

## ПРАКТИЧЕСКОЕ ЗАДАНИЕ № 6

Для организации удаленного доступа к консоли сервера ранее использовался протокол **telnet**, и в каждую сетевую операционную систему включался telnet-клиент. Эта программа так и называется – **telnet**.

Подключившись с помощью **telnet** к удаленному компьютеру, вы можете работать с ним как обычно. В окне telnet-клиента представлена как бы консоль удаленного компьютера: вы будете вводить команды и получать результат их выполнения – все так, как если бы вы работали непосредственно за удаленным компьютером.

Но технологии не стоят на месте, и протокол **telnet** устарел. Сейчас им практически никто не пользуется. На смену ему пришел протокол **SSH** (Secure Shell), который, как видно из названия, представляет собой безопасную оболочку. Главное отличие **SSH** от **telnet** состоит в том, что в соответствии с этим протоколом все данные (включая пароли доступа к удаленному компьютеру, отдельные файлы и пр.) передаются в зашифрованном виде. Причиной создания **SSH** и стало то, что во времена **telnet** участились случаи перехвата паролей и другой важной информации.

SSH (Secure Shell) – сетевой протокол, позволяющий удалённо управлять операционной системой. Благодаря шифрованию всего трафика - использование данного протокола является довольно безопасным решением.

В состав практически любого дистрибутива Linux входит SSH-сервер (программа, которая и обеспечивает удаленный доступ к компьютеру, на котором она установлена) и SSH-клиент (программа, позволяющая подключаться к SSH-серверу). Для установки SSH-сервера нужно установить пакет *openssh* (это разновидность SSH-сервера), а для установки SSH-клиента — пакет *openssh-clients*.

Если вы используете OpenSSH (а в большинстве случаев так оно и есть), все настройки SSH-сервера хранятся в одном-единственном файле: `/etc/ssh/sshd_config` (в старых версиях: `/etc/sshd_config`), а настройки программы-клиента – в файле `/etc/ssh/ssh_config` (в старых версиях: `/etc/ssh_config`). Настройки программы клиента обычно задавать не нужно, поскольку они приемлемы по умолчанию.

### Задание 1

Определите, установлен ли в вашей операционной системе пакет OpenSSH. Для этого введите в консоль команду `ssh` (рисунок 1).

```
$ ssh (1)
```

Если в терминал после применения команды выведена справка, это означает, что набор программ установлен.

```
vitaliivv@vuv-VirtualBox:~$ ssh
usage: ssh [-46AaCfGgKkMnqsTtVvXxYy] [-B bind_interface]
          [-b bind_address] [-c cipher_spec] [-D [bind_address:]port]
          [-E log_file] [-e escape_char] [-F configfile] [-I pkcs11]
          [-i identity_file] [-J [user@]host[:port]] [-L address]
          [-l login_name] [-m mac_spec] [-O ctl_cmd] [-o option] [-p port]
          [-Q query_option] [-R address] [-S ctl_path] [-W host:port]
          [-w local_tun[:remote_tun]] destination [command [argument ...]]
vitaliivv@vuv-VirtualBox:~$ ssh -v localhost
OpenSSH_8.9p1 Ubuntu-3ubuntu0.6, OpenSSL 3.0.2 15 Mar 2022
debug1: Reading configuration data /etc/ssh/ssh_config
debug1: /etc/ssh/ssh_config line 19: include /etc/ssh/ssh_config.d/*.conf matched
no files
debug1: /etc/ssh/ssh_config line 21: Applying options for *
debug1: Connecting to localhost [127.0.0.1] port 22.
debug1: connect to address 127.0.0.1 port 22: Connection refused
ssh: connect to host localhost port 22: Connection refused
vitaliivv@vuv-VirtualBox:~$
```

Рисунок 1 – Проверка наличия openSSH

Если вы не обнаружили в своей системе такого вывода, то вам следует установить OpenSSH одной из следующих команд (в зависимости от вашего менеджера пакетов):

```
$ sudo apt install openssh-server openssh-clients (2)
```

```
$ sudo dnf install -y openssh-server openssh-clients (3)
```

```
$ sudo yum -y install openssh-server openssh-clients (4)
```

Чтобы определить версию установленного OpenSSH, используйте команду (5):

```
$ ssh -v localhost (5)
```

## Задание 2

Запуск SSH-сервера производится как минимум 2 способами, представленными в таблице 1.

