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 _____ Красиков И. А.

(подпись, дата)

Руководитель

_____ Кургасов В.В.

(подпись, дата)

Липецк 2023

Оглавление

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 5
5. Вывод информации об удаленной системе 6
6. Создание файла и передача его на удаленную систему по шифровальному каналу 6
7. Формирование зашифрованного ключа 7
8. Передача зашифрованного ключа на удаленную систему 7
9. Попытка подключения по ssh к удаленной системе 8
10. Повторная передача файла 8
11. Содержимое файлов telnet.log и ssh.log 9
Ответы на контрольные вопросы 10

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

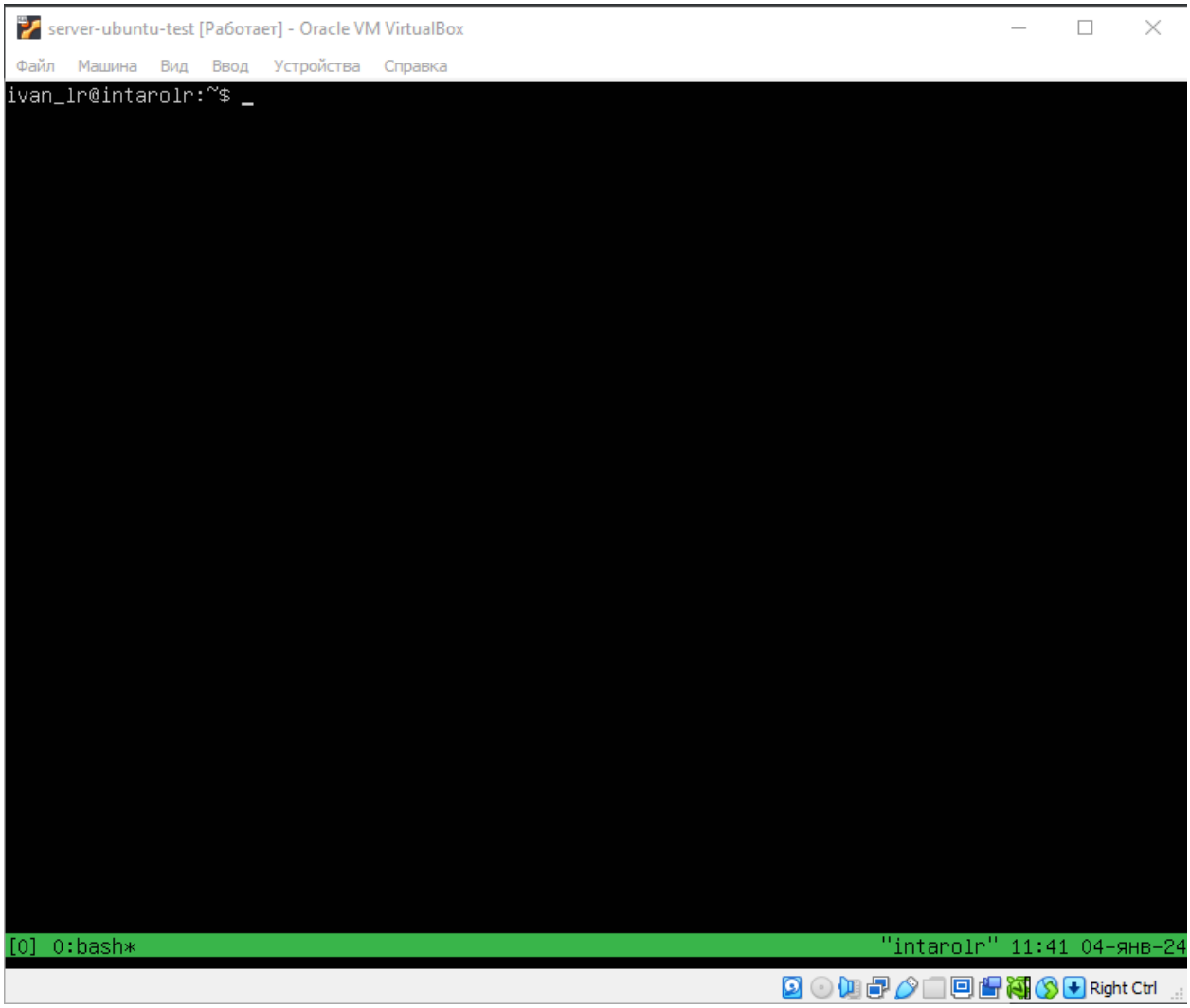

Рисунок 1 – Запуск tmux

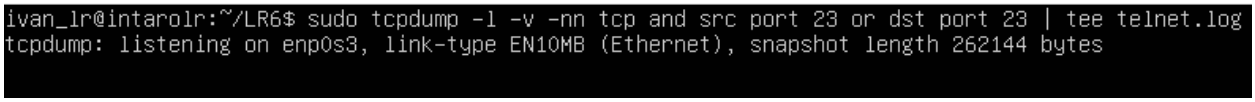

