

**Липецкий государственный технический университет**

**Факультет автоматизации и информатики**

**Кафедра автоматизированных систем управления**

**Лабораторная работа № 7**

**по OS Linux**

**Работа с SSH**

Студент

Комаричев А. В.

Группа АИ-19

Руководитель

Кургасов В. В.

Липецк 2021г.

Содержание	
Цель работы .....	3
Ход работы.....	4
1. Запустить терминал с командной оболочкой ОС и ввести команду <code>tmux</code> (терминальный мультиплексор). .....	4
2. Соединение с удаленным сервером .....	5
3. Анализатор сетевого трафика.....	5
4. Установить шифрованное соединение с удаленным сервером. ....	6
5. Выполнить команду <code>uname -a</code> , выведя информацию об удаленной системе .....	6
6. Передать файл по шифрованному каналу на удаленную систему .....	7
7. Формирование зашифрованных ключей .....	8
8. Переслать публичный ключ на удаленный узел .....	9
9. Выполнить подключение к удаленной системе .....	10
10. Произвести повторную передачу текстового файла на удаленный узел. 11	
11. Просмотреть содержимое файлов <code>ssh.log</code> , <code>telnet.log</code> . ....	12
Вывод.....	13
Контрольные вопросы .....	14

### Цель работы

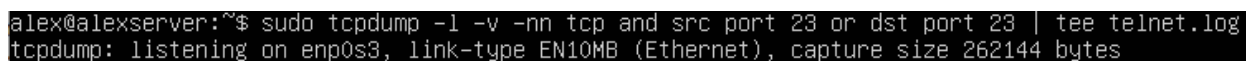
Лабораторная работа предназначена для целей практического ознакомления с программным обеспечением удаленного доступа к распределенным системам обработки данных.

## Ход работы

1. Запустить терминал с командной оболочкой ОС и ввести команду `tmux` (терминальный мультиплексор).

Комбинациями клавиш `Ctrl-b+c` 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;`



```
alex@alexserver:~$ sudo tcpdump -l -v -nn tcp and src port 23 or dst port 23 | tee telnet.log
tcpdump: listening on enp0s3, link-type EN10MB (Ethernet), capture size 262144 bytes
```

Рисунок 1 – Анализатор трафика `tcpdump` (порт 23)

## 2. Соединение с удаленным сервером

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

Для авторизации следует использовать логин `student`;

Чтобы переключаться между окнами можно использовать `Ctrl-b+(1-9)`.

```
alex@alexserver:~$ telnet 178.234.29.197 23
Trying 178.234.29.197...
telnet: Unable to connect to remote host: Connection timed out
```

Рисунок 2 – Соединение с удаленным сервером по протоколу telnet

Теперь сделаем все то же самое с портом 22.

```
alex@alexserver:~$ sudo tcpdump -l -v -nn tcp and src port 22 or dst port 22 | tee telnet.log
tcpdump: listening on enp0s3, link-type EN10MB (Ethernet), capture size 262144 bytes
```

Рисунок 3 – Анализатор трафика tcpdump (порт 22)

```
alex@alexserver:~$ telnet 178.234.29.197 22
Trying 178.234.29.197...
Connected to 178.234.29.197.
Escape character is '^]'.
SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubuntu2.10
_
```

Рисунок 4 – Соединение с удаленным сервером по протоколу telnet

Я смог подключиться к удаленному серверу.

## 3. Анализатор сетевого трафика

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 используем команду:

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

```
alex@alexserver:~$ sudo tcpdump -l -v -nn tcp and src port 22 or dst port 22 | tee ssh.log
[sudo] password for alex:
tcpdump: listening on enp0s3, link-type EN10MB (Ethernet), capture size 262144 bytes
```

Рисунок 5 – Анализатор трафика tcpdump

#### 4. Установить шифрованное соединение с удаленным сервером.

Таблица 1. Данные для удаленного подключения

IP	178.234.29.197
Порт	22
Логин	stud6
Пароль	n7fGhy81Tm

```
alex@alexserver:~$ ssh -l stud6 178.234.29.197
stud6@178.234.29.197's password:
Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 4.4.0-210-generic x86_64)

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

22 packages can be updated.
5 updates are security updates.

New release '18.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Tue Jan 25 16:19:02 2022 from 176.212.155.108
stud6@kurgasov:~$ _
```

Рисунок 6 – Шифрованное соединение с удаленным сервером

#### 5. Выполнить команду `uname -a`, выведя информацию об удаленной системе

```
stud6@kurgasov:~$ uname -a
Linux kurgasov.ru 4.4.0-210-generic #242-Ubuntu SMP Fri Apr 16 09:57:56 UTC 2021 x86_64 x86_64 x86_64 GNU/Linux
```

Рисунок 7 – Информация об удаленной системе

## 6. Передать файл по зашифрованному каналу на удаленную систему

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.

