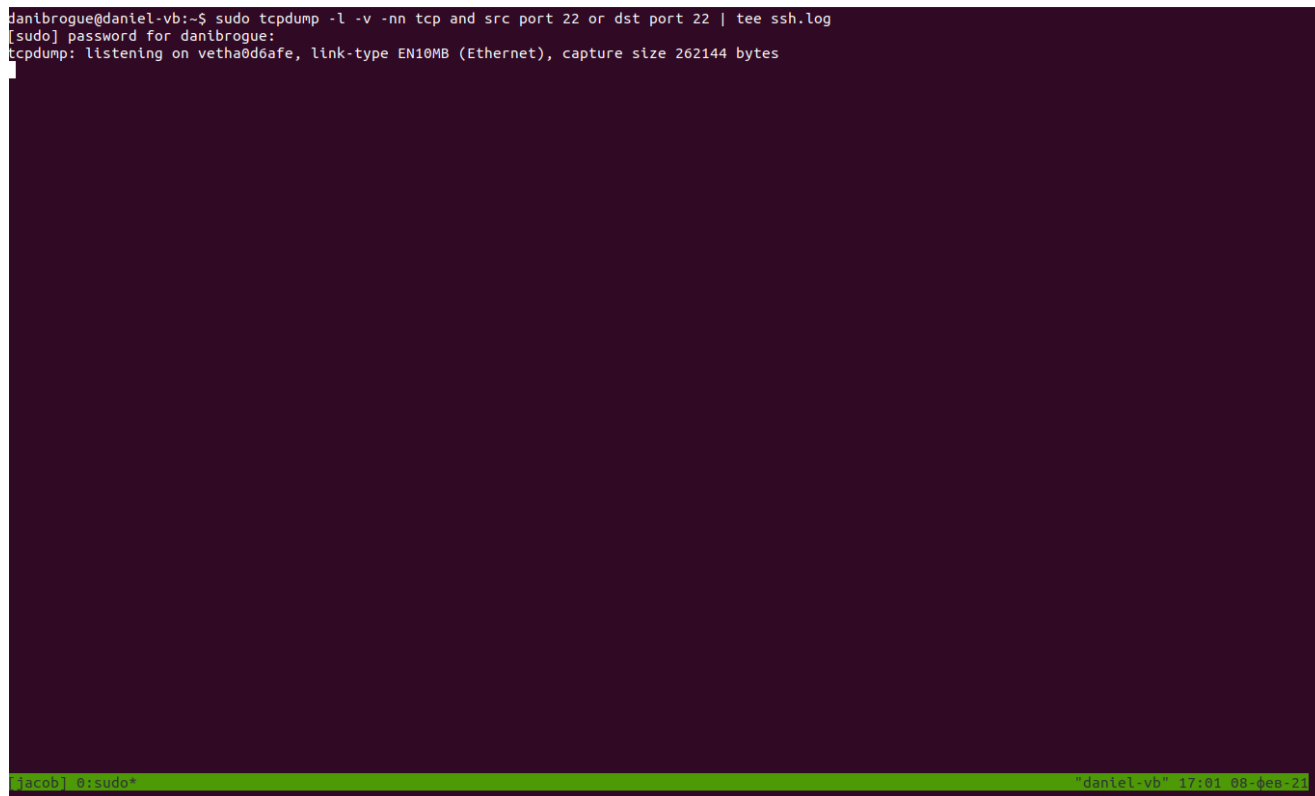
Липецк 2021г

Цель работы

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

Ход работы

С помощью утилиты `tmux`, создадим новое окно с помощью комбинации клавиш “Ctrl+b C” и запустим анализатор трафика `tcpdump` и введем команду «`sudo tcpdump -l -v -nn tcp and src port 22 or dst port 22 | tee ssh.log`» для вывода отфильтрованных IP-пакетов на терминал и сохраним данные в файл `ssh.log`.

