



БЕЗОПАСНОСТЬ ОПЕРАЦИОННЫХ СИСТЕМ

ПРАКТИЧЕСКОЕ ЗАДАНИЕ: БЕЗОПАСНОСТЬ ОС LINUX

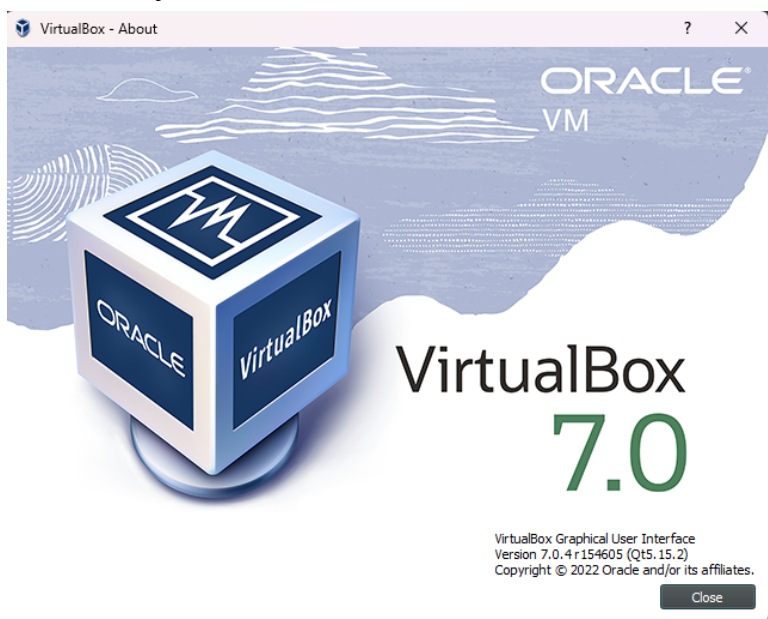
Выполнил: Андрей Степаненко (MIFIIB/2-й поток)

Разверните виртуальную машину на любом дистрибутиве, основанном на Debian (Ubuntu, Debian...)

В качестве гипервизора, для создания виртуальной машины с Ubuntu Desktop 22.04:

```
asko@LXSYNC-000WAVE:~$ lsb_release -drc
Description:    Ubuntu 22.04.3 LTS
Release:        22.04
Codename:       jammy
asko@LXSYNC-000WAVE:~$
```

используем Virtual Box 7.0::



со следующими параметрами:

OS: Ubuntu Desktop 22.04 (Codename: Jammy)

CPU: 2CPU

RAM: 8 GB

HDD: 15GB

Network: Intel PRO/1000 MT Desktop (NAT)

CD/DVD: для установки системы с ISO-образа

DISPLAY: VMSVGA/16MB - Максимальное разрешение

1. Установить SSH-сервер и настроить удаленное подключение по ключам, вместо пароля.

Установка SSH-сервера

В данной практической работе будет использоваться **OpenSSH**, который является одной из реализаций серверной части протокола **SSH** и предоставляет множество функций для обеспечения безопасного удаленного доступа к серверам.

Обновляем пакеты системы с помощью команды:

```
asko@LXSYNC-000WAVE: ~  
asko@LXSYNC-000WAVE:~$ sudo apt update  
[sudo] password for asko:  
Hit:1 http://security.ubuntu.com/ubuntu jammy-security InRelease  
Hit:2 http://sa.archive.ubuntu.com/ubuntu jammy InRelease  
Hit:3 http://sa.archive.ubuntu.com/ubuntu jammy-updates InRelease  
Hit:4 http://sa.archive.ubuntu.com/ubuntu jammy-backports InRelease  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
4 packages can be upgraded. Run 'apt list --upgradable' to see them.  
asko@LXSYNC-000WAVE:~$
```

Устанавливаем пакет **OpenSSH-server** с помощью команды :

```
asko@LXSYNC-000WAVE:~$ sudo apt install -y openssh-server  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following additional packages will be installed:  
  ncurses-term openssh-sftp-server ssh-import-id  
Suggested packages:  
  molly-guard monkeysphere ssh-askpass
```

Выводим информацию о версии **OpenSSH** командой:

```
asko@LXSYNC-000WAVE:~$ ssh -V  
OpenSSH_8.9p1 Ubuntu-3ubuntu0.3, OpenSSL 3.0.2 15 Mar 2022
```

"OpenSSH_8.9p1" - версия сервера OpenSSH

"Ubuntu-3ubuntu0.3" - версия OpenSSH, скомпилированная и упакованная для

операционной системы Ubuntu

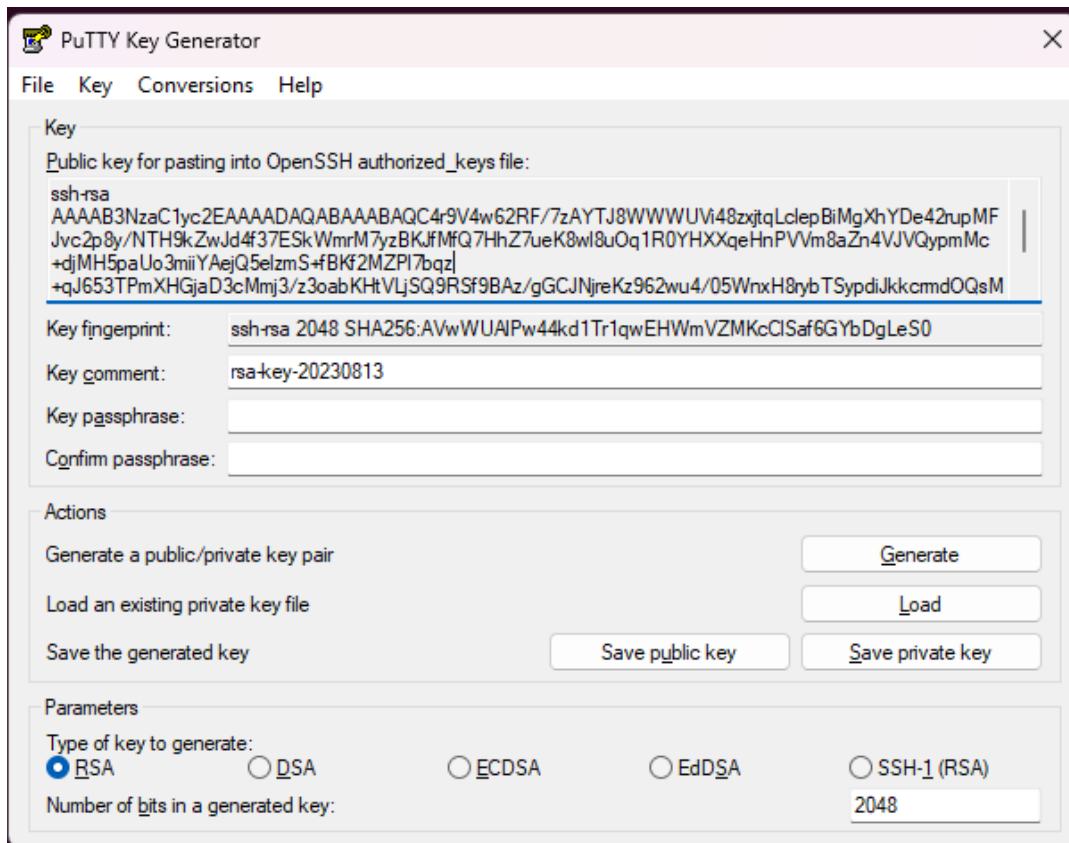
"OpenSSL 3.0.5" - версия библиотеки OpenSSL, используемой сервером OpenSSH для шифрования и аутентификации