Таблица 1 – Команды управления сервером SSH

Действие	Файл <i>init.d</i>	Утилита <b>systemctl</b>
Запуск сервера	<code>sudo /etc/init.d/ssh start</code>	<code>sudo systemctl start ssh</code>
Остановка сервера	<code>sudo /etc/init.d/ssh stop</code>	<code>sudo systemctl stop ssh</code>
Перезапуск сервера	<code>sudo /etc/init.d/ssh restart</code>	<code>sudo systemctl restart ssh</code>

1) Выполните запуск сервера SSH.

2) Узнайте, запустился ли сервер, при помощи одной из указанных далее команд (рисунок 2, 3).

```
$ /etc/init.d/ssh status (6)
```

```
$ systemctl status ssh (7)
```

```
vitaliivv@vuv-VirtualBox:~$ sudo systemctl start ssh
vitaliivv@vuv-VirtualBox:~$ systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: en
   Active: active (running) since Sat 2024-02-24 23:28:40 +05; 1min 2s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Main PID: 51594 (sshd)
     Tasks: 1 (limit: 7010)
    Memory: 1.7M
       CPU: 11ms
   CGroup: /system.slice/ssh.service
           └─51594 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

feb 24 23:28:40 vuv-VirtualBox systemd[1]: Starting OpenBSD Secure Shell server:
feb 24 23:28:40 vuv-VirtualBox sshd[51594]: Server listening on 0.0.0.0 port 22.
feb 24 23:28:40 vuv-VirtualBox sshd[51594]: Server listening on :: port 22.
feb 24 23:28:40 vuv-VirtualBox systemd[1]: Started OpenBSD Secure Shell server.
vitaliivv@vuv-VirtualBox:~$
```

Рисунок 2 – Статус сервера SSH

```
vitaliivv@vuvv-VirtualBox:~$ sudo /etc/init.d/ssh start
Starting ssh (via systemctl): ssh.service.
vitaliivv@vuvv-VirtualBox:~$ sudo /etc/init.d/ssh status
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: en
   Active: active (running) since Sat 2024-02-24 23:32:16 +05; 10s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
    Process: 51900 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 51901 (sshd)
      Tasks: 1 (limit: 7010)
     Memory: 1.7M
        CPU: 13ms
    CGroup: /system.slice/ssh.service
            └─51901 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

фев 24 23:32:16 vuvv-VirtualBox systemd[1]: Starting OpenBSD Secure Shell ....
фев 24 23:32:16 vuvv-VirtualBox sshd[51901]: Server listening on 0.0.0.0 p..22.
фев 24 23:32:16 vuvv-VirtualBox sshd[51901]: Server listening on :: port 22.
фев 24 23:32:16 vuvv-VirtualBox systemd[1]: Started OpenBSD Secure Shell s..er.
Hint: Some lines were ellipsized, use -l to show in full.
vitaliivv@vuvv-VirtualBox:~$
```

Рисунок 3 – Статус сервера SSH

3) Определите IP-адрес вашего SSH-сервера, используя материалы практической работы № 5.

4) Выполните подключение к вашему SSH-серверу. Введите существующий в вашей операционной системе login и его пароль (рисунок 4).

\$ ssh ip\_адрес\_сервера

(8)

```
vitaliivv@vuvv-VirtualBox:~$ ssh 10.0.2.15
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
   inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
   link/ether 08:00:27:5e:d3:c0 brd ff:ff:ff:ff:ff:ff
   inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
       valid_lft 82745sec preferred_lft 82745sec
   inet6 fe80::1c75:226d:853c:4a4a/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
vitaliivv@vuvv-VirtualBox:~$ ssh 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:FfKpP2SXI3BqfDsFJ/WvkuleK5EHMd6fBZTQY/cc4SY.
This key is not known by any other names
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.
vitaliivv@10.0.2.15's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.5.0-17-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.

77 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

*** System restart required ***

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

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
```

Рисунок 4 – Удаленное подключение к SSH-серверу

После первого подключения к серверу – вы должны согласиться с предупреждением, которое означает, что после соединения будет сохранён специальный идентификатор этого соединения.

