



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «ЛИПЕЦКИЙ
ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет автоматизации и информатики
Кафедра автоматизированных систем управления

ЛАБОРАТОРНАЯ РАБОТА №6
по курсу “ОС Linux”

Студент ПИ-21-1

(подпись, дата)

Морозов Д. С.

Руководитель

(подпись, дата)

Кургасов В.В.

Липецк 2023

Цель работы	3
Ход работы.....	4
1. Запуск анализатора трафика tcpdump на 23 порту	4
2. Установка соединения с удаленной системой по порту 23.....	5
3. Запуск tcpdump на порту 22	5
4. Подключение по ssh к удаленной системе	6
5. Вывод информации об удаленной системе	6
6. Создание файла и передача его на удаленную систему по шифровальному каналу	6
7. Формирование зашифрованного ключа	8
8. Передача зашифрованного ключа на удаленную систему перед этим создав там необходимую директорию и файл	8
9. Попытка подключения по ssh к удаленной системе.....	9
10. Повторная передача файла	9
11. Содержимое файлов telnet.log и ssh.log.....	10
Ответы на контрольные вопросы.....	11

Цель работы

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

Ход работы

1. Запуск анализатора трафика tcpdump на 23 порту



Рисунок 1 – Запуск tmux

```
daniil@ubuntuserver:~$ sudo tcpdump -l -v -nn tcp and src port 23 or dst port 23 | tee telnet.log
[sudo] password for daniil:
tcpdump: listening on enp0s3, link-type EN10MB (Ethernet), snapshot length 262144 bytes
```

Рисунок 2 – Запуск tcpdump по порту 23

2. Установка соединения с удаленной системой по порту 23

Для удаленной системы используется вторая ВМ соединенная с основной ВМ по сетевому мосту.

```
daniil@ubunto:~$ telnet 10.0.2.15 23
Trying 10.0.2.15...
telnet: Unable to connect to remote host: Connection refused
```

Рисунок 3 – Подключение к удаленной системе

Получаем ошибку подключения.

3. Запуск tcpdump на порту 22

```
sudo tcpdump -l -v -nn tcp and src port 22 or dst port 22 | tee telnet.log
```

Рисунок 4 – Запуск tcpdump на порту 22

```
daniil@ubunto:~$ telnet 10.0.2.15 22
Trying 10.0.2.15...
Connected to 10.0.2.15.
Escape character is '^]'.
SSH-2.0-OpenSSH_8.9p1 Ubuntu-3ubuntu0.4
```

Рисунок 5 – подключение к удаленной системе по порту 22 через telnet

4. Подключение по ssh к удаленной системе

```
daniil@ubuntuserver:~$ ssh luke@10.0.2.15
The authenticity of host '10.0.2.15 (10.0.2.15)' can't be established.
ED25519 key fingerprint is SHA256:pgj9Dmo9qEmcQp86Uw90GkryId2P779d3Hd3bX6y1Ug.
This host key is known by the following other names/addresses:
  ~/.ssh/known_hosts:1: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.2.15' (ED25519) to the list of known hosts.
luke@10.0.2.15's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-91-generic x86_64)

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

System information as of Вc 07 янв 2024 19:20:09 UTC

System load:                0.0068359375
Usage of /:                  49.6% of 47.93GB
Memory usage:               18%
Swap usage:                 0%
Processes:                  163
Users logged in:            1
IPv4 address for br-14bc085b7755: 192.168.160.1
IPv4 address for br-9e7160926254: 172.19.0.1
IPv4 address for br-b1f1c4ec0d30: 172.21.0.1
IPv4 address for br-be11b3132591: 172.20.0.1
```

Рисунок 6 – Подключение по ssh

5. Вывод информации об удаленной системе

```
luke@ubuntuserver:~$ uname -a
Linux ubuntuserver 5.15.0-91-generic #101-Ubuntu SMP Tue Nov 14 13:30:08 UTC 2023 x86_64 x86_64 x86_64 GNU/Linux
```

Рисунок 7 – uname -a

6. Создание файла и передача его на удаленную систему по шифрованному каналу

```
GNU nano 6.2                                     lr6
Morozov Daniil PI-21-1
LR6|
```

Рисунок 8 – Содержимое файла lr6

```
daniil@ubuntuserver:~$ scp lr6 luke@10.0.2.15:/home/luke
luke@10.0.2.15's password:
Permission denied, please try again.
luke@10.0.2.15's password:
lr6                                     100% 27    30.5KB/s   00:00
daniil@ubuntuserver:~$
```