"15 Mar 2022" - дата выпуска библиотеки OpenSSL

Вводим команды для запуска службы **SSH** и вывода информации о её текущем состоянии:

```
asko@LXSYNC-000WAVE:~$ sudo systemctl start ssh ←  
[sudo] password for asko:  
asko@LXSYNC-000WAVE:~$ sudo systemctl status ssh ←  
● ssh.service - OpenBSD Secure Shell server  
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)  
   → Active: active (running) since Sun 2023-08-13 09:36:34 MSK; 23min ago  
     Docs: man:sshd(8)  
           man:sshd_config(5)  
  Main PID: 4112 (sshd)  
    Tasks: 1 (limit: 4476)  
   Memory: 1.7M  
      CPU: 15ms  
   CGroup: /system.slice/ssh.service  
           └─4112 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"  
  
abr 13 09:36:34 LXSYNC-000WAVE systemd[1]: Starting OpenBSD Secure Shell server...  
abr 13 09:36:34 LXSYNC-000WAVE sshd[4112]: Server listening on 0.0.0.0 port 22.  
abr 13 09:36:34 LXSYNC-000WAVE sshd[4112]: Server listening on :: port 22.  
abr 13 09:36:34 LXSYNC-000WAVE systemd[1]: Started OpenBSD Secure Shell server.  
asko@LXSYNC-000WAVE:~$
```

В данной работе мы будем подключаться к SSH-серверу с удаленного компьютера с ОС Windows, а также, сгенерируем наши ключи при помощи программы PuTTYgen. Сгенерируем пару ключей с помощью утилиты PuTTYgen:



PuTTY Key Generator

File Key Conversions Help

Key

Public key for pasting into OpenSSH authorized_keys file:

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCA4r9V4w62RF/7zAYTJ8WWVUvi48zxtqLclepBiMgXhYDe42rupMFJvc2p8y/NTH9kZwJd4f37ESkWmrM7yzBKJfMfQ7HhZ7ueK8wl8uOq1R0YHXXqeHnPVVm8aZn4VJVQypmMc+djMH5paUo3miiYAejQ5elzmS+fBKf2MZPI7bqz|+qJ653TPmXHGjaD3cMmj3/z3oabKHtVLjSQ9RSf9BAz/gGCJNjreKz962wu4/05WnxH8rybTSypdiJkkcmdOQsM
```

Key fingerprint: ssh-rsa 2048 SHA256:AVwWUAlPw44kd1Tr1qwEHWmVZMKcClSaf6GYbDgLeS0

Key comment: rsa-key-20230813

Key passphrase:

Confirm passphrase:

Actions

Generate a public/private key pair Generate

Load an existing private key file Load

Save the generated key Save public key Save private key

Parameters

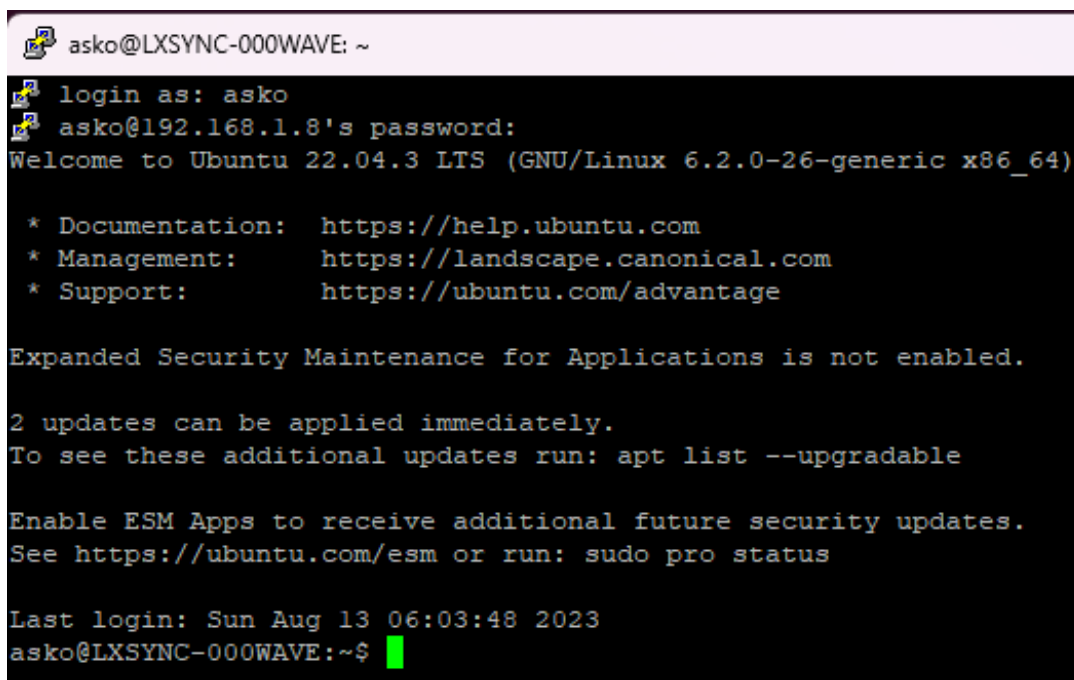
Type of key to generate:

☒ RSA ☐ DSA ☐ ECDSA ☐ EdDSA ☐ SSH-1 (RSA)

Number of bits in a generated key: 2048

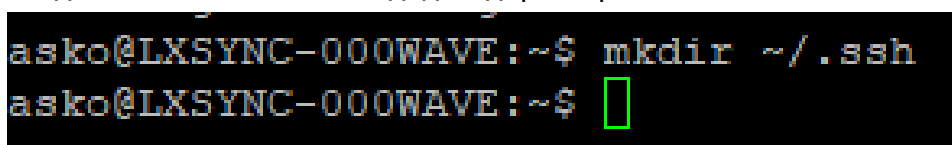
Далее, нам необходимо сохранить и загрузить Открытый(Public key) ключ на сервер, а Закрытый(Private key) ключ мы сохраним на свой компьютер, с которого будем подключаться к SSH-серверу.

Подключаемся к SSH-серверу, с помощью логина и пароля:



```
asko@LXSYNC-000WAVE: ~  
login as: asko  
asko@192.168.1.8's password:  
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-26-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage  
  
Expanded Security Maintenance for Applications is not enabled.  
  