Создадим новое окно в `tmux` (`Ctrl-b+c`).

```
alex@alexserver:~$ nano text.txt
alex@alexserver:~$ cat text.txt
Komarichev Alexandr Vital'evich, laboratornaya 7
```

Рисунок 8 – Текстовый файл

```
alex@alexserver:~$ scp ~/text.txt stud6@kurgasov.ru:/home/stud6/
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' (ECDSA) to the list of known hosts.
stud6@kurgasov.ru's password:
text.txt                                100% 49    0.9KB/s   00:00
alex@alexserver:~$ _
```

Рисунок 9 – Передача файла на удаленную систему по зашифрованному каналу  
Переключимся на удаленную систему, чтобы проверить наличие файла.

Левая панель				Файл	Команда	Настройки	Правая панель			
< ~						. [^]>	< ~			
				Имя	Размер	Время	Имя			
					-ВВЕРХ-	январь 8 2021				
				./..	4096	декабрь 3 2019				
				./.cache	4096	декабрь 3 2019				
				./.config	4096	декабрь 3 2019				
				./.local	4096	декабрь 3 2019				
				./.nano	4096	декабрь 3 2019				
				./.ssh	4096	январь 16 2021				
				./conf	4096	декабрь 2 2019				
				./mail	4096	декабрь 2 2019				
				./tmp	4096	декабрь 2 2019				
				./web	4096	декабрь 2 2019				
				./bash_history	30	январь 25 16:21				
				./bash_logout	220	сентябрь 1 2015				
				./bashrc	3771	сентябрь 1 2015				
				./profile	655	июнь 24 2016				
				./ssh.log.swp	1024	январь 16 2021				
				text.txt	49	январь 25 18:42				

```
/home/stud6/text.txt  
Komarichev Alexandr Vital'evich, laboratornaya 7
```

Рисунок 11 – Содержимое файла

## 7. Формирование зашифрованных ключей

Отключиться от удаленного узла (команда exit) и сформировать зашифрованные ключи на локальном хосте.

Чтобы сформировать ключи используем команду ssh-keygen.

```
stud6@kurgasov:~$ exit  
ВЫХОД  
Connection to 178.234.29.197 closed.  
alex@alexserver:~$ _
```

Рисунок 12 – Выход

```
alex@alexserver:~$ ssh-keygen  
Generating public/private rsa key pair.  
Enter file in which to save the key (/home/alex/.ssh/id_rsa): /home/alex/.ssh/id_rsa  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /home/alex/.ssh/id_rsa  
Your public key has been saved in /home/alex/.ssh/id_rsa.pub  
The key fingerprint is:  
SHA256:tEw88TqTCLw1/DIv9wSsSL63s5pdgefj8NQ8Jn0oSzI alex@alexserver  
The key's randomart image is:  
+---[RSA 3072]---+  
|                 |  
|  . . . 0        |  
| 0 + = .        |  
|  + X =         |  
| 0 = S          |  
| 0 . B 0        |  
|  E = @ B       |  
|  BoX 0 .       |  
|  +0*=0 .       |  
|                 |  
+---[SHA256]-----+
```

Рисунок 13 – Генерация ssh ключа



## 8. Переслать публичный ключ на удаленный узел

Используя команду `scp` с указанием расположения файла (публичного ключа) на локальной системе (`/home/alex/.ssh/id_rsa.pub`), произвести его передачу по зашифрованному туннелю на удаленный узел в заданный каталог `/home/stud6/.ssh/` под именем `authorized_keys`.

```
alex@alexserver:~$ scp ~/.ssh/id_rsa.pub stud6@kurgasov.ru:/home/stud6/.ssh/authorized_keys
stud6@kurgasov.ru's password:
id_rsa.pub                                100% 569      12.3KB/s   00:00
alex@alexserver:~$
```

Рисунок 14 – Передача публичного ключа

## 9. Выполнить подключение к удаленной системе

Воспользовавшись командой `ssh -l stud6 178.234.29.197`, снова сделать попытку подключения к удаленной системе.