Рисунок 9 – Передача файла

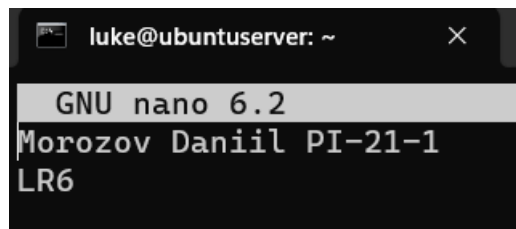


Рисунок 10 – Проверка наличия и содержимого файла на удаленной системе

7. Формирование зашифрованного ключа

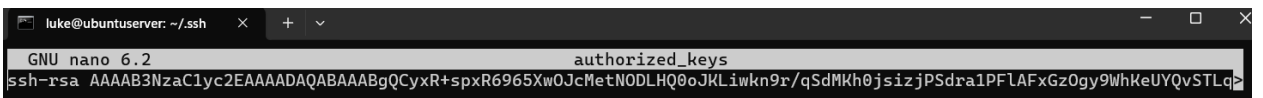
```
daniil@ubuntuuser:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/daniil/.ssh/id_rsa):
/home/daniil/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/daniil/.ssh/id_rsa
Your public key has been saved in /home/daniil/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:cD6uNEHbBfrCn0ZECWI2SP9hfqDIOaaMPPrD8kLEy8Sw daniil@ubuntuuser
The key's randomart image is:
+---[RSA 3072]-----+
| .. = .. o .      |
| .+ o o . .      |
| . B o .          |
| . o B @ .        |
| ..* . B S        |
| +*+. B o         |
| E*o   o =        |
| +=. . +          |
| .o. .            |
+---[SHA256]-----+
daniil@ubuntuuser:~$
```

Рисунок 11 – формирование зашифрованного ключа

8. Передача зашифрованного ключа на удаленную систему перед этим создав там необходимую директорию и файл

```
scp: /home/daniil/.ssh/authorized_keys: No such file or directory
daniil@ubuntuuser:~$ ssh luke@10.0.2.15 "mkdir -p ~/.ssh && touch ~/.ssh/authorized_keys"
luke@10.0.2.15's password:
daniil@ubuntuuser:~$ scp /home/daniil/.ssh/id_rsa.pub luke@10.0.2.15:~/.ssh/authorized_keys
luke@10.0.2.15's password:
id_rsa.pub                                100% 573   514.5KB/s   00:00
daniil@ubuntuuser:~$
```

Рисунок 12 – Передача ключа на удаленную систему



```
luke@ubuntuuser: ~/.ssh
GNU nano 6.2 authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGCyxR+spXR6965Xw0JcMetNODLHQ0oJKLiwn9r/qSdMKh0jsizjPSdra1PFLAFxGz0gy9WhKeUYQvSTLq
```

Рисунок 13 – Проверка передачи ключа

9. Попытка подключения по ssh к удаленной системе

```
daniil@ubuntuserver:~$ ssh luke@10.0.2.15
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-91-generic x86_64)

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

System information as of Вc 07 янв 2024 19:49:52 UTC

System load:                0.00048828125
Usage of /:                  49.6% of 47.93GB
Memory usage:               18%
Swap usage:                 0%
Processes:                  163
Users logged in:            2
IPv4 address for br-14bc085b7755: 192.168.160.1
IPv4 address for br-9e7160926254: 172.19.0.1
IPv4 address for br-b1f1c4ec0d30: 172.21.0.1
IPv4 address for br-be11b3132591: 172.20.0.1
IPv4 address for br-d4bb05fe62dc: 172.23.0.1
IPv4 address for docker0:    172.17.0.1
IPv4 address for enp0s3:     10.0.2.15

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Расширенное поддержание безопасности (ESM) для Applications выключено.

27 обновлений может быть применено немедленно.
Чтобы просмотреть дополнительные обновления выполните: apt list --upgradable

Включите ESM Apps для получения дополнительных будущих обновлений безопасности.
Смотрите https://ubuntu.com/esm или выполните: sudo pro status

Last login: Sun Jan  7 19:20:09 2024 from 10.0.2.15
luke@ubuntuserver:~$
```

Рисунок 14 – Подключение по ssh

При подключение по ssh пароль не требуется.