2 updates can be applied immediately.  
To see these additional updates run: apt list --upgradable  
  
Enable ESM Apps to receive additional future security updates.  
See https://ubuntu.com/esm or run: sudo pro status  
  
Last login: Sun Aug 13 06:03:48 2023  
asko@LXSYNC-000WAVE:~$
```

В домашнем каталоге создадим директорию .ssh:



```
asko@LXSYNC-000WAVE:~$ mkdir ~/.ssh  
asko@LXSYNC-000WAVE:~$
```

Далее необходимо будет добавить наш Public key в файл «.ssh/authorized_keys».

Переходим в эту папку, создаём файл и вставляем туда Public key:

```
asko@LXSYNC-000WAVE: ~/.ssh
asko@LXSYNC-000WAVE:~/.ssh$ nano authorized_keys
asko@LXSYNC-000WAVE:~/.ssh$ ls
authorized_keys
asko@LXSYNC-000WAVE:~/.ssh$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQAC4r9V4w62RF/7zAYTJ8WWWUVi48zxjtqLc
lepBiMgXhYDe42rupMFJvc2p8y/NTH9kZwJd4f37ESkWmrM7yzBKJfMfQ7HhZ7ue
K8w18uOq1R0YHXXqeHnPVVm8aZn4VJVQypmMc+djMH5paUo3miiYAejQ5elzmS+f
BKf2MZP17bqz+qJ653TPmXHGjaD3cMmj3/z3oabKHtVLjSQ9RSf9BAz/gGCJNjre
Kz962wu4/05WnxH8rybTSypdiJkkcrmdOQsMpesVsiZCIe5PLfAF29tVW94KTeyT
fsxUatERY2qHtx2JZcseXuxSVkGpjw/6evnhCFhjepShyGQQFbmV rsa-key-20230813
asko@LXSYNC-000WAVE:~/.ssh$
```

Настройка SSH-сервера

С помощью редактора Nano, отредактируем конфигурационный файл SSH-сервера "/etc/ssh/sshd_config"

```
asko@LXSYNC-000WAVE:~$ sudo nano /etc/ssh/sshd_config
```

```
asko@LXSYNC-000WAVE: ~
GNU nano 6.2 /etc/ssh/sshd_config
PermitRootLogin no
PasswordAuthentication no
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

PubkeyAuthentication yes
AuthorizedKeysFile .ssh/authorized_keys
```

PermitRootLogin no - запретить аутентификацию по паролю для аккаунта root

PasswordAuthentication no - запретить аутентификацию по паролю

PubkeyAuthentication yes - разрешить аутентификацию по открытым ключам.

AuthorizedKeysFile .ssh/authorized_keys - путь к файлу с открытыми ключами пользователей

Также, можно определить и другие настройки безопасности для SSH-сервера, вот некоторые из них:

Изменение порта SSH сервера (по умолчанию 22) для усложнения попыток взлома

Ограничение доступа только определенным пользователям или группам

Ограничение доступа к SSH-серверу с определенных IP-адресов или диапазонов IP-адресов.

AllowUsers <имя_пользователя>@<IP-адрес или диапазон адресов

После внесения изменений в файл настроек необходимо перезапустить SSH-сервер, чтобы изменения вступили в силу

```
asko@LXSYNC-000WAVE: ~  
● ssh.service - OpenBSD Secure Shell server  
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enable  
   Active: active (running) since Sun 2023-08-13 13:39:36 MSK; 10s ago  
     Docs: man:sshd(8)  
           man:sshd_config(5)  
  Process: 4012 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)  
 Main PID: 4013 (sshd)  
    Tasks: 1 (limit: 4476)  
   Memory: 1.7M  
      CPU: 14ms  
   CGroup: /system.slice/ssh.service  
           └─4013 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"  
  
авг 13 13:39:36 LXSYNC-000WAVE systemd[1]: Starting OpenBSD Secure Shell server...  
авг 13 13:39:36 LXSYNC-000WAVE sshd[4013]: Server listening on 0.0.0.0 port 22.  
авг 13 13:39:36 LXSYNC-000WAVE sshd[4013]: Server listening on :: port 22.  
авг 13 13:39:36 LXSYNC-000WAVE systemd[1]: Started OpenBSD Secure Shell server.
```

Проверяем подключение к SSH-серверу с помощью нашего закрытого ключа, который мы сгенерировали и сохранили ранее:

```
asko@LXSYNC-000WAVE: ~  
login as: asko  
Authenticating with public key "rsa-key-20230813" ←  
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-26-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:       https://ubuntu.com/advantage  
  
Expanded Security Maintenance for Applications is not enabled.  
  
3 updates can be applied immediately.  
1 of these updates is a standard security update.  
To see these additional updates run: apt list --upgradable  
  
Enable ESM Apps to receive additional future security updates.  
See https://ubuntu.com/esm or run: sudo pro status  
  
Last login: Mon Aug 14 14:29:27 2023 from 192.168.1.5  
asko@LXSYNC-000WAVE:~$
```

Мы видим, что ввод пароля для пользователя больше не требуется.

Содержимое конфигурационного файла `sshd_config`

```
asko@LXSYNC-000WAVE:~$ sudo nano /etc/ssh/sshd_config
[sudo] password for asko:
asko@LXSYNC-000WAVE:~$ sudo grep -v '^#' /etc/ssh/sshd_config
```

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

```
Port 22
```

```
PermitRootLogin no
PasswordAuthentication no
PubkeyAuthentication yes
```

```
AuthorizedKeysFile .ssh/authorized_keys
```

```
KbdInteractiveAuthentication no
```

```
KbdInteractiveAuthentication no
UsePAM yes
```

```
X11Forwarding yes
PrintMotd no
```

```
AcceptEnv LANG LC_*
```

```
Subsystem sftp /usr/lib/openssh/sftp-server
```

```
asko@LXSYNC-000WAVE:~$ █
```

Содержимое файла `authorized_keys`:

```
GNU nano 6.2 authorized_keys *
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCA4r9V4w62RF/7zAYTJ8WWUUVi48zxjqtqLclepBiMgXhYDe42rupMFJvc2p8y/
NTH9kZwJd4f37ESkWmrM7yzBKJfMfQ7HhZ7ueK8w18u0q1R0YHXXqeHnPVVm8aZn4VJVQypmMc+djMH5paUo3miiVAejQ5elzmS
+fBKf2MZPL7bqz+qJ653TPmXHGjaD3cMmj3/z3oabKHtVLjSQ9RSf9BAz/gGCJNjreKz962wu4/05WnxH8rybTSypdiJkkcrmd0
QsMpesVsiZCie5PLfAF29tVW94KTeyTfsxUatERY2qHtx2JZcseXuxSVkGpJw/6evnhCFhjepShyGQQfbmV rsa-key-20230813
```

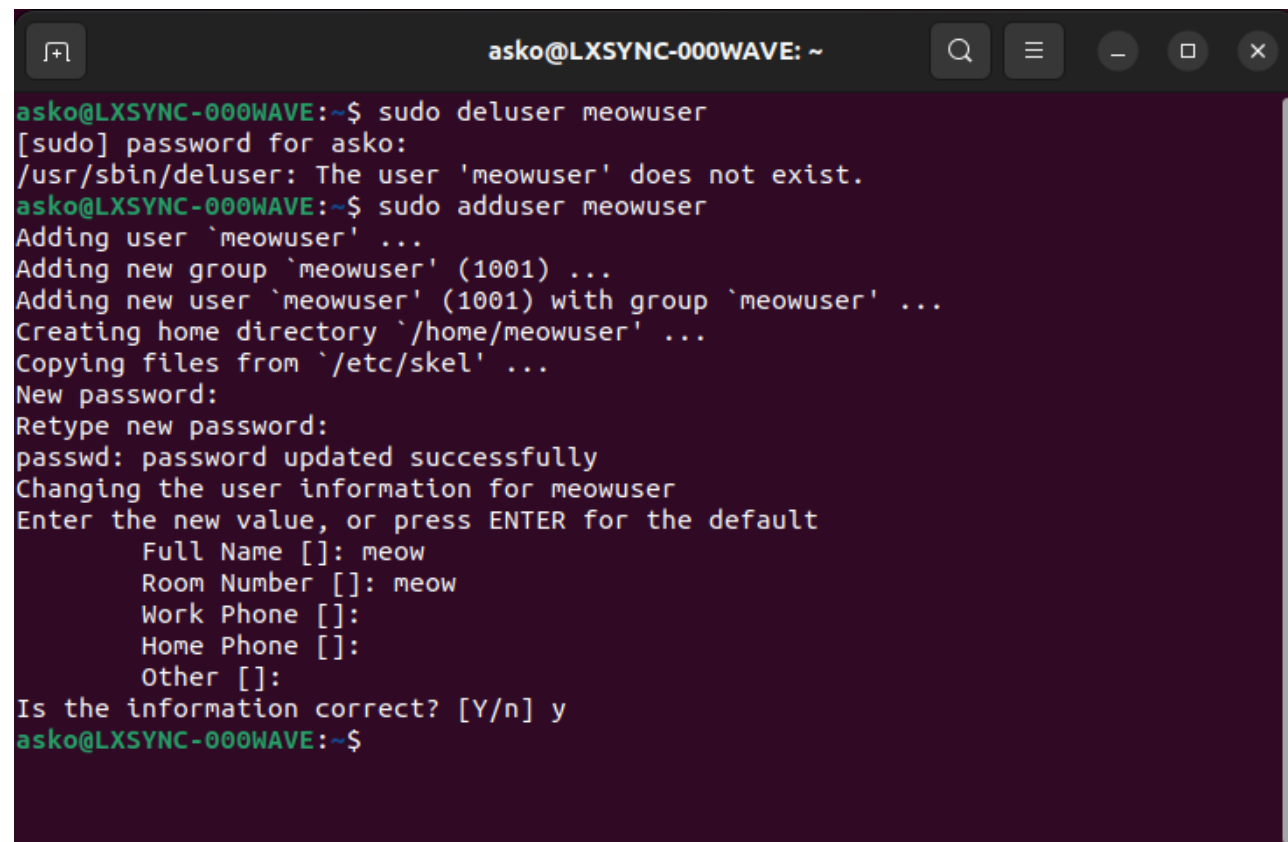
2. Создать нового пользователя с домашней директорией и выдать ему возможность запускать следующие утилиты без требования пароля:

```
# /sbin/route, /sbin/iptables, /sbin/ifup, /sbin/ifdown  
# /usr/bin/nmap, /usr/sbin/hping3, /usr/bin/systemctl
```

Для установки утилит, кроме дефолтной `systemctl`, используем следующие команды:

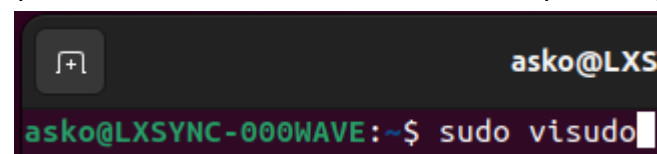
```
# /sbin/route - sudo apt install net-tools  
# /sbin/iptables - sudo apt install iptables  
# /sbin/ifup, /sbin/ifdown - sudo apt install ifupdown  
# /usr/bin/nmap - sudo apt install nmap  
# /usr/sbin/hping3 - sudo apt-get install hping3
```

Создаём нового пользователя, «meowuser» следующей командой:



```
asko@LXSYNC-000WAVE: ~  
asko@LXSYNC-000WAVE:~$ sudo deluser meowuser  
[sudo] password for asko:  
/usr/sbin/deluser: The user 'meowuser' does not exist.  
asko@LXSYNC-000WAVE:~$ sudo adduser meowuser  
Adding user `meowuser' ...  
Adding new group `meowuser' (1001) ...  
Adding new user `meowuser' (1001) with group `meowuser' ...  
Creating home directory `/home/meowuser' ...  
Copying files from `/etc/skel' ...  
New password:  
Retype new password:  
passwd: password updated successfully  
Changing the user information for meowuser  
Enter the new value, or press ENTER for the default  
Full Name []: meow  
Room Number []: meow  
Work Phone []:  
Home Phone []:  
Other []:  
Is the information correct? [Y/n] y  
asko@LXSYNC-000WAVE:~$
```

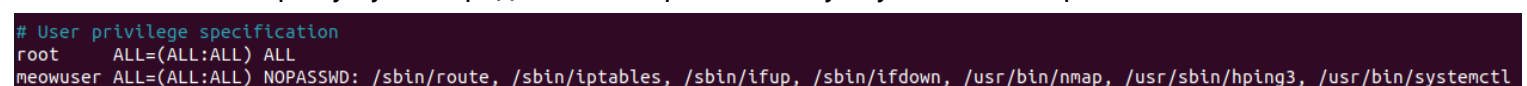
Для того чтобы пользователь мог запускать указанные утилиты без требования пароля, воспользуемся механизмом `sudoers` и настроим соответствующие правила доступа. Вводим команду для редактирования файла `sudoers` с помощью текстового редактора `visudo`:



```
asko@LXSYNC-000WAVE:~$ sudo visudo
```

В открывшемся файле `sudoers` в секции "User privilege specification" (спецификация привилегий пользователя).

Добавим следующую строку с нашими утилитами через запятую и без пробелов, указывая имя пользователя, которому нужно предоставить права на запуск утилит без пароля:



```
# User privilege specification  
root    ALL=(ALL:ALL) ALL  
meowuser ALL=(ALL:ALL) NOPASSWD: /sbin/route, /sbin/iptables, /sbin/ifup, /sbin/ifdown, /usr/bin/nmap, /usr/sbin/hping3, /usr/bin/systemctl
```

Сохраним и закроем файл `sudoers`, для того чтобы изменения вступили в силу, перезагружаемся и заходим с аккаунта нового пользователя.

Проверяем результат - ввод пароля для запуска вышеперечисленных утилит не требуется:

```
meowuser@LXSYNC-000WAVE: ~  
meowuser@LXSYNC-000WAVE:~$ sudo systemctl status ssh  
● ssh.service - OpenBSD Secure Shell server  
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: e  
   Active: active (running) since Mon 2023-08-14 14:34:34 MSK; 59min ago  
     Docs: man:sshd(8)  
           man:sshd_config(5)  
   Main PID: 3391 (sshd)  
     Tasks: 1 (limit: 4592)  
    Memory: 1.7M  
       CPU: 88ms  
    CGroup: /system.slice/ssh.service  
            └─3391 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"  
  
авг 14 14:34:33 LXSYNC-000WAVE systemd[1]: Starting OpenBSD Secure Shell server>  
авг 14 14:34:34 LXSYNC-000WAVE sshd[3391]: Server listening on 0.0.0.0 port 22.  
авг 14 14:34:34 LXSYNC-000WAVE sshd[3391]: Server listening on :: port 22.  
авг 14 14:34:34 LXSYNC-000WAVE systemd[1]: Started OpenBSD Secure Shell server.  
авг 14 14:35:12 LXSYNC-000WAVE sshd[3396]: error: Received disconnect from 192.>  
авг 14 14:35:12 LXSYNC-000WAVE sshd[3396]: Disconnected from authenticating use>  
авг 14 14:35:40 LXSYNC-000WAVE sshd[3400]: Connection closed by 192.168.1.5 por>  
авг 14 14:36:31 LXSYNC-000WAVE sshd[3402]: Accepted publickey for asko from 192>  
авг 14 14:36:31 LXSYNC-000WAVE sshd[3402]: pam_unix(sshd:session): session open>  
lines 1-21/21 (END)
```

вывод команды ls в директории home:

```
meowuser@LXSYNC-000WAVE: ~  
meowuser@LXSYNC-000WAVE:~$ ls /home/  
asko  meowuser  
meowuser@LXSYNC-000WAVE:~$
```


вывод файла passwd:

```
asko@LXSYNC-000WAVE:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin)/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:102:105:/:nonexistent:/usr/sbin/nologin
systemd-timesync:x:103:106:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
syslog:x:104:111:/:home/syslog:/usr/sbin/nologin
_apt:x:105:65534:/:nonexistent:/usr/sbin/nologin
tss:x:106:113:TPM software stack,,,:/var/lib/tpm:/bin/false
uidd:x:107:116:/:run/uidd:/usr/sbin/nologin
systemd-oom:x:108:117:systemd Userspace OOM Killer,,,:/run/systemd:/usr/sbin/nologin
tcpdump:x:109:118:/:nonexistent:/usr/sbin/nologin
avahi-autoipd:x:110:119:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/usr/sbin/nologin
usbmux:x:111:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
dnsmasq:x:112:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
kernoops:x:113:65534:Kernel Oops Tracking Daemon,,,:/usr/sbin/nologin
avahi:x:114:121:Avahi mDNS daemon,,,:/run/avahi-daemon:/usr/sbin/nologin
cups-pk-helper:x:115:122:user for cups-pk-helper service,,,:/home/cups-pk-helper:/usr/sbin/nologin
rtkit:x:116:123:RealtimeKit,,,:/proc:/usr/sbin/nologin
whoopsie:x:117:124:/:nonexistent:/bin/false
sssd:x:118:125:SSSD system user,,,:/var/lib/sss:/usr/sbin/nologin
speech-dispatcher:x:119:29:Speech Dispatcher,,,:/run/speech-dispatcher:/bin/false
fwupd-refresh:x:120:126:fwupd-refresh user,,,:/run/systemd:/usr/sbin/nologin
nm-openvpn:x:121:127:NetworkManager OpenVPN,,,:/var/lib/openvpn/chroot:/usr/sbin/nologin
saned:x:122:129:/:/var/lib/saned:/usr/sbin/nologin
colord:x:123:130:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
geoclue:x:124:131:/:/var/lib/geoclue:/usr/sbin/nologin
pulse:x:125:132:PulseAudio daemon,,,:/run/pulse:/usr/sbin/nologin
gnome-initial-setup:x:126:65534:/:run/gnome-initial-setup:/bin/false
hplip:x:127:7:HPLIP system user,,,:/run/hplip:/bin/false
gdm:x:128:134:Gnome Display Manager:/var/lib/gdm3:/bin/false
asko:x:1000:1000:asko,,,:/home/asko:/bin/bash
vboxadd:x:999:1:/:var/run/vboxadd:/bin/false
sshd:x:129:65534:/:run/sshd:/usr/sbin/nologin
meowuser:x:1001:1001:meow,meow,,,:/home/meowuser:/bin/bash
asko@LXSYNC-000WAVE:~$
```

вывод файла sudoers:

```
asko@LXSYNC-000WAVE:~$ sudo grep -v '^#' /etc/sudoers
[sudo] password for asko:
Defaults                env_reset
Defaults                mail_badpass
Defaults                secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin"
Defaults                use_pty

root    ALL=(ALL:ALL) ALL
meowuser ALL=(ALL:ALL) NOPASSWD: /sbin/route,/sbin/iptables,/sbin/ifup,/sbin/ifdown,/usr/bin/nmap,/usr/sbin/hping3,/usr/bin/systemctl
%admin  ALL=(ALL) ALL

