



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ОБРАЗОВАНИЯ «ЛИПЕЦКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Институт
Кафедра

компьютерных наук
автоматизированных систем управления

ЛАБОРАТОРНАЯ РАБОТА №6
по операционным системам Linux
«Работа с SSH»

Студент

ПИ-22-1

подпись, дата

Пахомов А.А.

Руководитель

подпись, дата

Кургасов В.В.

Липецк, 2024 г.

Цель работы

Практическое ознакомление с программным обеспечением удаленного доступа к распределённым системам обработки данных.

Ход работы

1. Часть I

Настроим для нашей виртуальной машины сетевой мост (рис. 1).

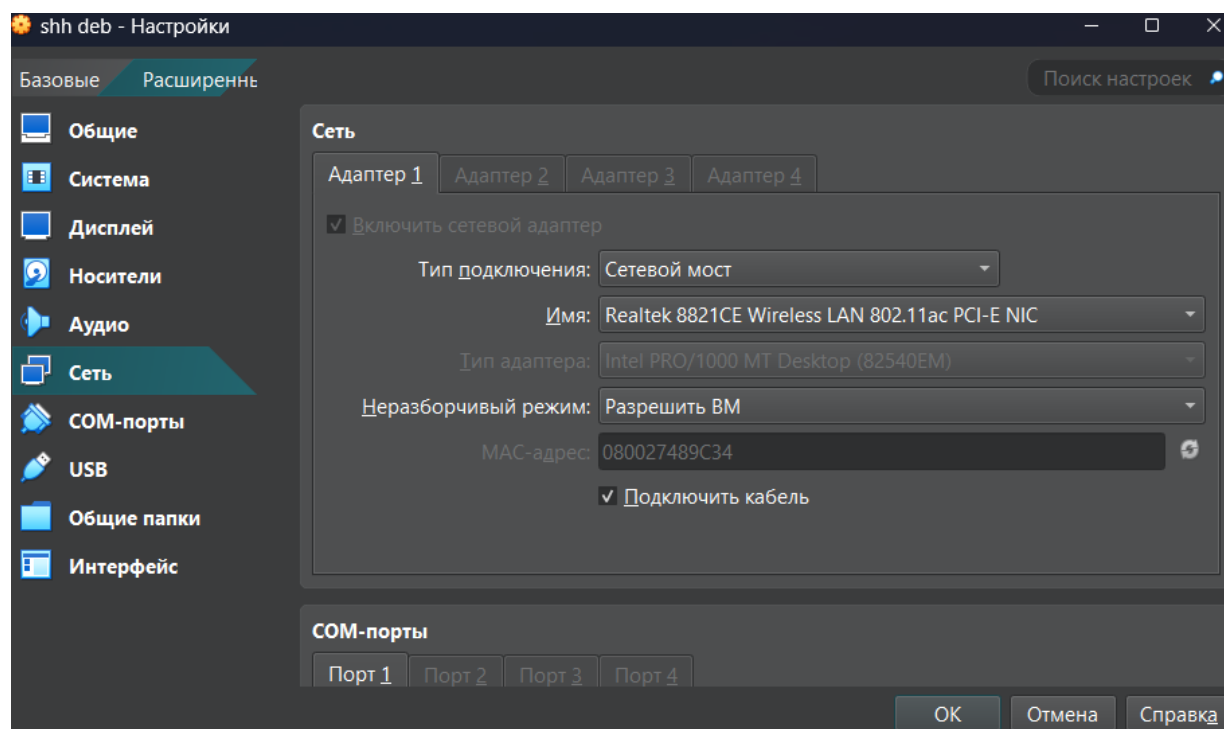


Рисунок 1 – Настройка сетевого подключения

Узнаем ip-адрес виртуальной машины и проверим с помощью команды ping на хостовой системе подключение (рис. 2).

```
PS C:\Users\alexa> ping 192.168.1.10

Обмен пакетами с 192.168.1.10 по с 32 байтами данных:
Ответ от 192.168.1.10: число байт=32 время<1мс TTL=64
Ответ от 192.168.1.10: число байт=32 время<1мс TTL=64
Ответ от 192.168.1.10: число байт=32 время<1мс TTL=64
Ответ от 192.168.1.10: число байт=32 время<1мс TTL=64

Статистика Ping для 192.168.1.10:
    Пакетов: отправлено = 4, получено = 4, потеряно = 0
    (0% потерь)
Приблизительное время приема-передачи в мс:
    Минимальное = 0мсек, Максимальное = 0 мсек, Среднее = 0 мсек
PS C:\Users\alexa> |
```