Если вы увидите это предупреждение при повторном подключении к серверу – возможно настройки соединения были изменены или вас атакует хакер!

### Задание 3

После успешного подключения к серверу – его необходимо защитить. Для этого, необходимо правильно настроить SSH-соединение.

1) Откройте файл с настройками SSH-сервера в одном из установленных текстовых редакторов, например, **nano** (рисунок 5):

\$ sudo nano /etc/ssh/sshd\_config (9)

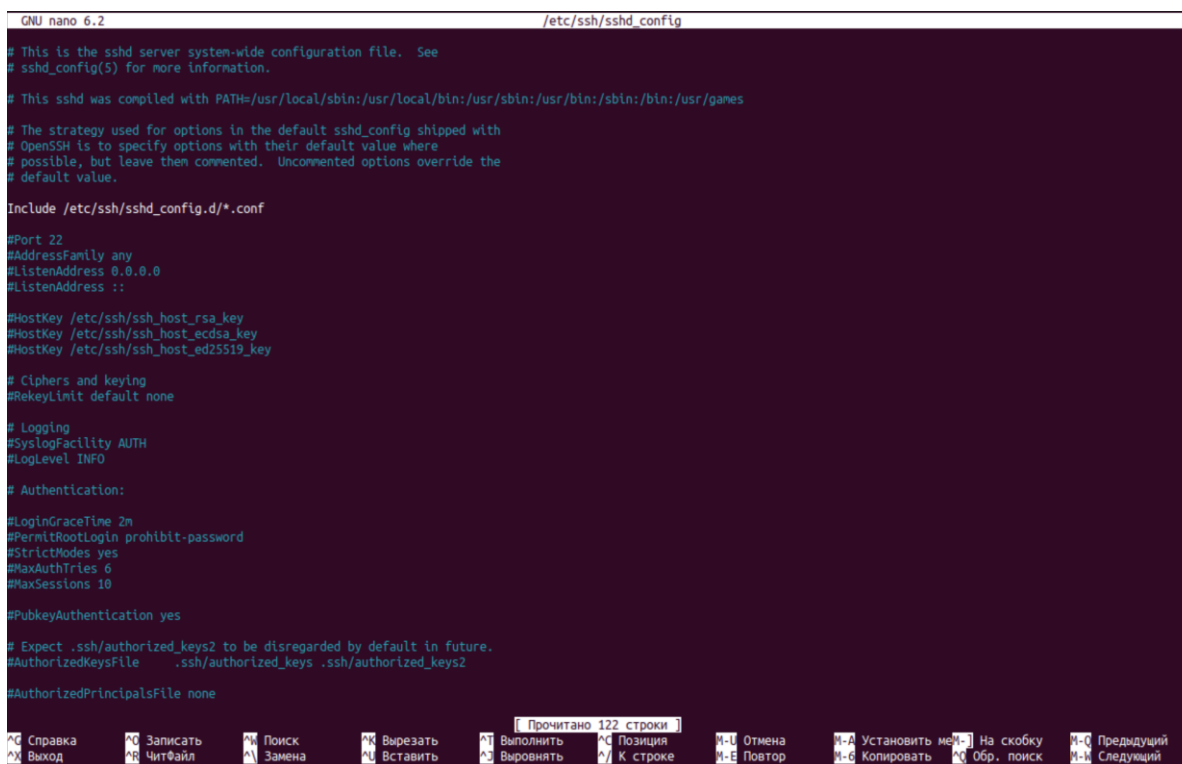


Рисунок 5 – Редактирование файла /etc/ssh/sshd\_config

Внесение/изменение настроек в этом файле работают по следующему шаблону:  
**ключевое\_слово** *аргумент*

2) Измените порт для подключения к серверу. Это необходимо для того, чтобы предотвратить хакерские атаки на дефолтный порт. Раскомментируйте строку (уберите #) и измените номер порта на четырехзначное число (рисунок 6). Последние две цифры значения порта должны соответствовать вашему номеру в списке группы.



```

GNU nano 6.2 /etc/ssh/sshd_config *

# This is the sshd server system-wide configuration file. See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr>

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented. Uncommented options override the
# default value.

Include /etc/ssh/sshd_config.d/*.conf

Port 2200
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

^C Справка ^O Записать ^M Поиск ^K Вырезать ^T Выполнить ^S Позиция
^X Выход ^R ЧитФайл ^_ Замена ^U Вставить ^J Выводить ^_ К строке

```

Рисунок 6 – Изменение значения порта для подключения

3) В таблице 2 приведены некоторые другие настройки сервера, которые чаще всего необходимо исправить.