%sudo   ALL=(ALL:ALL) ALL

@includedir /etc/sudoers.d
asko@LXSYNC-000WAVE:~$
```

Для вывода содержимого файла **/etc/sudoers** используем команду **grep** в сочетании с регулярным выражением для поиска строк, не начинающихся с символа **#**, который используется для комментариев в файле **sudoers**.

grep -v '^#' /etc/sudoers

Эта команда выводит содержимое файла **/etc/sudoers**, исключая строки, которые начинаются с символа **#**, таким образом, игнорируя комментарии. Опция **-v** в команде **grep** означает "выводить только строки, не соответствующие регулярному выражению".

3. Установить минимальную длину пароля для пользователя в 8 символов.

Для установки минимальной длины пароля в 8 символов воспользуемся инструментом Pluggable Authentication Modules (PAM). Отредактируем файл настроек PAM для паролей, используя команду:

```
asko@LXSYNC-000WAVE: ~  
asko@LXSYNC-000WAVE:~$ sudo nano /etc/pam.d/common-password
```

в строке «**password requisite pam_pwquality.so retry=3**», добавим опцию «**minlen=8**»

```
# here are the per-package modules (the "Primary" block)  
password      requisite      pam_pwquality.so retry=3 minlen=8
```

"**pam_pwquality.so**" — это имя модуля **PAM**, который отвечает за проверку качества паролей. •

"**retry=3**" — это опция, указывающая количество разрешенных попыток ввода неправильного пароля перед блокировкой учетной записи. В данном случае значение равно 3, то есть пользователь может попытаться ввести неправильный пароль три раза, после чего учетная запись будет заблокирована, если пароль все еще неправильный.

Сохраняем изменения и закрываем файл. Для применения изменений необходимо перезагрузить систему или выполнить команду:

```
asko@LXSYNC-000WAVE: ~  
asko@LXSYNC-000WAVE:~$ sudo systemctl restart system-logind.service
```

При следующей смене пароля пользователю будет требоваться указать пароль, состоящий минимум из 8 символов.

Содержимое файла **common-passwords**:

```
asko@LXSYNC-000WAVE: ~  
asko@LXSYNC-000WAVE:~$ sudo grep -v '^#' /etc/pam.d/common-password  
[sudo] password for asko:  
  
password      requisite      pam_pwquality.so retry=3 minlen=8  
password      [success=2 default=ignore] pam_unix.so obscure use_authtok  
try_first_pass yescrypt  
password      sufficient    pam_sss.so use_authtok  
password      requisite     pam_deny.so  
password      required      pam_permit.so  
password      optional      pam_gnome_keyring.so  
asko@LXSYNC-000WAVE:~$
```

4. Установить пакеты Java

Для установки **Java Runtime Environment (JRE)** выполним следующую команду:

```
asko@LXSYNC-000WAVE: ~  
asko@LXSYNC-000WAVE:~$ sudo apt install default-jre
```

При этом будет установлена последняя версия JRE, доступная в официальных репозиториях **Ubuntu**. После установки можно проверить, что **JRE** установлена корректно, выполнив следующую команду:

```
asko@LXSYNC-000WAVE:~$ java -version  
openjdk version "11.0.20" 2023-07-18  
OpenJDK Runtime Environment (build 11.0.20+8-post-Ubuntu-1ubuntu122.04)  
OpenJDK 64-Bit Server VM (build 11.0.20+8-post-Ubuntu-1ubuntu122.04, mixed mode,  
sharing)  
asko@LXSYNC-000WAVE:~$
```

Результат успешной установки Java (последняя доступная версия JRE)

Java Runtime Environment (JRE) предназначен для выполнения, а не для разработки **Java**-приложений. Если требуется возможность разрабатывать **Java-приложения**, нам также необходимо установить **Java Development Kit (JDK)**.

5. Настроить автоматическое сканирование антивирусом всей ОС каждый понедельник в 4 утра. При этом раз в месяц должно происходить обновление базы данных антивирусов.

В качестве антивирусного программного обеспечения мы будем использовать **ClamAV**, одна из популярных антивирусных программ для **Linux**. Выполним следующую команду, чтобы установить пакеты, связанные с этим антивирусом:

```
asko@LXSYNC-000WAVE:~$ sudo apt install clamav clamav-daemon clamav-freshclam
```

clamav — это основной пакет **ClamAV**, который содержит само антивирусное программное обеспечение, его исполняемые файлы и библиотеки

clamav-daemon - этот пакет содержит службу демона **ClamAV**, которая позволяет запускать **ClamAV** в фоновом режиме как системную службу. Демон **ClamAV** отвечает за сканирование файлов и директорий на предмет наличия вредоносного кода в фоновом режиме, без необходимости запуска антивирусной проверки вручную.

clamav-freshclam - этот пакет содержит инструменты для обновления базы данных вирусных определений **ClamAV**.

Проверим статус службы командой:

```
asko@LXSYNC-000WAVE:~$ sudo systemctl status clamav-freshclam
```

```
asko@LXSYNC-000WAVE: ~  
● clamav-freshclam.service - ClamAV virus database updater  
   Loaded: loaded (/lib/systemd/system/clamav-freshclam.service; enabled; vendor preset: enabled)  
   Active: active (running) since Mon 2023-08-14 17:56:48 MSK; 42s ago  
     Docs: man:freshclam(1)  
           man:freshclam.conf(5)  
           https://docs.clamav.net/  
  Main PID: 4453 (freshclam)  
    Tasks: 2 (limit: 4592)  
   Memory: 578.9M  
      CPU: 16.604s  
   CGroup: /system.slice/clamav-freshclam.service  
           └─4453 /usr/bin/freshclam -d --foreground=true  
             5014 /usr/bin/freshclam -d --foreground=true
```

Для обновления антивирусной базы выполним команды:

```
asko@LXSYNC-000WAVE: ~
asko@LXSYNC-000WAVE:~$ sudo systemctl stop clamav-freshclam
asko@LXSYNC-000WAVE:~$ sudo freshclam
Mon Aug 14 18:01:31 2023 -> ClamAV update process started at Mon Aug 14 18:01:31 2023
Mon Aug 14 18:01:31 2023 -> daily.cvd database is up-to-date (version: 27000, sigs: 2039863, f-level: 90, builder: raynman)
Mon Aug 14 18:01:31 2023 -> main.cvd database is up-to-date (version: 62, sigs: 6647427, f-level: 90, builder: sigmgr)
Mon Aug 14 18:01:31 2023 -> bytecode.cvd database is up-to-date (version: 334, sigs: 91, f-level: 90, builder: anvilleg)
asko@LXSYNC-000WAVE:~$ sudo systemctl start clamav-freshclam
asko@LXSYNC-000WAVE:~$
```

проверим версию ClamAV командой:

```
asko@LXSYNC-000WAVE:~$ clamscan -V
ClamAV 0.103.8/27000/Mon Aug 14 10:37:02 2023
asko@LXSYNC-000WAVE:~$
```

ClamAV — это название антивирусной программы

0.103.8 — это конкретная версия программы.

27000 — это количество сигнатур вредоносных программ в базе данных, используемой **ClamAV**

С помощью текстового редактора Nano отредактируем конфигурационный файл, который находится в **/etc/clamav/clamd.conf**