Рисунок 2 – Проверка подключения

Установим необходимые пакеты для ssh-сервера. Запустим ssh-сервер с помощью команды `systemctl start ssh` и проверим статус подключения (рис. 3).

```
Настраивается пакет openssh-server (1:9.2p1-2+deb12u4) ...
rescue-ssh.target is a disabled or a static unit not running, not starting it.
ssh.socket is a disabled or a static unit not running, not starting it.
Настраивается пакет screen (4.9.0-4) ...
Настраивается пакет inetutils-inetd (2:2.4-2+deb12u1) ...
Настраивается пакет inetutils-telnetd (2:2.4-2+deb12u1) ...
Настраивается пакет telnetd (0.17+2.4-2+deb12u1) ...
Обрабатываются триггеры для man-db (2.11.2-2) ...
Обрабатываются триггеры для debianutils (5.7-0.5~deb12u1) ...
Обрабатываются триггеры для libc-bin (2.36-9+deb12u9) ...
alex@vbox:~$ sudo systemctl enable ssh
Synchronizing state of ssh.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable ssh
alex@vbox:~$ sudo systemctl start ssh
alex@vbox:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; preset: enabled)
   Active: active (running) since Wed 2025-01-15 23:06:22 MSK; 1min 16s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Main PID: 1939 (sshd)
     Tasks: 1 (limit: 2315)
    Memory: 1.4M
       CPU: 43ms
    CGroup: /system.slice/ssh.service
            └─1939 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

янв 15 23:06:22 vbox systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
янв 15 23:06:22 vbox systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
alex@vbox:~$
```

Рисунок 3 – Запуск ssh-сервера

Для работы `telnetd` настроим демон `inetd`, запускающий по необходимости другие сетевые серверные процессы. Раскомментируем строку с `telnet` в файле `inetd.conf` и сохраним изменения (рис. 4).

```
# /etc/inetd.conf: see inetd(8) for further informations.
#
# Internet superserver configuration database.
#
# Lines starting with "#:LABEL:" or "#<off>#" should not
# be changed unless you know what you are doing!
#
# If you want to disable an entry so it is not touched during
# package updates just comment it out with a single '#' character.
#
# Packages should modify this file by using update-inetd(8).
#
# <service_name> <sock_type> <proto> <flags> <user> <server_path> <args>
#
# :INTERNAL: Internal services
#discard          stream  tcp6    nowait  root    internal
#discard          dgram   udp6    wait    root    internal
#daytime          stream  tcp6    nowait  root    internal
#time            stream  tcp6    nowait  root    internal
# :STANDARD: These are standard services.
telnet stream tcp    nowait  root    /usr/sbin/tcpd  /usr/sbin/telnetd
# :BSD: Shell, login, exec and talk are BSD protocols.
# :MAIL: Mail, news and uucp services.
# :INFO: Info services
# :BOOT: TFTP service is provided primarily for booting. Most sites
#        run this only on machines acting as "boot servers."
# :RPC: RPC based services
# :HAM-RADIO: amateur-radio services
# :OTHER: Other services
```

Рисунок 4 – Файл `inetd.conf`

Выведем в терминал строки установки и завершения соединения из файла telnet.log (рис. 8).

```
alex@vbox:~$ cat telnet.log | grep -P '\[[SF].*?\]' telnet.log
192.168.1.6.49879 > 192.168.1.10.23: Flags [S], cksum 0x465f (correct), seq 1685290478, win 65535, options [mss 1460, ssthresh 65535, tsval 0, tsoffset 0, tkeep 0, ackOK, nop, wscale 7], length 0
192.168.1.10.23 > 192.168.1.6.49879: Flags [S.], cksum 0x8387 (incorrect -> 0xb394), seq 1456357627, ack 1685290479, length 0
192.168.1.10.23 > 192.168.1.6.49879: Flags [FP.], cksum 0x8385 (incorrect -> 0xd72), seq 838:848, ack 111, win 500, length 0
192.168.1.6.49879 > 192.168.1.10.23: Flags [F.], cksum 0xea9c (correct), seq 111, ack 849, win 252, length 0
alex@vbox:~$
```

Рисунок 8 – Содержимое файла telnet.log

2. Часть II

Для настройки сетевого соединения через ssh воспользуемся утилитой tcpdump (рис. 9), позволяющую перехватывать и анализировать сетевой трафик, проходящий через компьютер, на котором запущена данная программа.