A screenshot of a terminal window with a dark purple background. The prompt is 'danibrogue@daniel-vb:~\$'. The command entered is 'sudo tcpdump -l -v -nn tcp and src port 22 or dst port 22 | tee ssh.log'. The terminal shows the password prompt '[sudo] password for danibrogue:' and the output 'tcpdump: listening on vetha0d6afe, link-type EN10MB (Ethernet), capture size 262144 bytes'. The terminal is mostly empty, indicating it is listening for traffic. At the bottom, there is a green status bar with the text '[jacob] 0:sudo*' on the left and '"daniel-vb" 17:01 08-фев-21' on the right.

```
danibrogue@daniel-vb:~$ sudo tcpdump -l -v -nn tcp and src port 22 or dst port 22 | tee ssh.log
[sudo] password for danibrogue:
tcpdump: listening on vetha0d6afe, link-type EN10MB (Ethernet), capture size 262144 bytes

[jacob] 0:sudo* "daniel-vb" 17:01 08-фев-21
```

Рисунок 1 – Запуск анализатор трафика

Переключившись на первое окно терминального мультиплексора, с помощью команды «`ssh -l stud11 kurgasov.ru`» (по варианту), после чего введем пароль для входа на удаленную систему.

```
danibrogue@daniel-vb:~$ ssh -l stud5 kurgasov.ru
The authenticity of host 'kurgasov.ru (178.234.29.197)' can't be established.
ECDSA key fingerprint is SHA256:c7y8uU2/zFt5w6UuLfUeRk/OhPMih9uki+EYZVo1qik.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'kurgasov.ru,178.234.29.197' (ECDSA) to the list of known hosts.
stud5@kurgasov.ru'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

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

*** Требуется перезагрузка системы ***
Last login: Fri Dec 13 08:55:16 2019 from 213.87.159.228
$
```

```
[jacob] 0:sudo- i:ssh*
```

```
"daniel-vb" 17:04 08-фев-21
```

Рисунок 2 – Вход на удаленную систему

Введем команду «uname -a» для вывода информации об удаленной системе.

```
$ 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 3 – Ввод команды на удаленной системе

Создадим для удобства ещё одно окно, создадим текстовый файл с содержанием ФИО и номера лабораторной работы и с помощью команды «scp ~/lab7 stud11@kurgasov.ru:/home/stud11» передать файл по зашифрованному каналу на удаленную систему.

```
GNU nano 4.8
FIO: Pechenkin Danila Vladimirovich
Lab: Lab7
```

Рисунок 4 – Содержимое текстового файла lab7