```
asko@LXSYNC-000WAVE: ~
asko@LXSYNC-000WAVE:~$ sudo nano /etc/clamav/clamd.conf
```

Установим список путей, которые нам необходимо исключить из сканирования **ClamAV**. Начиная с версии **ClamAV 0.103.0**, исключение каталогов из сканирования теперь осуществляется с использованием параметра **"ExcludePath"**:

```
GNU nano 6.2 /etc/clamav/clamd.conf *
MaxRecursion 16
MaxFiles 10000
MaxPartitions 50
MaxIconsPE 100
PCREMatchLimit 10000
PCRERecMatchLimit 5000
PCREMaxFileSize 25M
ScanXMLDOCS true
ScanHWP3 true
MaxRechWP3 16
StreamMaxLength 25M
LogFile /var/log/clamav/clamav.log
LogTime true
LogFileUnlock false
LogFileMaxSize 0
Bytecode true
BytecodeSecurity TrustSigned
BytecodeTimeout 60000
OnAccessMaxFileSize 5M

ExcludePath /tmp /var/cache /var/tmp
```


При настройке параметров **ClamAV** также важно учитывать ресурсозатраты, производительность и требования к безопасности, поэтому изменение других настроек в конфигурационном файле **ClamAV** может варьироваться.

Сохраним и закроем конфигурационный файл **clamd.conf**. Далее, перезапустим службу **ClamAV**, чтобы изменения вступили в силу:

```
asko@LXSYNC-000WAVE: ~  
asko@LXSYNC-000WAVE:~$ sudo systemctl restart clamav-daemon  
[sudo] password for asko:  
asko@LXSYNC-000WAVE:~$
```

Создаём задачу в cron (sudo crontab -e) запуск автоматического сканирования и обновления базы данных антивируса:

```
GNU nano 6.2 /tmp/crontab.nkQZHs/crontab  
#  
# To define the time you can provide concrete values for  
# minute (m), hour (h), day of month (dom), month (mon),  
# and day of week (dow) or use '*' in these fields (for 'any').  
#  
# Notice that tasks will be started based on the cron's system  
# daemon's notion of time and timezones.  
#  
# Output of the crontab jobs (including errors) is sent through  
# email to the user the crontab file belongs to (unless redirected).  
#  
# For example, you can run a backup of all your user accounts  
# at 5 a.m every week with:  
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/  
#  
# For more information see the manual pages of crontab(5) and cron(8)  
#  
# m h dom mon dow command  
0 4 * * 1 /usr/bin/clamscan -r / # ClamAV - fullscan  
0 0 1 * * /usr/bin/freshclam # ClamAV - database update
```

автоматическое сканирование каждый понедельник в 4 утра и обновление базы данных антивируса раз в месяц

0 4 * * 1 - выполнить команду «/usr/bin/clamscan -r /» о -r / - рекурсивное сканирование, ClamAV будет сканировать все файлы и подкаталоги в корневом каталоге "/"

0 - минуты (в 0 минут)

4 - часы (в 4 утра)

*** *** - дни месяца и месяцы (звездочка означает "любое значение")

1 - день недели (1 означает понедельник)

0 0 1 * * - выполнить команду «/usr/bin/freshclam»

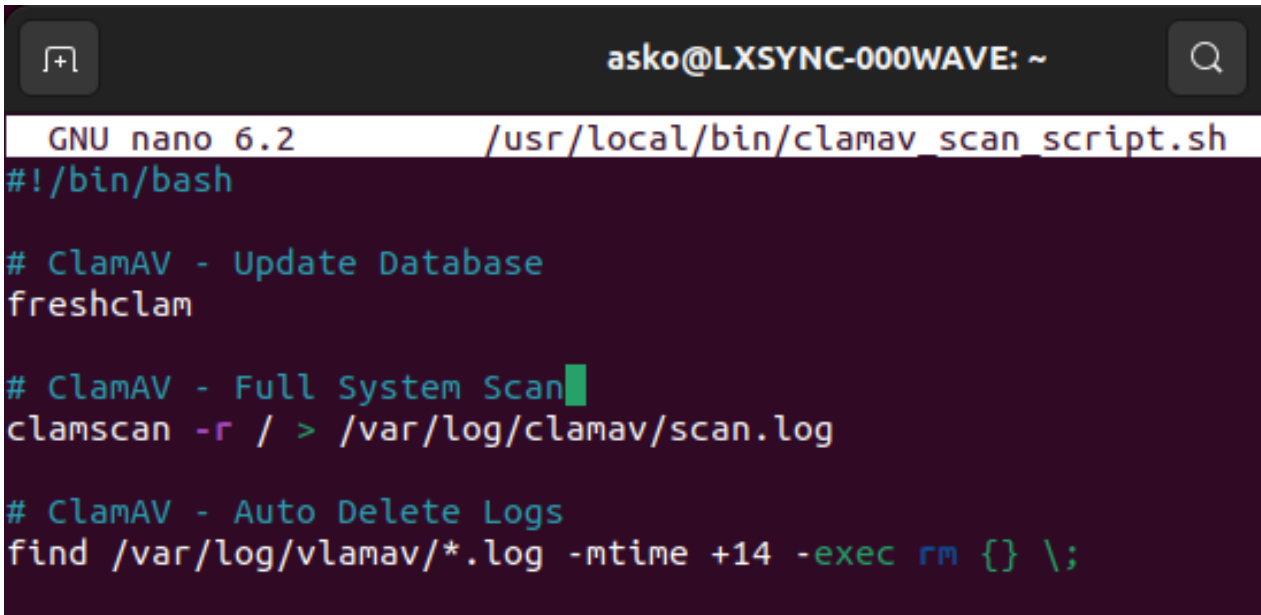
0 - минуты 0 0 - часы (полночь)

1 - дни месяца (в данном случае, первое число месяца)

*** *** - месяцы и дни недели (звездочка означает "любое значение")

Также можно создать простой Bash-скрипт и добавить запуск в cron:

```
asko@LXSYNC-000WAVE:~$ sudo nano /usr/local/bin/clamav_scan_script.sh
```



The screenshot shows a terminal window with the title bar "asko@LXSYNC-000WAVE: ~". The terminal content shows the nano 6.2 editor editing the file "/usr/local/bin/clamav_scan_script.sh". The script content is as follows:

```
#!/bin/bash

# ClamAV - Update Database
freshclam

# ClamAV - Full System Scan
clamscan -r / > /var/log/clamav/scan.log

# ClamAV - Auto Delete Logs
find /var/log/vlamav/*.log -mtime +14 -exec rm {} \;
```

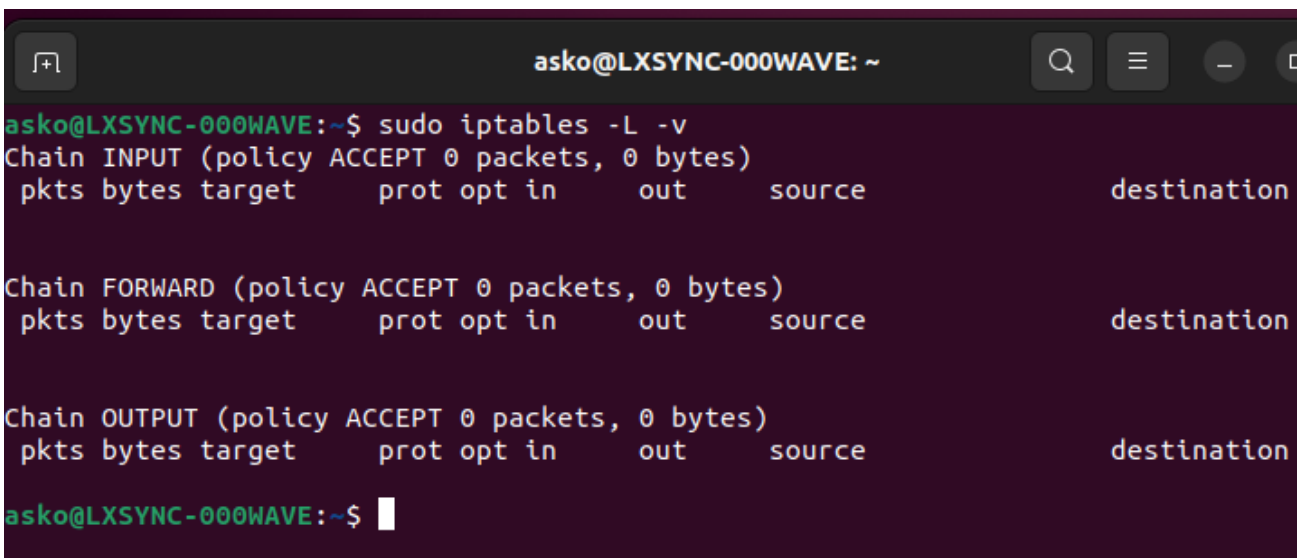
Сохраняем и закрываем редактор. Переходим в директорию, где находится созданный файл скрипта.

Выполним команду «**chmod +x clamav_scan_script.sh**», чтобы дать права на выполнение скрипта.

В **cron** добавим: **0 4 * * 1 /usr/local/bin/clamav_scan_script.sh**

6. Настроить фаерволл на блокирование всего входящего и исходящего трафика.

Выполним команду для вывода всех **цепочек(chains)** и **правил(rules)** в **таблицах фильтрации(filter)**:



The screenshot shows a terminal window with the title bar "asko@LXSYNC-000WAVE: ~". The terminal content shows the command "sudo iptables -L -v" being executed, resulting in the following output:

```
asko@LXSYNC-000WAVE:~$ sudo iptables -L -v
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source                   destination

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source                   destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source                   destination

asko@LXSYNC-000WAVE:~$
```

L - используется для вывода списка правил в **таблице фильтрации (filter)** по умолчанию

-v - режим подробного вывода

Добавляем правила для блокирования всего входящего и исходящего трафика:

```
asko@LXSYNC-000WAVE: ~  
asko@LXSYNC-000WAVE:~$ sudo iptables -L -v  
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)  
pkts bytes target      prot opt in      out     source  
  
Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)  
pkts bytes target      prot opt in      out     source  
  
Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)  
pkts bytes target      prot opt in      out     source  
  
asko@LXSYNC-000WAVE:~$ sudo iptables -P INPUT DROP  
asko@LXSYNC-000WAVE:~$ sudo iptables -P OUTPUT DROP  
asko@LXSYNC-000WAVE:~$
```

Разрешаем локальный трафик (loopback):

```
asko@LXSYNC-000WAVE: ~  
asko@LXSYNC-000WAVE:~$ sudo iptables -A INPUT -i lo -j ACCEPT  
asko@LXSYNC-000WAVE:~$ sudo iptables -A OUTPUT -o lo -j ACCEPT  
asko@LXSYNC-000WAVE:~$
```

Сохраняем настройки правил **iptables**, чтобы они пережили перезагрузку системы:

```
asko@LXSYNC-000WAVE: ~  
asko@LXSYNC-000WAVE:~$ sudo mkdir /etc/iptables && sudo touch /etc/iptables/rules.v4  
  
asko@LXSYNC-000WAVE: ~  
asko@LXSYNC-000WAVE:~$ sudo mkdir /etc/iptables && sudo touch /etc/iptables/rules.v4  
asko@LXSYNC-000WAVE:~$ sudo chown $(whoami) /etc/iptables/rules.v4  
asko@LXSYNC-000WAVE:~$ ls -l /etc/iptables/rules.v4  
-rw-r--r-- 1 asko root 0 abr 14 19:00 /etc/iptables/rules.v4  
asko@LXSYNC-000WAVE:~$ sudo iptables-save > /etc/iptables/rules.v4  
asko@LXSYNC-000WAVE:~$
```

«iptables-restore» загружает настройки iptables из файла, например: «iptables-restore < /etc/iptables/rules.v4»

Вывод всех цепочек и правил iptables:

```
asko@LXSYNC-000WAVE: ~  
asko@LXSYNC-000WAVE:~$ sudo mkdir /etc/iptables && sudo touch /etc/iptables/rules.v4  
asko@LXSYNC-000WAVE:~$ sudo chown $(whoami) /etc/iptables/rules.v4  
asko@LXSYNC-000WAVE:~$ ls -l /etc/iptables/rules.v4  
-rw-r--r-- 1 asko root 0 abr 14 19:00 /etc/iptables/rules.v4  
asko@LXSYNC-000WAVE:~$ sudo iptables-save > /etc/iptables/rules.v4  
asko@LXSYNC-000WAVE:~$ sudo iptables -L -v  
Chain INPUT (policy DROP 4 packets, 284 bytes)  
pkts bytes target      prot opt in      out     source      destination  
    64  4988 ACCEPT      all  --  lo      any     anywhere    anywhere  
     0     0 ACCEPT      all  --  lo      any     anywhere    anywhere  
  
Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)  
pkts bytes target      prot opt in      out     source      destination  
  
Chain OUTPUT (policy DROP 2048 packets, 164K bytes)  
pkts bytes target      prot opt in      out     source      destination  
    64  4988 ACCEPT      all  --  any     lo      anywhere    anywhere  
  
asko@LXSYNC-000WAVE:~$
```