В этот раз вводить пароль не потребовалось.

```
alex@alexserver:~$ ssh -l stud6 178.234.29.197
Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 4.4.0-210-generic x86_64)

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

22 packages can be updated.
5 updates are security updates.

New release '18.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Tue Jan 25 18:27:15 2022 from 176.212.155.108
stud6@kurgasov:~$
```

Рисунок 15 – Подключение к удаленной системе

10. Произвести повторную передачу текстового файла на удаленный узел.

Убедиться в наличии переданной копии файла на удаленном хосте. Отметить отличия в процедуре передачи файла;

```
alex@alexserver:~$ scp ~/text.txt stud6@kurgasov.ru:/home/stud6/text2.txt
text.txt                                100% 49      0.8KB/s  00:00
```

Рисунок 16 – Передача файла на удаленный узел

Вводить пароль не потребовалось.

Левая панель				Правая панель			
Файл				Файл			
Команда				Команда			
Настройки				Настройки			
Правая панель				Правая панель			
. [^]>				. [^]>			
Имя	Размер	Время	правки	Имя	Размер	Время	правки
-ВВЕРХ-		янв 8	2021	-ВВЕРХ-		янв 8	2021
/..	4096	дек 3	2019	/..	4096	дек 3	2019
/.cache	4096	дек 3	2019	/.cache	4096	дек 3	2019
/.config	4096	дек 3	2019	/.config	4096	дек 3	2019
/.local	4096	дек 3	2019	/.local	4096	дек 3	2019
/.nano	4096	дек 3	2019	/.nano	4096	дек 3	2019
/.ssh	4096	янв 16	2021	/.ssh	4096	янв 16	2021
/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_history	50	янв 25	18:48	.bash_history	50	янв 25	18:48
.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
.ssh.log.swp	1024	янв 16	2021	.ssh.log.swp	1024	янв 16	2021
text.txt	49	янв 25	18:42	text.txt	49	янв 25	18:42
text2.txt	49	янв 25	19:04	text2.txt	49	янв 25	19:04
text2.txt				-ВВЕРХ-			
1639G/1818G (90%)				1639G/1818G (90%)			