```
danibrogue@daniel-vb:~$ scp ~/lab7 stud5@kurgasov.ru:/home/stud5
stud5@kurgasov.ru's password:
lab7
danibrogue@daniel-vb:~$
```

100% 46 1.1KB/s 00:00

Рисунок 5 – Передача текстового файла

Проверим наличие файла на удаленной системе воспользовавшись файловым менеджером «Midnight Commander» (команда mc).

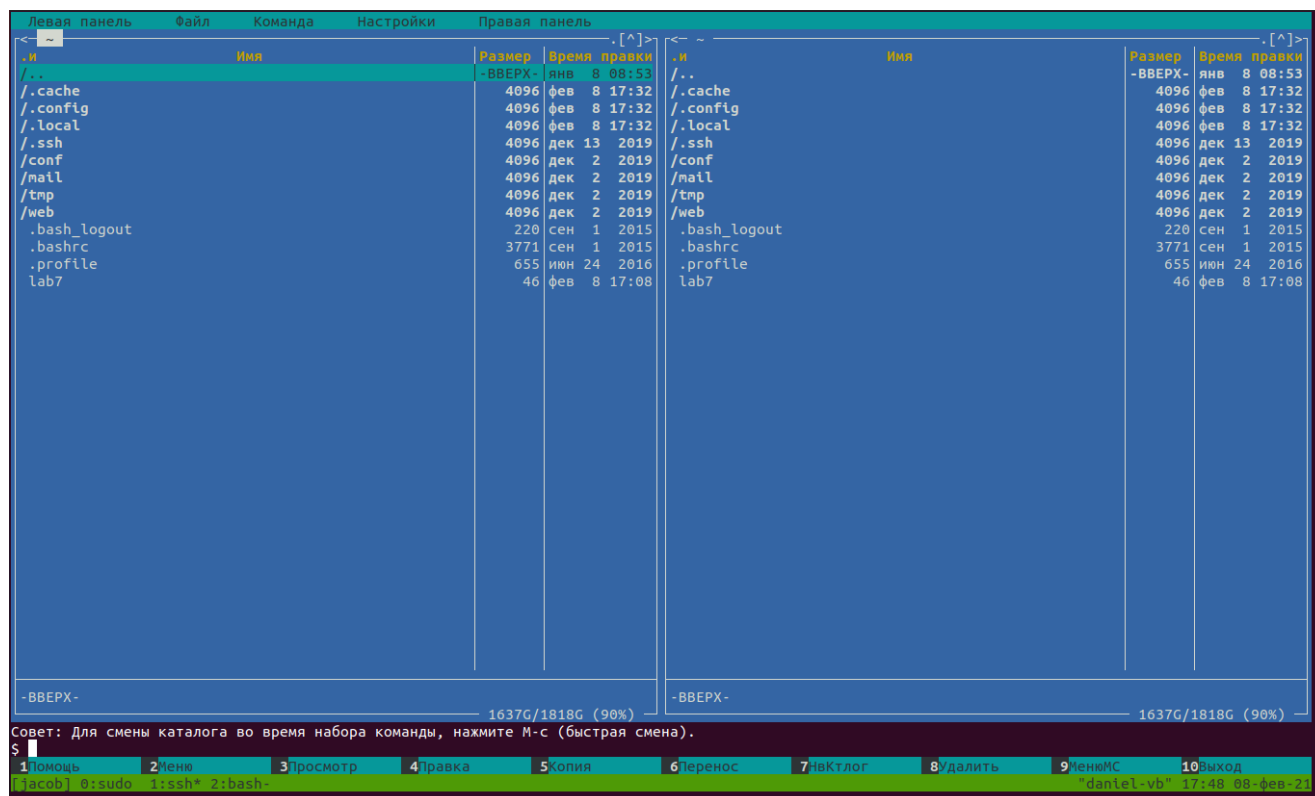


Рисунок 6 – Проверка наличия переданного файла

Выйдем из удаленного узла командой exit и сформируем зашифрованные ключи, воспользовавшись командой ssh-keygen.

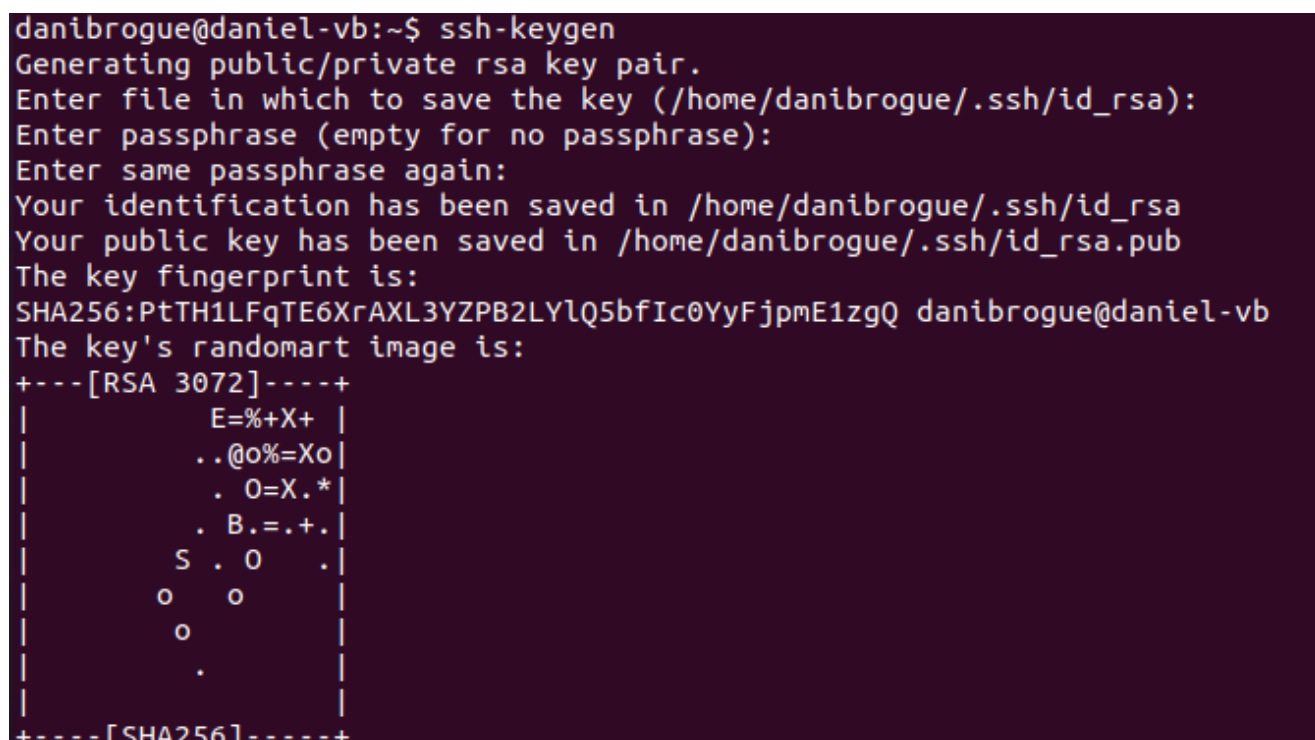


Рисунок 7 – Формирование зашифрованных ключей

Передадим публичный ключ ssh удаленной системе с помощью команды:

```
ssh-copy-id -i ~/.ssh/id_rsa.pub stud11@kurgasov.ru
```

```
danibroque@daniel-vb:~$ ssh-copy-id -i ~/.ssh/id_rsa.pub stud5@kurgasov.ru
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/danibroque/.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 the new keys
stud5@kurgasov.ru's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh 'stud5@kurgasov.ru'"
and check to make sure that only the key(s) you wanted were added.

danibroque@daniel-vb:~$
```

Рисунок 8 – Передача публичного ключа ssh

Подключимся к удаленной системе командой ssh -l stud11 kurgasov.ru

```
danibroque@daniel-vb:~$ ssh -l stud5 kurgasov.ru
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

12 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 Feb  8 18:22:20 2021 from 94.181.28.7
$
```

Рисунок 9 – Подключение к удаленной системе

После передачи публичного ключа ssh, вход произошел без ввода пароля.

Передадим ещё один текстовый файл на удаленный узел и проверим его наличие.

```
danibrogue@daniel-vb:~$ scp ~/lab7_1 stud5@kurgasov.ru:/home/stud5
lab7_1
danibrogue@daniel-vb:~$
```

Рисунок 10 – Передача файла lab7_1

Как можно заметить, ввод пароля не потребовался благодаря SSH

```
$ ls
conf lab7 lab7_1 mail tmp web
$
```

Рисунок 11 – Проверка наличия файла.

Остановим анализатор сетевых пакетов, воспользовавшись комбинацией Ctrl+c и просмотрим содержимое файла ssh.log.

```
10.0.2.15.43282 > 178.234.29.197.22: Flags [.], cksum 0xcdcd8 (incorrect ->
0xe59d), ack 6307, win 63440, length 0
19:53:08.589011 IP (tos 0x0, ttl 64, id 4795, offset 0, flags [none], proto TCP
(6), length 84)
178.234.29.197.22 > 10.0.2.15.43282: Flags [P.], cksum 0x20ed (correct), se
q 6307:6351, ack 6846, win 65535, length 44
19:53:08.589032 IP (tos 0x10, ttl 64, id 12527, offset 0, flags [DF], proto TCP
(6), length 40)
10.0.2.15.43282 > 178.234.29.197.22: Flags [.], cksum 0xcdcd8 (incorrect ->
0xe571), ack 6351, win 63440, length 0
^C2317 packets captured
2317 packets received by filter
0 packets dropped by kernel
```

Рисунок 12 – Остановка анализатора

```
GNU nano 4.8 ssh.log
10.0.2.15.43186 > 178.234.29.197.22: Flags [P.], cksum 0xdce8 (incorrect ->
17:41:48.561285 IP (tos 0x0, ttl 64, id 3264, offset 0, flags [none], proto TC
178.234.29.197.22 > 10.0.2.15.43186: Flags [.], cksum 0xb2ca (correct), ac
17:41:48.586402 IP (tos 0x0, ttl 64, id 32649, offset 0, flags [DF], proto TCP
10.0.2.15.43186 > 178.234.29.197.22: Flags [P.], cksum 0xdd04 (incorrect ->
17:41:48.586778 IP (tos 0x0, ttl 64, id 3265, offset 0, flags [none], proto TC
178.234.29.197.22 > 10.0.2.15.43186: Flags [.], cksum 0xb29e (correct), ac
17:41:48.617427 IP (tos 0x0, ttl 64, id 3266, offset 0, flags [none], proto TC
178.234.29.197.22 > 10.0.2.15.43186: Flags [P.], cksum 0x77b9 (correct), s
17:41:48.617452 IP (tos 0x0, ttl 64, id 32650, offset 0, flags [DF], proto TCP
10.0.2.15.43186 > 178.234.29.197.22: Flags [.], cksum 0xcdcd8 (incorrect ->
17:41:48.617602 IP (tos 0x0, ttl 64, id 32651, offset 0, flags [DF], proto TCP
10.0.2.15.43186 > 178.234.29.197.22: Flags [P.], cksum 0xddic (incorrect ->
17:41:48.617939 IP (tos 0x0, ttl 64, id 3267, offset 0, flags [none], proto TC
178.234.29.197.22 > 10.0.2.15.43186: Flags [.], cksum 0xb22e (correct), ac
17:41:48.647204 IP (tos 0x0, ttl 64, id 3268, offset 0, flags [none], proto TC
178.234.29.197.22 > 10.0.2.15.43186: Flags [P.], cksum 0x5ae6 (correct), s
17:41:48.689375 IP (tos 0x0, ttl 64, id 32652, offset 0, flags [DF], proto TCP
10.0.2.15.43186 > 178.234.29.197.22: Flags [.], cksum 0xcdcd8 (incorrect ->
17:42:18.466005 IP (tos 0x0, ttl 64, id 32653, offset 0, flags [DF], proto TCP
10.0.2.15.43186 > 178.234.29.197.22: Flags [P.], cksum 0xdd6c (incorrect ->
17:42:18.466432 IP (tos 0x0, ttl 64, id 3269, offset 0, flags [none], proto TC
178.234.29.197.22 > 10.0.2.15.43186: Flags [.], cksum 0xb166 (correct), ac
17:42:18.517105 IP (tos 0x0, ttl 64, id 3270, offset 0, flags [none], proto TC