Рисунок 2 – Запуск tcpdump по порту 23

2. Установка соединения с удаленной системой по порту 23

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

Рисунок 3 – Подключение к удаленной системе

Получаем ошибку подключения.

3. Запуск tcpdump на порту 22

Рисунок 4 – Запуск tcpdump на порту 22

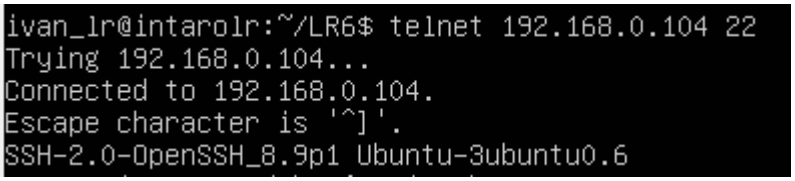

Рисунок 5 – подключение к удаленной системе по порту 22 через telnet

4. Подключение по ssh к удаленной системе

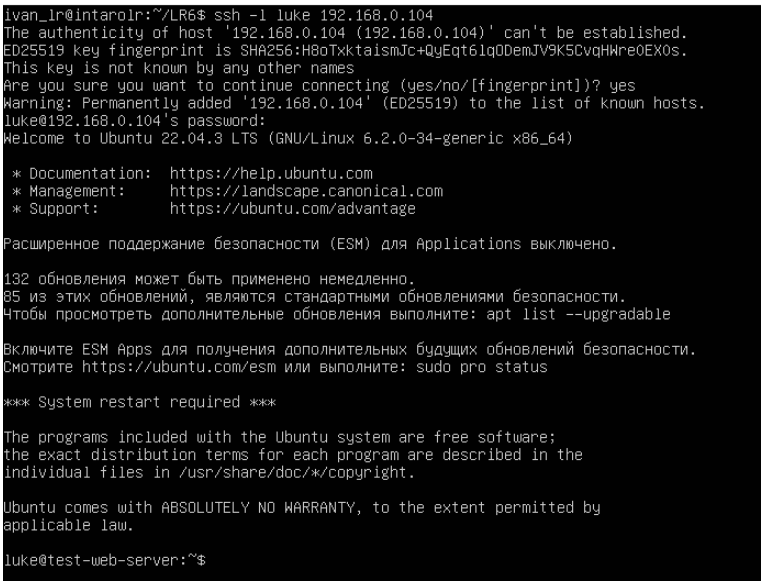

Рисунок 6 – Подключение по ssh

5. Вывод информации об удаленной системе

```
luke@test-web-server:~$ uname -a
Linux test-web-server 6.2.0-34-generic #34~22.04.1-Ubuntu SMP PREEMPT_DYNAMIC Thu Sep  7 13:12:03 UT
C 2 x86_64 x86_64 x86_64 GNU/Linux
luke@test-web-server:~$
```