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

Рисунок 9 – Анализ сетевого трафика

На хостовой системе подключимся к гостевой с помощью команды ssh <имя пользователя в гостевой системе>@<ip-адрес гостевой системы> (рис. 10).

```
PS C:\Users\alexa> ssh alex@192.168.1.10
The authenticity of host '192.168.1.10 (192.168.1.10)' can't be established.
ED25519 key fingerprint is SHA256:VmcbeyECvB/SWyoMJDle7x/KpNQKTBYP47NqWPS3PUs.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.10' (ED25519) to the list of known hosts.
alex@192.168.1.10's password:
Linux vbox 6.1.0-28-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.119-1 (2024-11-22) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Jan 15 23:26:16 2025 from 192.168.1.6
alex@vbox:~$ uname -a
Linux vbox 6.1.0-28-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.119-1 (2024-11-22) x86_64 GNU/Linux
```

Рисунок 10 – Подключение по ssh

Создадим файл LR_6_test_file.txt в хостовой системе в папке C:/Users/alexa/. С помощью утилиты для безопасного копирования данных между системами по протоколу SSH scp передадим файл в домашнюю директорию гостевой системы (рис. 11).

```
PS C:\Users\alexa> scp C:\Users\alexa\LR_6_test_file.txt alex@192.168.1.10:/home/alex/
alex@192.168.1.10's password:
LR_6_test_file.txt                               100% 43    4.7KB/s   00:00
PS C:\Users\alexa> |
```

Рисунок 11 – Передача файла по протоколу SSH

На рисунке 12 показана успешная передача файла.

```
alex@vbox:~$ ls
composer-setup.php demo demodir LR_6_test_file.txt ssh.log telnet.log
alex@vbox:~$ cat LR_6_test_file.txt
Pakhomov Alexander Andreevich
```

Рисунок 12 – Успешная пересылка файла

Теперь настроим ssh-ключи для подключения без пароля. Для этого изменим права доступа и создадим `authorized_keys`, в котором будем хранить ключи (рис. 13).

```
alex@vbox:~$ chmod 700 /home/alex/.ssh
alex@vbox:~$ touch /home/alex/.ssh/authorized_keys
alex@vbox:~$ chmod 600 /home/alex/.ssh/authorized_keys
```

Рисунок 13 – Настройка прав доступа

Для создания ключа на гостевой машине воспользуемся командой `ssh-keygen` (рис. 14). Публичный ключ назовем под именем `keyForLab`.

```
PS C:\Users\alexa> ssh-keygen
Generating public/private ed25519 key pair.
Enter file in which to save the key (C:\Users\alexa/.ssh/id_ed25519): keyForLab
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in keyForLab
Your public key has been saved in keyForLab.pub
The key fingerprint is:
SHA256:vM50m9xxCSeHdQeg/ZQ4DU5GeK3/b2xcNgAsd5cFCs alexa@Alexander
The key's randomart image is:
+--[ED25519 256]--+
|      oo+++.o. |
|      o.==+.o. |
|      ++*E=o.  |
|      . .++Bo. |
|      S +.++o  |
|      . = .+.. |
|      o .oo +o |
|      = + o o + |
|      .B .   o |
+-----[SHA256]-----+
PS C:\Users\alexa> cat .\keyForLab
-----BEGIN OPENSSH PRIVATE KEY-----
b3B1bnNzaC1rZXktdjEAAAABG5vbmUAAAABbm9uZQAAAAAAAAABAAAAMwAAAAtzc2gtZW
QyNTUxOQAAACCB9lJBhjgqWGPz/DrI1cJ89hLqz46QiIA9EgXLlxflcAAAAJgIebMfCHmz
HwAAAAtzc2gtZWQyNTUxOQAAACCB9lJBhjgqWGPz/DrI1cJ89hLqz46QiIA9EgXLlxflcA
AAAEbQMfz00FfwKC7hzkB9sVCqE9ELxLN3BPcNP9pw8TIVZIH2UkGG0CrAanP80sjVwnz2
EurPjpCIgD0SBcuXF+VwAAAAD2FsZXhhbmlrcGECawQFBg==
-----END OPENSSH PRIVATE KEY-----
PS C:\Users\alexa> cat .\keyForLab.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIIH2UkGG0CrAanP80sjVwnz2EurPjpCIgD0SBcuXF+Vw alexa@Alexander
```