Таблица 2 – Основные настройки сервера SSH

Описание	Ключевое слово	Аргументы
Настройка прослушиваемого IP-адреса, если на сервере их несколько	<b>ListenAddress</b>	<i>hostname   hostname:port</i>
Настройка отвечает за вход на сервер от имени <b>root</b>	<b>PermitRootLogin</b>	<i>no   yes   prohibit-root   forced-command-only</i>
Настройка отвечает за вход по паролю на сервер	<b>PasswordAuthentication</b>	<i>no   yes</i>
Если здесь выставлено значение «yes», то будет возможность войти на сервер с пустой строкой пароля.	<b>PermitEmptyPasswords</b>	<i>no   yes</i>
Настройка входа по SSH-ключу	<b>PubkeyAuthentication</b>	<i>no   yes</i>
Настройка максимального количества попыток входа.	<b>MaxAuthTries</b>	<i>число</i>
Настройка максимального количества одновременных подключений с одного IP	<b>MaxStartups</b>	<i>число</i>
Настройка времени, за которое должен залогиниться пользователь	<b>LoginGraceTime</b>	<i>число</i>
Настройка пользователей или групп, которым можно войти на сервер*	<b>AllowUsers</b> или <b>AllowGroups</b>	<i>имя_пользователя</i> или <i>имя_группы</i>
Настройка пользователей или групп, которым нельзя входить на сервер*	<b>DenyUsers</b> или <b>DenyGroups</b>	<i>имя_пользователя</i> или <i>имя_группы</i>

\* Эти переменные принимают так называемые «шаблоны»: а) символ «\*» – означает любое значение, например: **AllowUsers**

**\*@192.168.0.11** – вход под любым пользователем разрешён только для IP 192.168.0.11 или: **AllowUsers** *user1*@192.168.0.\*

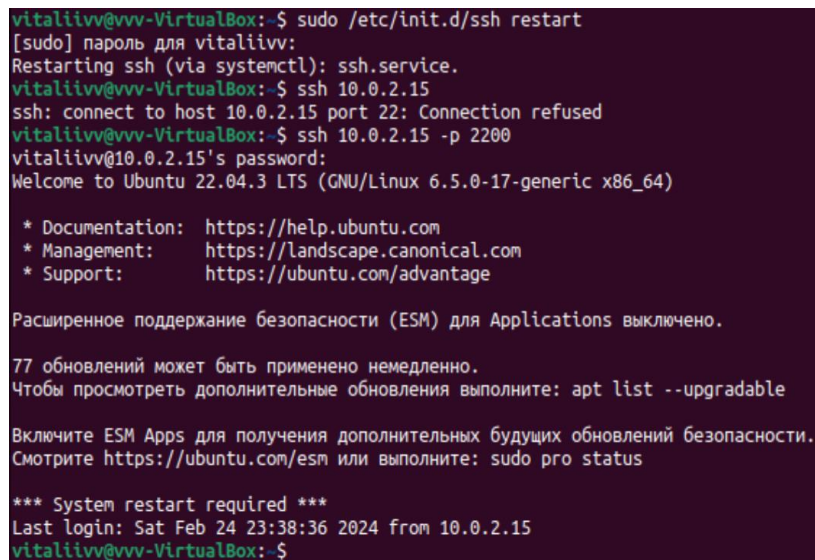
– вход под юзером «user1» разрешён любому IP-адресу, который начинается на «192.168.0.»

Настройте следующие ограничения для вашего сервера:

- А) запретить вход от пользователя root;
- Б) ограничить время, за которое Пользователь должен залогиниться 20 секундами;
- В) ограничить количество попыток входа 2-мя;
- Г) ограничить количество входов с одного IP-адреса 1-им.