Рисунок 7 – uname -a

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

```
Красиков Иван Александрович
Лабораторная работа №6_
~
~
~
~
~
```

Рисунок 8 – Содержимое файла lr6

```
ivan_lr@intarolr:~/LR6$ scp lr6 luke@192.168.0.104:/home/luke
luke@192.168.0.104's password:
lr6                                           100%   96    40.1KB/s   00:00
ivan_lr@intarolr:~/LR6$
```

Рисунок 9 – Передача файла

```
/home/luke/lr6                                96/96                    100%
Красиков Иван Александрович
Лабораторная работа №6

1Помощь 2Раз~рн 3Выход 4Hex  5Пер~ти 6       7Поиск 8Исх~ый 9Формат10Выход
```

Рисунок 10 – Проверка наличия и содержимого файла на удаленной системе

7. Формирование зашифрованного ключа

```
ivan_lr@intarolr:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ivan_lr/.ssh/id_rsa): /home/ivan_lr/.ssh/key
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ivan_lr/.ssh/key
Your public key has been saved in /home/ivan_lr/.ssh/key.pub
The key fingerprint is:
SHA256:g6NuKgmulyHpRWDj38/FXflbjMUqPqte37rIHbwTPBw ivan_lr@intarolr
The key's randomart image is:
+---[RSA 3072]----+
|                 |
| +               |
|o o           .. |
| . .   .     oE o|
| .o . o S . .o.* |
|+ .o o . + ...B.o|
|+o.o. o .  ...ooo|
|ooo..  o   oo+.= |
|oo.o.    .o.+o*+.|
+----[SHA256]-----+
ivan_lr@intarolr:~$ _
```

Рисунок 11 – формирование зашифрованного ключа

8. Передача зашифрованного ключа на удаленную систему

```
ivan_lr@intarolr:~$ scp /home/ivan_lr/.ssh/key.pub luke@192.168.0.104:/home/luke/.ssh/autorized_keys
luke@192.168.0.104's password:
key.pub                                       100%  570   338.4KB/s   00:00
ivan_lr@intarolr:~$ _
```

Рисунок 12 – Передача ключа на удаленную систему

```
/home/luke/.ssh/autorized_keys                570/570              100%
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCy/gJkIQqJXZsifWsgvNiTbAgk29YtDGghCvqLQ9S6
plAiUvjzwoQiaxkRyfDAw64EYqU8sDS2YKqo2f2wiclm80R16GJuTBxIqr4UI5gvKSldAvfejxps9ZQQ
Si/OG2jzXMYuIm9U9Y45O/X2RU559V3E62h9gCdwF6mUZIX9Lo+YKTjS8J2oYy8JiskhSpOPKElOSdJn
Hq15HjcTV+rpWzlaF7+AbimJk2o+zKoD7ZYd6S/N5fZE8AQnC88ogAh+i0IOg4BVleGSq06guoaNXT2n
NVeIVlCqNdlH+ajgV3gZxwiBPkxhtsmWXaS4/2NIkhKcoKT40G3CA3w2i3dk4sHtXmuRgxR/oZDVnAkF
X3+jKH9fx3rTT/iGDzazjO3novX4VbwZiS1BKwp7ynNqQ7iqVe+Ufe/4lKWBFYCNAlsY6WcYpYDMWSgn
irJlVVXrifsOhpnsd/hMV07sAzZ76VL2wJ07bvxH8Jx8//uzVKAn98ZUgAsBsfG0/1DiLDE= ivan_lr
@intarolr
```

Рисунок 13 – Проверка передачи ключа

9. Попытка подключения по ssh к удаленной системе

При подключение указываем шифрованный ключ

```
ivan_lr@intarolr:~/.ssh$ ssh -i key luke@192.168.0.104
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-34-generic x86_64)

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

Расширенное поддержание безопасности (ESM) для Applications выключено.

132 обновления может быть применено немедленно.
85 из этих обновлений, являются стандартными обновлениями безопасности.
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

*** System restart required ***
Last login: Thu Jan  4 17:44:33 2024 from 192.168.0.102
luke@test-web-server:~$ _
```

Рисунок 14 – Подключение по ssh

При подключение по ssh с указание ключа, пароль не требуется.