Рисунок 17 – Проверка наличия файла на удаленном узле

## 11.Просмотреть содержимое файлов ssh.log, telnet.log.

Остановить анализатор сетевых пакетов, воспользовавшись Ctrl-c.

```
GNU nano 4.8 telnet.log
16:53:51.385894 IP (tos 0x10, ttl 64, id 39747, offset 0, flags [DF], proto TCP (6), length 60)
  10.0.2.15.51666 > 178.234.29.197.22: Flags [S], cksum 0xdcec (incorrect -> 0x33dc), seq 1780609>
16:53:51.422553 IP (tos 0x0, ttl 64, id 1043, offset 0, flags [none], proto TCP (6), length 44)
  178.234.29.197.22 > 10.0.2.15.51666: Flags [S.], cksum 0x3af7 (correct), seq 660288001, ack 178>
16:53:51.422579 IP (tos 0x10, ttl 64, id 39748, offset 0, flags [DF], proto TCP (6), length 40)
  10.0.2.15.51666 > 178.234.29.197.22: Flags [.], cksum 0xdcd8 (incorrect -> 0x57c3), ack 1, win >
16:53:51.465773 IP (tos 0x0, ttl 64, id 1044, offset 0, flags [none], proto TCP (6), length 82)
  178.234.29.197.22 > 10.0.2.15.51666: Flags [P.], cksum 0x2c2d (correct), seq 1:43, ack 1, win 6>
16:53:51.465794 IP (tos 0x10, ttl 64, id 39749, offset 0, flags [DF], proto TCP (6), length 40)
  10.0.2.15.51666 > 178.234.29.197.22: Flags [S.], cksum 0xdcd8 (incorrect -> 0x57c3), ack 43, win>
16:55:12.983214 IP (tos 0x10, ttl 64, id 39750, offset 0, flags [DF], proto TCP (6), length 43)
  10.0.2.15.51666 > 178.234.29.197.22: Flags [P.], cksum 0xdcd8 (incorrect -> 0xdcaa), seq 1:4, a>
16:55:12.983432 IP (tos 0x0, ttl 64, id 1045, offset 0, flags [none], proto TCP (6), length 82)
  178.234.29.197.22 > 10.0.2.15.51666: Flags [S.], cksum 0x5287 (correct), ack 4, win 65535, lengt>
16:55:13.019483 IP (tos 0x0, ttl 64, id 1046, offset 0, flags [none], proto TCP (6), length 59)
  178.234.29.197.22 > 10.0.2.15.51666: Flags [P.], cksum 0xd5e1 (correct), seq 43:62, ack 4, win >
16:55:13.019484 IP (tos 0x0, ttl 64, id 1047, offset 0, flags [none], proto TCP (6), length 40)
  178.234.29.197.22 > 10.0.2.15.51666: Flags [F.], cksum 0x5273 (correct), seq 62, ack 4, win 655>
16:55:13.019510 IP (tos 0x10, ttl 64, id 39751, offset 0, flags [DF], proto TCP (6), length 40)
  10.0.2.15.51666 > 178.234.29.197.22: Flags [S.], cksum 0xdcd8 (incorrect -> 0x57c0), ack 62, win>
16:55:13.040673 IP (tos 0x10, ttl 64, id 39752, offset 0, flags [DF], proto TCP (6), length 40)
  10.0.2.15.51666 > 178.234.29.197.22: Flags [F.], cksum 0xdcd8 (incorrect -> 0x57bf), seq 4, ack>
16:55:13.041028 IP (tos 0x0, ttl 64, id 1048, offset 0, flags [none], proto TCP (6), length 40)
  178.234.29.197.22 > 10.0.2.15.51666: Flags [S.], cksum 0x5272 (correct), ack 5, win 65535, lengt>
16:59:42.840935 IP (tos 0x0, ttl 64, id 27773, offset 0, flags [DF], proto TCP (6), length 60)
  10.0.2.15.51668 > 178.234.29.197.22: Flags [S], cksum 0xdcec (incorrect -> 0xa147), seq 2601318>
16:59:42.877317 IP (tos 0x0, ttl 64, id 1051, offset 0, flags [none], proto TCP (6), length 44)
  178.234.29.197.22 > 10.0.2.15.51668: Flags [S.], cksum 0x5493 (correct), seq 672064001, ack 260>
16:59:42.877338 IP (tos 0x0, ttl 64, id 27774, offset 0, flags [DF], proto TCP (6), length 40)
  10.0.2.15.51668 > 178.234.29.197.22: Flags [S.], cksum 0xdcd8 (incorrect -> 0x715f), ack 1, win >
16:59:42.896281 IP (tos 0x0, ttl 64, id 27775, offset 0, flags [DF], proto TCP (6), length 81)
  10.0.2.15.51668 > 178.234.29.197.22: Flags [P.], cksum 0xdcd8 (incorrect -> 0x715f), ack 1, win >
16:59:42.896281 IP (tos 0x0, ttl 64, id 27775, offset 0, flags [DF], proto TCP (6), length 81)
  10.0.2.15.51668 > 178.234.29.197.22: Flags [P.], cksum 0xdcd8 (incorrect -> 0x715f), ack 1, win >
```