Рисунок 14 – Создание ssh-ключа

На рисунке 15 предыдущем способом с помощью команды `scp` передаем созданный ключ на гостевую систему.

```
PS C:\Users\alexa> scp C:\Users\alexa\keyForLab.pub alex@192.168.1.10:/home/alex/.ssh/temp_key.pub
alex@192.168.1.10's password:
keyForLab.pub
PS C:\Users\alexa> |
100% 98 31.9KB/s
```

Рисунок 15 – Передача ключа на гостевую систему

Скопируем ключ в ранее созданный файл (рис. 16).

```
alex@vbox:~/ssh$ cat /home/alex/.ssh/temp_key.pub >> /home/alex/.ssh/authorized_keys
alex@vbox:~/ssh$ cat authorized_keys
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIIH2UkGG0CrAanP80sjVwnz2EurPjpCIgD0SBcuXF+Vw alexa@Alexander
```

Рисунок 16 – Копирование ключа

Теперь можно создать подключение через ssh с помощью указания явного расположения ключа (рис. 17).

```
PS C:\Users\alexa> ssh alex@192.168.1.10 -i C:\Users\alexa\keyForLab
Linux vbox 6.1.0-28-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.119-1 (2024-11-22) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Jan 15 23:31:43 2025 from 192.168.1.6
```

Рисунок 17 – Подключение по ключу

Теперь можно передать новый файл с помощью scp (рис. 18).

```
PS C:\Users\alexa> scp C:\Users\alexa\keyForLab C:\Users\alexa\LR_6_test_file_NEW.txt alex@192.168.1.10:/home/alex/
alex@192.168.1.10's password:
keyForLab                                     100% 411 200.7KB/s 00:00
LR_6_test_file_NEW.txt                       100% 43 21.0KB/s 00:00
```

Рисунок 18 – Передача файла

Проверим успешность передачи файла на гостевой системе (рис. 19).

```
alex@vbox:~$ ls
composer-setup.php  keyForLab          ssh.log
demo                LR_6_test_file_NEW.txt telnet.log
demodir             LR_6_test_file.txt
alex@vbox:~$ cat LR_6_test_file_NEW.txt
Pakhomov Alexander Andreevich
LAB number 6alex@vbox:~$
```

Рисунок 19 – Успешная передача файла

Как и в предыдущей части, выведем содержимое файла ssh.log (рис. 20).

```
alex@vbox:~$ cat ssh.log | grep -P '\[SF].*?\]'
192.168.1.6.49904 > 192.168.1.10.22: Flags [S], cksum 0xa32e (correct), seq 1485974760, win 65535, op
192.168.1.10.22 > 192.168.1.6.49904: Flags [S.], cksum 0x8387 (incorrect -> 0x18e3), seq 2390622412,
ckOK,nop,wscale 7], length 0
192.168.1.6.49987 > 192.168.1.10.22: Flags [S], cksum 0xb66b (correct), seq 3053501417, win 65535, op
192.168.1.10.22 > 192.168.1.6.49987: Flags [S.], cksum 0x8387 (incorrect -> 0x3fcc), seq 3118580156,
ckOK,nop,wscale 7], length 0
192.168.1.6.50136 > 192.168.1.10.22: Flags [S], cksum 0x5029 (correct), seq 4238400246, win 65535, op
192.168.1.10.22 > 192.168.1.6.50136: Flags [S.], cksum 0x8387 (incorrect -> 0x3ff6), seq 2147783981,
ckOK,nop,wscale 7], length 0
192.168.1.6.50169 > 192.168.1.10.22: Flags [S], cksum 0x8b1b (correct), seq 3750411513, win 65535, op
192.168.1.10.22 > 192.168.1.6.50169: Flags [S.], cksum 0x8387 (incorrect -> 0x9944), seq 1384948809,
ckOK,nop,wscale 7], length 0
192.168.1.6.50196 > 192.168.1.10.22: Flags [S], cksum 0xe48f (correct), seq 1307899136, win 65535, op
192.168.1.10.22 > 192.168.1.6.50196: Flags [S.], cksum 0x8387 (incorrect -> 0x726), seq 681294285, a
ckOK,nop,wscale 7], length 0
alex@vbox:~$
```