10. Повторная передача файла

```
ivan_lr@intarolr:~$ scp -i .ssh/key LR6/lr6 luke@192.168.0.104:/home/luke
lr6                                           100%   96    48.3KB/s   00:00
ivan_lr@intarolr:~$
```

Рисунок 15 – Передача файла с указание шифрованного ключа

```
/home/luke/lr6                                96/96                  100%
Красиков Иван Александрович
Лабораторная работа №6

1Помощь 2Раз~рн 3Выход 4Hex  5Пер~ти 6       7Поиск 8Исх~ый 9Формат10Выход
```

Рисунок 16 – Проверка передачи файла

При передачи файла с указанием ключа, не требуется пароль.

11. Содержимое файлов telnet.log и ssh.log

```
    192.168.0.102.49552 > 192.168.0.104.22: Flags [P.], cksum 0x82c9 (incorrect -> 0x2dc1), seq 3526
:3658, ack 3374, win 501, options [nop,nop,TS val 1211683787 ecr 1710156105], length 132
14:56:43.023216 IP (tos 0x8, ttl 64, id 15487, offset 0, flags [DF], proto TCP (6), length 88)
    192.168.0.104.22 > 192.168.0.102.49552: Flags [P.], cksum 0xb3f0 (correct), seq 3374:3410, ack 3
658, win 501, options [nop,nop,TS val 1710156107 ecr 1211683787], length 36
14:56:43.023779 IP (tos 0x8, ttl 64, id 21529, offset 0, flags [DF], proto TCP (6), length 88)
    192.168.0.102.49552 > 192.168.0.104.22: Flags [P.], cksum 0x8269 (incorrect -> 0x8ab9), seq 3658
:3694, ack 3410, win 501, options [nop,nop,TS val 1211683789 ecr 1710156107], length 36
14:56:43.028360 IP (tos 0x8, ttl 64, id 15488, offset 0, flags [DF], proto TCP (6), length 88)
    192.168.0.104.22 > 192.168.0.102.49552: Flags [P.], cksum 0x1184 (correct), seq 3410:3446, ack 3
694, win 501, options [nop,nop,TS val 1710156112 ecr 1211683789], length 36
14:56:43.030064 IP (tos 0x8, ttl 64, id 15489, offset 0, flags [DF], proto TCP (6), length 140)
    192.168.0.104.22 > 192.168.0.102.49552: Flags [P.], cksum 0x870f (correct), seq 3446:3534, ack 3
694, win 501, options [nop,nop,TS val 1710156114 ecr 1211683789], length 88
14:56:43.030256 IP (tos 0x8, ttl 64, id 21530, offset 0, flags [DF], proto TCP (6), length 52)
    192.168.0.102.49552 > 192.168.0.104.22: Flags [.], cksum 0x8245 (incorrect -> 0x2f7c), ack 3534,
 win 501, options [nop,nop,TS val 1211683796 ecr 1710156112], length 0
14:56:43.030424 IP (tos 0x8, ttl 64, id 21531, offset 0, flags [DF], proto TCP (6), length 88)
    192.168.0.102.49552 > 192.168.0.104.22: Flags [P.], cksum 0x8269 (incorrect -> 0x6385), seq 3694
:3730, ack 3534, win 501, options [nop,nop,TS val 1211683796 ecr 1710156112], length 36
14:56:43.030562 IP (tos 0x8, ttl 64, id 21532, offset 0, flags [DF], proto TCP (6), length 112)
    192.168.0.102.49552 > 192.168.0.104.22: Flags [P.], cksum 0x8281 (incorrect -> 0xce05), seq 3730
:3790, ack 3534, win 501, options [nop,nop,TS val 1211683796 ecr 1710156112], length 60
14:56:43.030694 IP (tos 0x8, ttl 64, id 21533, offset 0, flags [DF], proto TCP (6), length 52)
    192.168.0.102.49552 > 192.168.0.104.22: Flags [F.], cksum 0x8245 (incorrect -> 0x2f1b), seq 3790
, ack 3534, win 501, options [nop,nop,TS val 1211683796 ecr 1710156112], length 0
14:56:43.032495 IP (tos 0x8, ttl 64, id 15490, offset 0, flags [DF], proto TCP (6), length 52)
    192.168.0.104.22 > 192.168.0.102.49552: Flags [.], cksum 0x2f17 (correct), ack 3791, win 501, op
tions [nop,nop,TS val 1710156116 ecr 1211683796], length 0
14:56:43.044979 IP (tos 0x8, ttl 64, id 15491, offset 0, flags [DF], proto TCP (6), length 52)
    192.168.0.104.22 > 192.168.0.102.49552: Flags [F.], cksum 0x2f09 (correct), seq 3534, ack 3791,
win 501, options [nop,nop,TS val 1710156129 ecr 1211683796], length 0
14:56:43.045029 IP (tos 0x8, ttl 64, id 0, offset 0, flags [DF], proto TCP (6), length 52)
    192.168.0.102.49552 > 192.168.0.104.22: Flags [.], cksum 0x2efb (correct), ack 3535, win 501, op
tions [nop,nop,TS val 1211683810 ecr 1710156129], length 0
```