10. Повторная передача файла

```
daniil@ubuntuserver:~$ scp lr6 luke@10.0.2.15:/home/luke
lr6
 100% 27 26.5KB/s 00:00
daniil@ubuntuserver:~$
```

Рисунок 15 – Передача файла с указанием шифрованного ключа

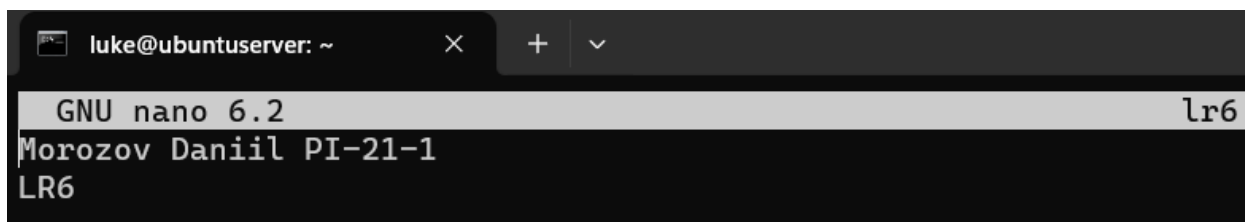


Рисунок 16 – Проверка передачи файла

При передачи файла не требуется пароль.

11. Содержимое файлов telnet.log и ssh.log

```

10.0.2.2.49983 > 10.0.2.15.22: Flags [..], cksum 0xc2af (correct), ack 32476, win 65535, length 0
20:00:51.391442 IP (tos 0x10, ttl 64, id 1437, offset 0, flags [DF], proto TCP (6), length 556)
10.0.2.15.22 > 10.0.2.2.49983: Flags [P.], cksum 0x1a2f (incorrect -> 0xb91d), seq 32476:32992, ack 37, win 62780, l
length 516
20:00:51.391972 IP (tos 0x0, ttl 64, id 4075, offset 0, flags [none], proto TCP (6), length 40)
10.0.2.2.49983 > 10.0.2.15.22: Flags [..], cksum 0xc0ab (correct), ack 32992, win 65535, length 0
20:00:51.495024 IP (tos 0x10, ttl 64, id 1438, offset 0, flags [DF], proto TCP (6), length 556)
10.0.2.15.22 > 10.0.2.2.49983: Flags [P.], cksum 0x1a2f (incorrect -> 0xc381), seq 32992:33508, ack 37, win 62780, l
length 516
20:00:51.595484 IP (tos 0x0, ttl 64, id 4076, offset 0, flags [none], proto TCP (6), length 40)
10.0.2.2.49983 > 10.0.2.15.22: Flags [..], cksum 0xbea7 (correct), ack 33508, win 65535, length 0
20:00:51.599111 IP (tos 0x10, ttl 64, id 1439, offset 0, flags [DF], proto TCP (6), length 556)
10.0.2.15.22 > 10.0.2.2.49983: Flags [P.], cksum 0x1a2f (incorrect -> 0xd10a), seq 33508:34024, ack 37, win 62780, l
length 516
20:00:51.599593 IP (tos 0x0, ttl 64, id 4077, offset 0, flags [none], proto TCP (6), length 40)
10.0.2.2.49983 > 10.0.2.15.22: Flags [..], cksum 0xbca3 (correct), ack 34024, win 65535, length 0
20:00:51.702983 IP (tos 0x10, ttl 64, id 1440, offset 0, flags [DF], proto TCP (6), length 556)
10.0.2.15.22 > 10.0.2.2.49983: Flags [P.], cksum 0x1a2f (incorrect -> 0xeef3), seq 34024:34540, ack 37, win 62780, l
length 516
20:00:51.703271 IP (tos 0x0, ttl 64, id 4078, offset 0, flags [none], proto TCP (6), length 40)
10.0.2.2.49983 > 10.0.2.15.22: Flags [..], cksum 0xba9f (correct), ack 34540, win 65535, length 0
20:00:51.807046 IP (tos 0x10, ttl 64, id 1441, offset 0, flags [DF], proto TCP (6), length 556)
10.0.2.15.22 > 10.0.2.2.49983: Flags [P.], cksum 0x1a2f (incorrect -> 0xa213), seq 34540:35056, ack 37, win 62780, l
length 516
20:00:51.807312 IP (tos 0x0, ttl 64, id 4079, offset 0, flags [none], proto TCP (6), length 40)
10.0.2.2.49983 > 10.0.2.15.22: Flags [..], cksum 0xb89b (correct), ack 35056, win 65535, length 0