^C Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify
^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell
```

Рисунок 13 – Открытие файла ssh.log

Вывод

В ходе выполнения лабораторной работы было изучено программное обеспечение удаленного доступа к определенным системам обработки данных.

Контрольные вопросы

1) Удаленный доступ — функция, дающая пользователю возможность подключаться к компьютеру с помощью другого устройства через интернет практически отовсюду. Пользователь работает с файлами и программами точно так же, как если бы он находился возле этого компьютера. Особенно пригодится эта функция тем компаниям, где большинство сотрудников находится за пределами офиса, на частичном фрилансе, аутсорсинге или в командировках, но при этом они нуждаются в обновлении рабочей информации, просмотре корпоративной почты и пр. Им не нужно будет скачивать все необходимые для работы данные на внешний носитель или отправлять их по почте — достаточно связаться с офисным компьютером.

2) SSH и Telnet - это сетевые протоколы, которые позволяют пользователям входить в удаленные системы и выполнять на них команды.

Доступ к командной строке удаленного хоста одинаков для обоих протоколов, но основное различие этих протоколов зависит от меры безопасности каждого из них. SSH более защищен, чем Telnet.

По умолчанию SSH использует порт 22, а Telnet использует порт 23 для связи, и оба используют стандарт TCP.

SSH отправляет все данные в зашифрованном формате, а Telnet отправляет данные в виде обычного текста. Поэтому SSH использует безопасный канал для передачи данных по сети, а Telnet использует обычный способ подключения к сети и связи.

SSH использует шифрование с открытым ключом для аутентификации удаленных пользователей, а Telnet не использует механизмов аутентификации.

Учитывая безопасность, доступную в каждом протоколе, SSH подходит для использования в общедоступных сетях, а Telnet больше подходит для частных сетей.

3) Существует несколько конфигураций:

1. Порт 22, авторизация по паролю, без защиты. В данной конфигурации Защита- высокая и потери от флуда - высокие. (Расход ресурсов сервера на обработку запросов, обычно идущих на 22 порт)

2. 22 порт, авторизация по ключам, без защиты. Защита – средняя, потери от флуда- высокие.

3. 22 порт, авторизация по ключам, защита на основе ограничения неудачных попыток авторизации. Защита – низкая, потери от флуда - средние

4. Нестандартный порт, авторизация по паролю, без защиты. Защита – высокая, потери от флуда – низкие

5. Нестандартный порт, авторизация по ключам, без защиты. Защита – средняя, потери от флуда – низкие

6. Нестандартный порт, авторизация по ключам, защита на основе ограничения неудачных попыток авторизации. Защита – низкая, потери от флуда – низкие.

4) Удаленный доступ используют системные администраторы для управления системой и устранения сбоев в ее работе, и руководители, желающие проконтролировать процесс выполнения задачи своими подчиненными. Применяется он и для дистанционного обучения в образовательных учреждениях.

5) Распространенные сетевые службы, основанные на использовании шифрованного соединения по протоколу SSH:

OpenSSH, PuTTY/ KiTTY, SecureCRT, Xshell. Службы передачи файлов по безопасному туннелю можно использовать для передачи паролей.