Рисунок 17 – ssh.log

```
n 502, options [nop,nop,TS val 1211959124 ecr 1710431442], length 0
15:01:18.373444 IP (tos 0x0, ttl 64, id 15535, offset 0, flags [DF], proto TCP (6), length 93)
    192.168.0.104.22 > 192.168.0.102.47728: Flags [P.], cksum 0xb350 (correct), seq 1:42, ack 1, win
 510, options [nop,nop,TS val 1710431456 ecr 1211959124], length 41: SSH: SSH-2.0-OpenSSH_8.9p1 Ubun
tu-3ubuntu0.6
15:01:18.373495 IP (tos 0x10, ttl 64, id 48847, offset 0, flags [DF], proto TCP (6), length 52)
    192.168.0.102.47728 > 192.168.0.104.22: Flags [.], cksum 0x8245 (incorrect -> 0xe078), ack 42, w
in 502, options [nop,nop,TS val 1211959139 ecr 1710431456], length 0
15:01:43.962360 IP (tos 0x10, ttl 64, id 48848, offset 0, flags [DF], proto TCP (6), length 57)
    192.168.0.102.47728 > 192.168.0.104.22: Flags [P.], cksum 0x824a (incorrect -> 0x7683), seq 1:6,
 ack 42, win 502, options [nop,nop,TS val 1211984728 ecr 1710431456], length 5
15:01:43.963310 IP (tos 0x0, ttl 64, id 15536, offset 0, flags [DF], proto TCP (6), length 52)
    192.168.0.104.22 > 192.168.0.102.47728: Flags [.], cksum 0x187f (correct), ack 6, win 510, optio
ns [nop,nop,TS val 1710457047 ecr 1211984728], length 0
15:01:44.562798 IP (tos 0x10, ttl 64, id 48849, offset 0, flags [DF], proto TCP (6), length 54)
    192.168.0.102.47728 > 192.168.0.104.22: Flags [P.], cksum 0x8247 (incorrect -> 0x091b), seq 6:8,
 ack 42, win 502, options [nop,nop,TS val 1211985328 ecr 1710457047], length 2
15:01:44.563586 IP (tos 0x0, ttl 64, id 15537, offset 0, flags [DF], proto TCP (6), length 52)
    192.168.0.104.22 > 192.168.0.102.47728: Flags [.], cksum 0x13cd (correct), ack 8, win 510, optio
ns [nop,nop,TS val 1710457647 ecr 1211985328], length 0
15:01:44.564801 IP (tos 0x0, ttl 64, id 15538, offset 0, flags [DF], proto TCP (6), length 86)
    192.168.0.104.22 > 192.168.0.102.47728: Flags [P.], cksum 0x9173 (correct), seq 42:76, ack 8, wi
n 510, options [nop,nop,TS val 1710457648 ecr 1211985328], length 34
15:01:44.564829 IP (tos 0x10, ttl 64, id 48850, offset 0, flags [DF], proto TCP (6), length 52)
    192.168.0.102.47728 > 192.168.0.104.22: Flags [.], cksum 0x8245 (incorrect -> 0x13b0), ack 76, w
in 502, options [nop,nop,TS val 1211985330 ecr 1710457648], length 0
15:01:44.565803 IP (tos 0x0, ttl 64, id 15539, offset 0, flags [DF], proto TCP (6), length 54)
    192.168.0.104.22 > 192.168.0.102.47728: Flags [FP.], cksum 0x0694 (correct), seq 76:78, ack 8, w
in 510, options [nop,nop,TS val 1710457649 ecr 1211985328], length 2
15:01:44.565927 IP (tos 0x10, ttl 64, id 48851, offset 0, flags [DF], proto TCP (6), length 52)
    192.168.0.102.47728 > 192.168.0.104.22: Flags [F.], cksum 0x8245 (incorrect -> 0x13aa), seq 8, a
ck 79, win 502, options [nop,nop,TS val 1211985331 ecr 1710457649], length 0
15:01:44.566507 IP (tos 0x0, ttl 64, id 15540, offset 0, flags [DF], proto TCP (6), length 52)
    192.168.0.104.22 > 192.168.0.102.47728: Flags [.], cksum 0x13a1 (correct), ack 9, win 510, optio
ns [nop,nop,TS val 1710457650 ecr 1211985331], length 0
```