Рисунок 18 – Файл telnet.log

```
GNU nano 4.8 ssh.log
16:59:42.840935 IP (tos 0x0, ttl 64, id 27773, offset 0, flags [DF], proto TCP (6), length 60)
  10.0.2.15.51668 > 178.234.29.197.22: Flags [S], cksum 0xdcec (incorrect -> 0xa147), seq 2601318>
16:59:42.877317 IP (tos 0x0, ttl 64, id 1051, offset 0, flags [none], proto TCP (6), length 44)
  178.234.29.197.22 > 10.0.2.15.51668: Flags [S.], cksum 0x5493 (correct), seq 672064001, ack 260>
16:59:42.877338 IP (tos 0x0, ttl 64, id 27774, offset 0, flags [DF], proto TCP (6), length 40)
  10.0.2.15.51668 > 178.234.29.197.22: Flags [S.], cksum 0xdcd8 (incorrect -> 0x715f), ack 1, win >
16:59:42.896281 IP (tos 0x0, ttl 64, id 27775, offset 0, flags [DF], proto TCP (6), length 81)
  10.0.2.15.51668 > 178.234.29.197.22: Flags [P.], cksum 0xdcd8 (incorrect -> 0x715f), ack 1, win >
16:59:42.896281 IP (tos 0x0, ttl 64, id 27775, offset 0, flags [DF], proto TCP (6), length 81)
  10.0.2.15.51668 > 178.234.29.197.22: Flags [P.], cksum 0xdcd8 (incorrect -> 0x715f), ack 1, win >
16:59:42.896457 IP (tos 0x0, ttl 64, id 1052, offset 0, flags [none], proto TCP (6), length 40)
  178.234.29.197.22 > 10.0.2.15.51668: Flags [S.], cksum 0x6c27 (correct), ack 42, win 65535, lengt>
16:59:42.920187 IP (tos 0x0, ttl 64, id 1053, offset 0, flags [none], proto TCP (6), length 82)
  178.234.29.197.22 > 10.0.2.15.51668: Flags [P.], cksum 0x45a0 (correct), seq 1:43, ack 42, win >
16:59:42.920197 IP (tos 0x0, ttl 64, id 27776, offset 0, flags [DF], proto TCP (6), length 40)
  10.0.2.15.51668 > 178.234.29.197.22: Flags [S.], cksum 0xdcd8 (incorrect -> 0x7136), ack 43, win>
16:59:42.920564 IP (tos 0x0, ttl 64, id 27777, offset 0, flags [DF], proto TCP (6), length 1552)
  10.0.2.15.51668 > 178.234.29.197.22: Flags [P.], cksum 0xe2c0 (incorrect -> 0xbdd6), seq 42:155>
16:59:42.920798 IP (tos 0x0, ttl 64, id 1054, offset 0, flags [none], proto TCP (6), length 40)
  178.234.29.197.22 > 10.0.2.15.51668: Flags [S.], cksum 0x6649 (correct), ack 1502, win 65535, le>
16:59:42.920798 IP (tos 0x0, ttl 64, id 1055, offset 0, flags [none], proto TCP (6), length 40)
  178.234.29.197.22 > 10.0.2.15.51668: Flags [S.], cksum 0x6615 (correct), ack 1554, win 65535, le>
16:59:42.956483 IP (tos 0x0, ttl 64, id 1056, offset 0, flags [none], proto TCP (6), length 1016)
  178.234.29.197.22 > 10.0.2.15.51668: Flags [S.], cksum 0xa021 (correct), seq 43:1019, ack 1554,>
16:59:42.956506 IP (tos 0x0, ttl 64, id 27779, offset 0, flags [DF], proto TCP (6), length 40)
  10.0.2.15.51668 > 178.234.29.197.22: Flags [S.], cksum 0xdcd8 (incorrect -> 0x6a74), ack 1019, w>
16:59:42.958267 IP (tos 0x0, ttl 64, id 27780, offset 0, flags [DF], proto TCP (6), length 88)
  10.0.2.15.51668 > 178.234.29.197.22: Flags [P.], cksum 0xdd08 (incorrect -> 0x0a04), seq 1554:1>
16:59:42.958560 IP (tos 0x0, ttl 64, id 1057, offset 0, flags [none], proto TCP (6), length 40)
  178.234.29.197.22 > 10.0.2.15.51668: Flags [S.], cksum 0x6215 (correct), ack 1602, win 65535, le>
16:59:43.049565 IP (tos 0x0, ttl 64, id 1058, offset 0, flags [none], proto TCP (6), length 404)
  178.234.29.197.22 > 10.0.2.15.51668: Flags [P.], cksum 0x8bda (correct), seq 1019:1383, ack 160>
16:59:43.049593 IP (tos 0x0, ttl 64, id 27781, offset 0, flags [DF], proto TCP (6), length 40)
  10.0.2.15.51668 > 178.234.29.197.22: Flags [S.], cksum 0xdcd8 (incorrect -> 0x68d8), ack 1383, w>
```

Рисунок 19 – Файл ssh.log

## Вывод

В лабораторной работе я ознакомился на практике с программным обеспечением удаленного доступа к распределенным системам обработки данных. Также я научился передавать файлы по зашифрованному каналу и подключаться к удаленной системе без использования пароля.

## Контрольные вопросы

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

ПО удаленного доступа – это такое ПО, которое позволяет управлять одним устройством с помощью другого по сети.

Удаленный доступ нужен для:

- Решения технических проблем. Так как всегда присутствовать у ПК лично не получится, удобно подключаться к нему дистанционно.
- Управления сервером. Удаленный доступ используют для управления арендованным сервисом.
- Техподдержка. С удаленным доступом техподдержка может устранять проблемы самостоятельно, без вмешательства владельца ПК.

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

Telnet — это стандартный протокол TCP / IP для службы виртуальных терминалов. Он позволяет вам установить соединение с удаленной системой таким образом, чтобы она отображалась как локальная система. Полная форма TELNET — это Терминальная сеть.