4) После внесения изменений в файл конфигурации SSH-сервера, сохраните его и перезапустите службу SSH.

5) Протестируйте ограничения сервера, которые вы внесли в его конфигурацию (рисунки 7, 8).



```
vitaliiv@vuv-VirtualBox:~$ sudo /etc/init.d/ssh restart
[sudo] пароль для vitaliiv:
Restarting ssh (via systemctl): ssh.service.
vitaliiv@vuv-VirtualBox:~$ ssh 10.0.2.15
ssh: connect to host 10.0.2.15 port 22: Connection refused
vitaliiv@vuv-VirtualBox:~$ ssh 10.0.2.15 -p 2200
vitaliiv@10.0.2.15's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.5.0-17-generic x86_64)

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

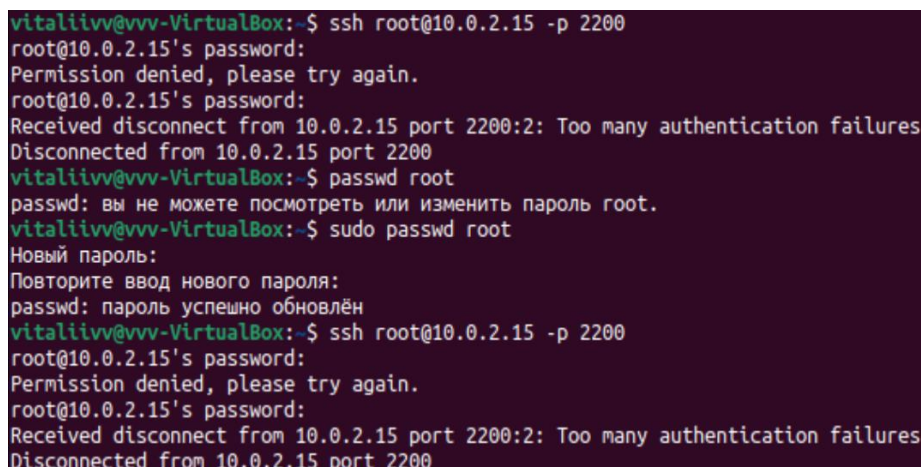
Расширенное поддержание безопасности (ESM) для Applications включено.

77 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

*** System restart required ***
Last login: Sat Feb 24 23:38:36 2024 from 10.0.2.15
vitaliiv@vuv-VirtualBox:~$
```

Рисунок 7 – Тестирование ограничения подключения по порту



```
vitaliiv@vuv-VirtualBox:~$ ssh root@10.0.2.15 -p 2200
root@10.0.2.15's password:
Permission denied, please try again.
root@10.0.2.15's password:
Received disconnect from 10.0.2.15 port 2200:2: Too many authentication failures
Disconnected from 10.0.2.15 port 2200
vitaliiv@vuv-VirtualBox:~$ passwd root
passwd: вы не можете посмотреть или изменить пароль root.
vitaliiv@vuv-VirtualBox:~$ sudo passwd root
Новый пароль:
Повторите ввод нового пароля:
passwd: пароль успешно обновлён
vitaliiv@vuv-VirtualBox:~$ ssh root@10.0.2.15 -p 2200
root@10.0.2.15's password:
Permission denied, please try again.
root@10.0.2.15's password:
Received disconnect from 10.0.2.15 port 2200:2: Too many authentication failures
Disconnected from 10.0.2.15 port 2200
```

Рисунок 8 – Тестирование ограничения подключения от имени пользователя root и по количеству попыток входа

#### Задание 4

Приветственное сообщение, которое операционная система выводит в консоль при подключении, можно изменять. Используйте свои навыки самостоятельного поиска информации в справочной литературе и сети Интернет и приведите приветственное сообщение к следующему виду, показанному на рисунке 9. Имя пользователя, от имени которого вы логинитесь, а также текущую

время и дату вы должны выводить актуальную, то есть соответствующую ситуации. Используйте для этого изученные ранее команды и возможности.

```
1 Приветствую вас, имя_пользователя!  
2 Сейчас _текущее_время_и_дата_  
3 Вы, просто молодец, что справились!  
4 Для вас нет ничего невозможного  
5  
6 Last login: _дата_последнего_входа_
```

Рисунок 9 – Текст приветственного сообщения при подключении по SSH

## Задание 5

По умолчанию для подключения к серверу через SSH используется пароль. Однако это не самый высокий уровень безопасности. Чтобы защитить систему от несанкционированного доступа, необходимо настроить ключи SSH.