Рисунок 18 – telnet.log

Ответы на контрольные вопросы

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

Программное обеспечение удаленного доступа предназначено для обеспечения возможности удаленного управления компьютерами и сетевыми ресурсами. Его основные цели включают удаленное администрирование, предоставление технической поддержки, доступ к файлам, обеспечение безопасности и мониторинг, а также поддержку дистанционной работы, что обеспечивает гибкость и эффективность в управлении информационными ресурсами из любой точки мира.

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

TELNET и SSH - это протоколы удаленного доступа, но SSH обладает рядом преимуществ и отличительных особенностей по сравнению с устаревшим TELNET:

Безопасность:

TELNET: Передает данные в открытом виде, что делает его небезопасным для передачи чувствительной информации.

SSH: Обеспечивает шифрование данных, обмен ключами и аутентификацию, обеспечивая высокий уровень безопасности при удаленном доступе.

Шифрование:

TELNET: Не предоставляет механизмы шифрования, что делает передаваемую информацию уязвимой для перехвата.

SSH: Использует криптографию для шифрования данных, предоставляя защиту от прослушивания и несанкционированного доступа.

Аутентификация:

TELNET: Обычно использует пароли для аутентификации, что менее безопасно и подвержено угрозам перехвата.

SSH: Предоставляет более сильные методы аутентификации, такие как использование ключей или двухфакторной аутентификации.

Управление сеансами:

TELNET: Просто передает текстовые команды, не предоставляя средства для эффективного управления сеансами и сетевыми ресурсами.

SSH: Позволяет более гибкое управление сеансами, поддерживает туннелирование, сжатие данных и перенаправление портов.

Порт подключения:

TELNET: Использует порт 23 для подключения, что может быть уязвимым к атакам и перехвату.

SSH: Обычно использует порт 22, но может быть настроен для использования другого порта, улучшая безопасность.

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

Пароль:

Положительные аспекты:

Прост в использовании и управлении.

Подходит для случаев, когда невозможно или неудобно использовать ключи.