Рисунок 20 – Содержимое файла ssh.log

Контрольные вопросы

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

Программное обеспечение удаленного доступа предназначено для:

- 1) Управления удаленными системами, обеспечения контроля и настройки серверов, рабочих станций или устройств.
- 2) Технической поддержки, удаленного решения проблем пользователей.
- 3) Совместной работы, предоставления доступа к файлам, приложениям и средам разработки.
- 4) Обеспечения мобильности, доступа к корпоративным системам из любого места.
- 5) Обучения и демонстрации, проведения презентаций или обучения через удаленный доступ.

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

SSH и Telnet – это два протокола, которые обычно используются для удалённого входа в систему и выполнения задач по настройке и управлению на устройствах, подключённых через сеть. Хотя оба типа беспроводных сетей имеют некоторое сходство в плане общего функционирования, они имеют больше особенностей в аспектах функций безопасности и операций. Благодаря строгим мерам безопасности, реализованным в его протоколах, SSH является наиболее современным в современных сетях.

SSH – это протокол, который обеспечивает безопасные, зашифрованные каналы связи по незащищённой сети. А Telnet – устаревший протокол, обеспечивающий незашифрованную связь по сети.

SSH поддерживает передачу файлов с помощью SCP (безопасное копирование) или SFTP (протокол безопасной передачи файлов). А Telnet изначально не поддерживает передачу файлов.

Telnet проще в использовании и реализации, но он считается небезопасным для большинства приложений и не используется так широко, как заменяющие его протоколы.

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

А) Способ входа по паролю. Ввод логина и пароля для доступа к удаленной системе: `ssh user@host`.

Плюсы: легкость настройки, не требуется предварительная конфигурация.

Минусы: менее безопасно.

Б) Способ установления соединения по ключам. Использование пары ключей: приватного (хранится на клиенте) и публичного (располагается на сервере).

```
ssh-keygen -t rsa
```

```
ssh-copy-id user@host
```

```
ssh user@host
```

Плюсы: высокая безопасность.

Минусы: требуется предварительная настройка.

В) Способ подключения через SSH-агент. Хранение ключей в памяти для упрощения многократных подключений.

```
eval $(ssh-agent)
```

```
ssh-add ~/.ssh/id_rsa
```

```
ssh user@host
```

Плюсы: удобство при множественных подключениях.

Минусы: потребность в поддержке SSH-агента.

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

Примером является подключение к удаленному серверу для выполнения системных команд, передачи файлов или мониторинга системы. В рамках лабораторной работы SSH использовался для подключения к серверу с ОС

Debian, выполнения команды `uname -a` для получения системной информации, а также передачи текстового файла через зашифрованный канал с использованием утилиты `scp`. Это иллюстрирует, как безопасный удаленный доступ позволяет эффективно управлять системой и обмениваться данными без необходимости физического присутствия.

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

Сетевые службы:

- Secure Shell: для удаленного доступа.
- SFTP: для передачи файлов.
- `rsync` через SSH: синхронизация данных.
- Git через SSH: безопасный доступ к репозиториям.
- Tunnels: защита сетевого трафика.

Передача файлов через безопасный SSH-туннель осуществляется с использованием таких утилит, как `scp` или `rsync` поверх SSH.

Передача файла `local_file.txt` с локального компьютера на сервер: `scp local_file.txt user@remote_host:/path/to/remote/directory/`.