Secure Shell обеспечивает безопасное удалённое подключение к операционной системе. С его помощью можно получить доступ к оболочке и передавать данные.

Базовая конфигурация состоит из клиента и сервера.

- Клиент используется на компьютере, который устанавливает соединение.
- Сервер работает в системе, с которой должно быть установлено соединение.

Важное преимущество – кроссплатформенность. Например, вы можете использовать клиент, работающий на Linux, Windows или macOS, для подключения к серверу, работающему на Ubuntu. Это позволит вам получить доступ к командной строке оболочки или выполнить передачу файлов. Все коммуникации между клиентом и сервером зашифрованы, чтобы предотвратить перехват данных посторонними лицами.

Слабость базовой реализации заключается в том, что она полностью зависит от надёжности парольных фраз, которые назначены учётным записям. Если злоумышленник узнает пароль от учётной записи, то система станет уязвимой. Устранить эту слабость помогает аутентификация на основе ключей.

Аутентификация на основе ключей использует асимметричное шифрование, чтобы добавить дополнительный уровень безопасности к удалённому доступу к системе. Концепция шифрования с открытым ключом была разработана в 1975 году Уитфилдом Диффи и Мартином Хеллманом и основана на концепции использования пары ключей – одного приватного (private key) и одного открытого (public key).

Публичная часть этой пары используется для шифрования данных, которые может расшифровать только владелец закрытой части пары.

При настройке аутентификации на основе SSH-ключей приватная часть хранится на хосте, на котором расположен клиент, а соответствующий открытый ключ находится в системе, в которой работает сервер SSH. Важно защитить private key, так как владение им позволит любому войти в удаленную систему. В качестве дополнительного уровня защиты закрытый ключ также может быть

зашифрован и защищен паролем, который необходимо вводить каждый раз при установлении соединения с сервером.

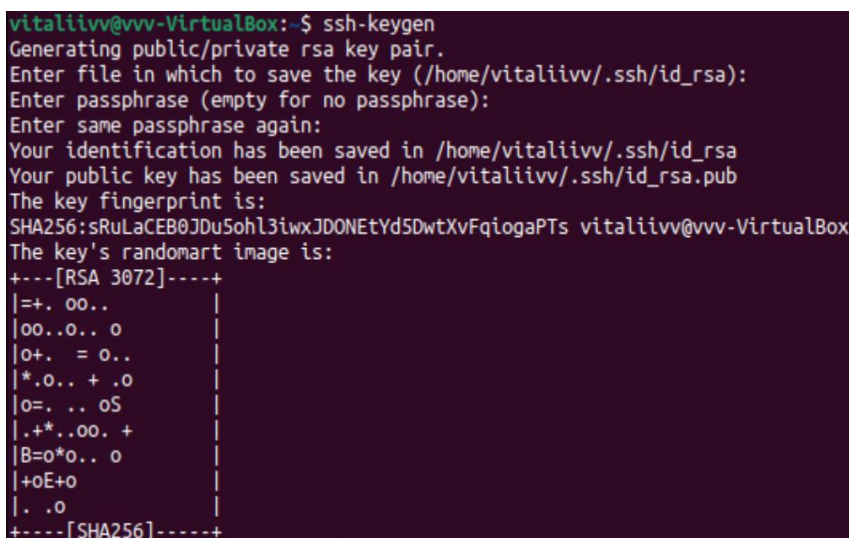
Этот подход можно сравнить с пазлом. Представьте, что у вас есть картинка. Вы взяли и разорвали её на две части – сгенерировали уникальную пару. Даже если вы распечатаете такую же картинку еще раз и снова её порвёте, повторить пару не получится.

Одну часть вы отдаёте хосту, а другую храните у себя. При встрече вы показываете свой фрагмент, а хост – свой. Если они совпадают, то вы пожимаете друг другу руки и обмениваетесь данными. Попытайтесь обмануть и подсунуть другой фрагмент – хост ничего не отдаст.