Отрицательные аспекты:

Менее безопасен, так как подвержен атакам перебора паролей.

Возможны проблемы с безопасностью при передаче пароля по сети.

Открытый ключ:

Положительные аспекты:

Гораздо более безопасен, так как использует криптографию с открытым ключом.

Удобен для автоматизации процесса входа в систему.

Отрицательные аспекты:

Может потребовать дополнительных шагов для настройки, особенно для новых пользователей.

Возможны проблемы, если ключ утрачен или скомпрометирован.

Аутентификация через агент:

Положительные аспекты:

Повышает удобство использования открытых ключей, предотвращая необходимость ввода пароля каждый раз.

Улучшает безопасность, так как ключи не хранятся на удаленном сервере.

Отрицательные аспекты:

Требует запуска агента аутентификации на локальной машине, что может быть неудобно для некоторых пользователей.

Неправильная настройка может сделать систему менее безопасной.

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

Выполняя почти все лабораторные работы, я использовал ssh для удаленного доступа из командной строки Windows к ВМ ubuntu-server

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

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

6) Что такое ключ ssh? В чем преимущество их использования?

Ключ SSH (Secure Shell) - это пара криптографических ключей, состоящая из открытого и закрытого ключа, используемая для аутентификации при подключении к удаленному серверу через протокол SSH. Преимущество использования ключей SSH заключается в усилении безопасности, поскольку они предоставляют более сильные механизмы аутентификации по сравнению с паролями. Кроме того, они обеспечивают удобство автоматизации процесса входа в систему и поддерживают безопасные методы передачи файлов и проксирования трафика.

7) Как сгенерировать ключи ssh в разных ОС?

Для работы с SSH-ключами используются утилиты, входящие в оболочку OpenSSH. Они работают под Linux, Windows и MacOS. Для Linux и Windows для генерации ключа отработает команда ssh-keygen. Еще ей можно указать параметр –t и после указать тип создаваемого ключа (алгоритм шифрования определяет тип).

8) Возможно ли из «секретного» ключа сгенерировать «публичный» и/или наоборот?

Нет, из секретного ключа нельзя восстановить публичный ключ, и наоборот. Секретный ключ используется для расшифровки данных или подписи, в то время как публичный ключ используется для проверки этих подписей или зашифрованных данных. Важным свойством криптографических алгоритмов, используемых в SSH, является то, что они являются односторонними функциями, и восстановление закрытого ключа из открытого или наоборот математически невозможно при текущем уровне знаний.

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

Пары ключей, сгенерированные на одном ПК при одинаковых исходных условиях (даже с одинаковыми параметрами, такими как отсутствие пароля на секретном ключе), обычно будут отличаться. Это связано с использованием случайных чисел при генерации ключей, что приводит к уникальным значениям даже при повторных попытках на том же компьютере. Тем не менее, степень уникальности зависит от качества генератора случайных чисел и может быть повлияна конкретными настройками генерации ключей.

10) Перечислите доступные ключи 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.

p - 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) Можно ли использовать один «секретный» ключ доступа с разных ОС, установленных на одном ПК/на разных ПК?

Можно. Считается, что ssh-ключи не изнашиваются, поэтому их можно не менять и использовать для доступа на разные удаленные системы. Но лучше так не делать, потому что это снижает безопасность – получив один раз закрытый ключ, мы сможем получить доступ ко всем удаленным системам, которые используют пару публичного ключа этого ssh-ключа.

12) Возможно ли организовать подключение «по ключу» ssh к системе с ОС Windows, в которой запущен OpenSSH сервер?

Конечно, ведь поддержку OpenSSH для этого и ввели. Подключаться к винде и без ssh и OpenSSH можно было, но это было трудным занятием, а соединение не всегда стабильным.

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

GitHub поддерживает авторизацию по ssh, для этого нужно создать секрет в аккаунте github и сохранить в секрете публичный ключ, а потом с машины постучаться на github – выполнить клон репозитория, пулл в него или что-то похожее.