```

Рисунок 17 – ssh.log

```

80, length 516
18:53:33.481526 IP (tos 0x0, ttl 64, id 2385, offset 0, flags [none], proto TCP (6), length 40)
10.0.2.2.49983 > 10.0.2.15.22: Flags [..], cksum 0x7d0a (correct), ack 1031620, win 65535, length 0
18:53:33.584681 IP (tos 0x10, ttl 64, id 627, offset 0, flags [DF], proto TCP (6), length 556)
10.0.2.15.22 > 10.0.2.2.49983: Flags [P.], cksum 0x1a2f (incorrect -> 0xa326), seq 1031620:1032136, ack 685, win 627
80, length 516
18:53:33.585081 IP (tos 0x0, ttl 64, id 2386, offset 0, flags [none], proto TCP (6), length 40)
10.0.2.2.49983 > 10.0.2.15.22: Flags [..], cksum 0x7b06 (correct), ack 1032136, win 65535, length 0
18:53:33.687275 IP (tos 0x10, ttl 64, id 628, offset 0, flags [DF], proto TCP (6), length 556)
10.0.2.15.22 > 10.0.2.2.49983: Flags [P.], cksum 0x1a2f (incorrect -> 0xc182), seq 1032136:1032652, ack 685, win 627
80, length 516
18:53:33.687732 IP (tos 0x0, ttl 64, id 2387, offset 0, flags [none], proto TCP (6), length 40)
10.0.2.2.49983 > 10.0.2.15.22: Flags [..], cksum 0x7902 (correct), ack 1032652, win 65535, length 0
18:53:33.792738 IP (tos 0x10, ttl 64, id 629, offset 0, flags [DF], proto TCP (6), length 556)
10.0.2.15.22 > 10.0.2.2.49983: Flags [P.], cksum 0x1a2f (incorrect -> 0x982a), seq 1032652:1033168, ack 685, win 627
80, length 516
18:53:33.793233 IP (tos 0x0, ttl 64, id 2388, offset 0, flags [none], proto TCP (6), length 40)
10.0.2.2.49983 > 10.0.2.15.22: Flags [..], cksum 0x76fe (correct), ack 1033168, win 65535, length 0
18:53:33.899248 IP (tos 0x10, ttl 64, id 630, offset 0, flags [DF], proto TCP (6), length 556)
10.0.2.15.22 > 10.0.2.2.49983: Flags [P.], cksum 0x1a2f (incorrect -> 0xfc1a), seq 1033168:1033684, ack 685, win 627
80, length 516
18:53:33.899744 IP (tos 0x0, ttl 64, id 2389, offset 0, flags [none], proto TCP (6), length 40)
10.0.2.2.49983 > 10.0.2.15.22: Flags [..], cksum 0x74fa (correct), ack 1033684, win 65535, length 0
18:53:34.003101 IP (tos 0x10, ttl 64, id 631, offset 0, flags [DF], proto TCP (6), length 556)
10.0.2.15.22 > 10.0.2.2.49983: Flags [P.], cksum 0x1a2f (incorrect -> 0x9fd0), seq 1033684:1034200, ack 685, win 627
80, length 516
18:53:34.003482 IP (tos 0x0, ttl 64, id 2390, offset 0, flags [none], proto TCP (6), length 40)
10.0.2.2.49983 > 10.0.2.15.22: Flags [..], cksum 0x72f6 (correct), ack 1034200, win 65535, length 0

```

Рисунок 18 – telnet.log

Ответы на контрольные вопросы

- 1) Определите основные цели и задачи решаемые с помощью ПО удаленного доступа?
- 2) Выделите отличительные особенности между режимами работы удаленного доступа по протоколам TELNET и SSH?
- 3) Опишите способы установления соединения при использовании протокола SSH? Охарактеризуйте положительные и отрицательные аспекты приведенных методов.
- 4) Основываясь на заданиях лабораторной работы, приведите практический пример использования систем удаленного доступа?
- 5) Перечислите распространенные сетевые службы, основанные на использовании шифрованного соединения по протоколу SSH? Приведите пример использования службы передачи файлов по безопасному туннелю?
- 6) Что такое ключ ssh? В чем преимущество их использования?
- 7) Как сгенерировать ключи ssh в разных ОС?
- 8) Возможно ли из «секретного» ключа сгенерировать «публичный» и/или наоборот?
- 9) Будут ли отличаться пары ключей, сгенерированные на одном ПК несколько раз с исходными условиями (наличие/отсутствие пароля на «секретный» ключ и т.п.)
- 10) Перечислите доступные ключи для ssh-keygen.exe
- 11) Можно ли использовать один «секретный» ключ доступа с разных ОС, установленных на одном ПК/на разных ПК?
- 12) Возможно ли организовать подключение «по ключу» ssh к системе с ОС Windows, в которой запущен OpenSSH сервер?
- 13) Какие известные Вам сервисы сети Интернет позволяют организовать доступ к ресурсам посредством SSH ключей?

1. Основные цели и задачи ПО удаленного доступа:

- Управление и администрирование удаленных серверов и сетевого оборудования.