Самый простой способ создать пару для аутентификации на Linux и macOS – использовать встроенную утилиту **ssh-keygen**.

1) Сгенерируйте пару ключей аутентификации. Для простоты генерации просто подтверждайте запрашиваемые опции нажатием клавиши <Enter> (рисунок 10). В будущем, самостоятельно изучите, почему так делать не стоит.

```
$ ssh-keygen (10)
```



```
vitaliiv@vuv-VirtualBox:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/vitaliiv/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/vitaliiv/.ssh/id_rsa
Your public key has been saved in /home/vitaliiv/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:sRuLaCEB0JD5oh13iwxJDONetYd5DwtXvFqiogaPTs vitaliiv@vuv-VirtualBox
The key's randomart image is:
+---[RSA 3072]-----+
|=. 00.. |
|00..0.. o |
|0+. = 0.. |
|*.0.. + .0 |
|0=. .. oS |
|.+*..00. + |
|B=0*o.. o |
|+0E+o |
|. .o |
+---[SHA256]-----+
```

Рисунок 10 – Создание пары ключей

2) Автоматически перенести значение public key помогает встроенная утилита **ssh-copy-id**. Этот способ доступен на Linux и macOS. В терминале выполните команду:

```
$ ssh-copy-id -p номер_порта имя_пользователя@адрес_сервера (11)
```

В каталоге пользователя, под которым вы хотите заходить на сервер, если создать файл `~/.ssh/authorized_keys` и положить туда открытый ключ, то можно будет заходить без пароля. Обратите внимание, права на файл не должны давать возможность писать в этот файл посторонним пользователям, иначе ssh его не примет. В ключе последнее поле – `user@machine`. Оно не имеет никакого отношения к авторизации и служит только для удобства определения – где чей ключ.

3) При использовании SSH-ключей в Ubuntu становится возможным отключение парольной проверки. Это сделает подключение безопаснее: нет пароля – никто не может его украсть и получить



доступ к вашему серверу. Отключите проверку пароля на вашем сервере и проверьте работу выхода по ключу на SSH-сервер (рисунок 11). Не забудьте включить поддержку аутентификации по ключу!

```
vitaliivv@vuvv-VirtualBox:~$ ssh 10.0.2.15 -p 2200
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.5.0-17-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.

77 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

*** System restart required ***
Last login: Sun Feb 25 01:49:32 2024 from 10.0.2.15
vitaliivv@vuvv-VirtualBox:~$
```

Рисунок 11 – Вход на SSH-сервер по ключу

Переименуйте файл `~/.ssh/authorized_keys` в `~/.ssh/authorized_keys1` и попробуйте вновь подключиться к серверу (рисунок 12).

```
vitaliivv@vuvv-VirtualBox:~/.ssh$ ls
authorized_keys1 id_rsa id_rsa.pub known_hosts known_hosts.old
vitaliivv@vuvv-VirtualBox:~/.ssh$ ssh 10.0.2.15 -p 2200
vitaliivv@10.0.2.15: Permission denied (publickey).
vitaliivv@vuvv-VirtualBox:~/.ssh$
```

Рисунок 12 – Ошибка входа на SSH-сервер при отсутствии файла `authorized_keys`

### *Список контрольных вопросов*

1. Что такое SSH-протокол?
2. В чем отличие протокола **telnet** от **SSH**?
3. Назовите хотя бы один способ определить, установлен в вашей операционной системе пакет OpenSSH.
4. Назовите хотя бы один способ вывести в консоль текущий статус SSH-сервера.
5. Какой шаблон настроек применяется для конфигурации сервера SSH?
6. Какая настройка SSH-сервера отвечает за ограничение входа по IP-адресу?
7. Какая настройка SSH-сервера отвечает за запрет входа от имени **root**?
8. Какая настройка SSH-сервера отвечает за отключения необходимости ввода пароля при подключении?
9. Какая настройка SSH-сервера отвечает за включение входа по ключу шифрования?
10. Какая настройка SSH-сервера отвечает за ограничение количества удаленных подключений?
11. Как работает аутентификация по ключам?