SSH — это сетевой протокол, который широко используется для удаленного доступа и управления устройством. Полная форма SSH — Secure Shell — это основной протокол для доступа к сетевым устройствам и серверам через Интернет.

Ключевые отличия:

- Telnet — это стандартный протокол TCP / IP для службы виртуальных терминалов, а SSH или Secure Shell — это программа для входа на другой компьютер по сети для выполнения команд на удаленном компьютере.

- Telnet уязвим для атак на безопасность, а SSH помогает преодолеть многие проблемы безопасности Telnet.
- Telnet использует порт 23, который был разработан специально для локальных сетей, тогда как SSH по умолчанию работает на порту 22.
- Telnet передает данные в виде простого текста, а в SSH данные отправляются в зашифрованном формате по защищенному каналу.
- Telnet подходит для частных сетей. С другой стороны, SSH подходит для публичных сетей.

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

<b>Конфигурация</b>	<b>Вероятность взлома</b>	<b>Расход ресурсов сервера на обработку запросов</b>
22 порт, авторизация по паролю, без защиты	высокая	высокие
22 порт, авторизация по ключам, без защиты	средняя*	высокие
22 порт, авторизация по ключам, защита на основе ограничения неудачных попыток авторизации	низкая	средние**

Нестандартный порт, авторизация по паролю, без защиты	высокая	низкие
Нестандартный порт, авторизация по ключам, без защиты	средняя*	низкие
Нестандартный порт, авторизация по ключам, защита на основе ограничения неудачных попыток авторизации	низкая	низкие

\* — произвести взлом, если для авторизации используются RSA-ключи, очень сложно, однако неограниченное количество попыток авторизации делает это возможным.

\*\* — количество попыток авторизации ограничено, но серверу всё равно приходится обрабатывать их от большого количества злоумышленников.

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

Удаленный доступ позволяет пользователям подключаться к ресурсам, расположенным в других местах. Инструменты доступа к удаленному рабочему столу позволяют идти еще дальше, позволяя пользователям управлять главным компьютером из любого места через Интернет.

На практике, удаленный доступ позволяет сотрудникам работать из дома, не приходя в офис.



**5) Перечислите распространенные сетевые службы, основанные на использовании шифрованного соединения по протоколу SSH?**

**Перечислите распространенные сетевые службы, основанные на использовании шифрованного соединения по протоколу SSH?**

**Приведите пример использования службы передачи файлов по безопасному туннелю?**

Распространенные сетевые службы на основе SSH: OpenSSH, freeSSHd, dropbear, PuTTY.

Передача файлов по безопасному туннелю может быть актуальна для файлов с паролями.

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

Каждая пара ключей состоит из открытого и закрытого ключа. Закрытый ключ сохраняется на стороне клиента. Открытый ключ используется для шифрования сообщений, которые можно расшифровать закрытым ключом. Открытый ключ загружается на удаленный сервер.

Когда клиент попытается выполнить проверку подлинности через этот ключ, сервер отправит сообщение, зашифрованное с помощью открытого ключа, если клиент сможет его расшифровать и вернуть правильный ответ – аутентификация пройдена.

SSH ключ позволяет входить на удаленный сервер без пароля и повышает безопасность.

**7) Как сгенерировать ключи ssh в разных ОС?**

В Linux генерация ключей выполняется командой ssh-keygen.

Для генерации ключей из Windows раньше использовали программу putty, но она устарела и сейчас более популярен OpenSSH.

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

Нет, не возможно.

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

Да, ключи будут отличаться

**10) Перечислите доступные ключи для ssh-keygen.exe**

-t type ssh-keygen, работает с тремя типами ключей. Возможные значения: RSA 1 - для протокола SSH версии 1. RSA - для протокола SSH версии 2. DSA - для протокола SSH версии 2.

-b Длина ключа в битах. RSA - минимальная длина, 768 бит, длина ключа по-умолчанию, 2048 бит. DSA - длина 1024 бита.

-i Данная опция используется для импорта ключей из одного формата ( например ключи сгенерированные программой PuTTYgen, для Windows ), в формат OpenSSH.

-l Посмотреть отпечаток секретного ключа ( fingerprint ).

-p Изменить секретную фразу приватного ключа.

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

SSH ключ можно использовать для аутентификации разных ПК. Также неважно какая ОС установлена на ПК.

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

Да, возможно.

**13) Какие известные вам сервисы сети интернет позволяют организовать доступ к ресурсам посредством SSH ключей?**  
timeweb, beget, VDSina, Спринтхост.