- Предоставление доступа к ресурсам и данным из удаленных мест.
- Обеспечение безопасности и защиты данных при удаленном обмене информацией.

2. TELNET vs SSH:

- **TELNET:**

- Не обеспечивает шифрование данных.
- Использует нешифрованный текстовый протокол.
- Уязвим к перехвату и атакам MITM.

- **SSH:**

- Обеспечивает шифрование данных.
- Использует зашифрованный протокол.
- Предоставляет более высокий уровень безопасности.

3. Способы установления соединения по SSH:

- **Парольный метод:**

- *Положительные аспекты:* Прост в использовании.
- *Отрицательные аспекты:* Уязвим к атакам перебора паролей.

- **Метод с использованием ключей:**

- *Положительные аспекты:* Высокий уровень безопасности, отсутствие необходимости ввода пароля.
- *Отрицательные аспекты:* Требуется предварительного обмена открытыми ключами.

4. Пример использования системы удаленного доступа:

- Работа в VirtualBox очень неудобная, нельзя создавать несколько терминалов, но подключаясь к серверу по ssh в командной строке Windows я смог убрать все неудобства .

5. Распространенные сетевые службы на основе SSH:

- **SSH для удаленного администрирования.**
- **SCP (Secure Copy) для безопасной передачи файлов.**
- **SFTP (SSH File Transfer Protocol) для безопасной передачи файлов.**
- **SSH tunneling для безопасной передачи данных между узлами сети.**

Пример использования службы передачи файлов по безопасному туннелю:

- Пользователь может использовать SCP для копирования файлов между локальной машиной и удаленным сервером через зашифрованный туннель SSH.

6. Ключ SSH:

- **Определение:** Ключ SSH – это пара ключей, приватный и публичный, используемых для аутентификации в процессе SSH-соединения.
- **Преимущества использования ключей:**
 - **Безопасность:** Исключение необходимости ввода пароля.
 - **Аутентификация:** Более надежный метод подтверждения личности пользователя.
 - **Удобство:** Без необходимости запоминания или ввода пароля.

7. Генерация ключей SSH в разных ОС:

- **Linux/macOS:**
 - Откройте терминал и используйте **ssh-keygen**:
- **Windows (через Git Bash):**
 - Запустите Git Bash и используйте тот же синтаксис, что и в Linux/macOS.

8. Генерация ключей:

- Да, из секретного ключа можно сгенерировать публичный, но не наоборот. Публичный ключ вычисляется из приватного.

9. Отличие пар ключей:

- Да, пары ключей будут отличаться при генерации с разными параметрами (например, с паролем или без).

10. Типы ключей для ssh-keygen.exe:

o - Заставляет ssh-keygen сохранять закрытые ключи, используя новый формат OpenSSH, а не более совместимый формат PEM.

t - Указывает тип ключа для создания. Возможными значениями являются `rsa1` для версии протокола 1 и `dsa`, `ecdsa`, `ed25519` или `rsa` для версии протокола 2.

v - Подробный режим. Заставляет ssh-keygen печатать сообщения об отладке о ее ходе. Это полезно для генерации модулей отладки.

y - Эта опция считывает закрытый файл формата OpenSSH и печатает открытый ключ OpenSSH в стандартный вывод.

r - Запрашивает изменение ключевой фразы файла закрытого ключа вместо создания нового закрытого ключа. 16

e - Эта опция будет считывать закрытый или общедоступный файл ключа OpenSSH и распечатывать для стандартного вывода ключ в одном из форматов, указанных параметром -m.

i - Этот параметр будет считывать незашифрованный файл закрытого (или открытого) ключа в формате, указанном -m выберите и распечатайте совместимый с OpenSSH закрытый (или открытый) ключ в стандартный вывод

11. Использование ключа на разных ОС/ПК:

- Да, можно использовать один секретный ключ на разных операционных системах и компьютерах.

12. Подключение по ключу в OpenSSH на Windows:

- Да, можно организовать подключение по ключу. Убедитесь, что OpenSSH сервер установлен и запущен на Windows, а в конфигурационном файле указан путь к вашему публичному ключу.

13. Сервисы с SSH-ключами:

- **GitHub:** Использует SSH-ключи для аутентификации при клонировании и пушах.
- **Bitbucket:** Также предоставляет возможность использовать SSH-ключи для доступа к репозиториям.
- **GitLab:** Поддерживает аутентификацию по SSH-ключам.
- **DigitalOcean, AWS, и другие облачные провайдеры:** Часто используются SSH-ключи для безопасного доступа к виртуальным машинам.