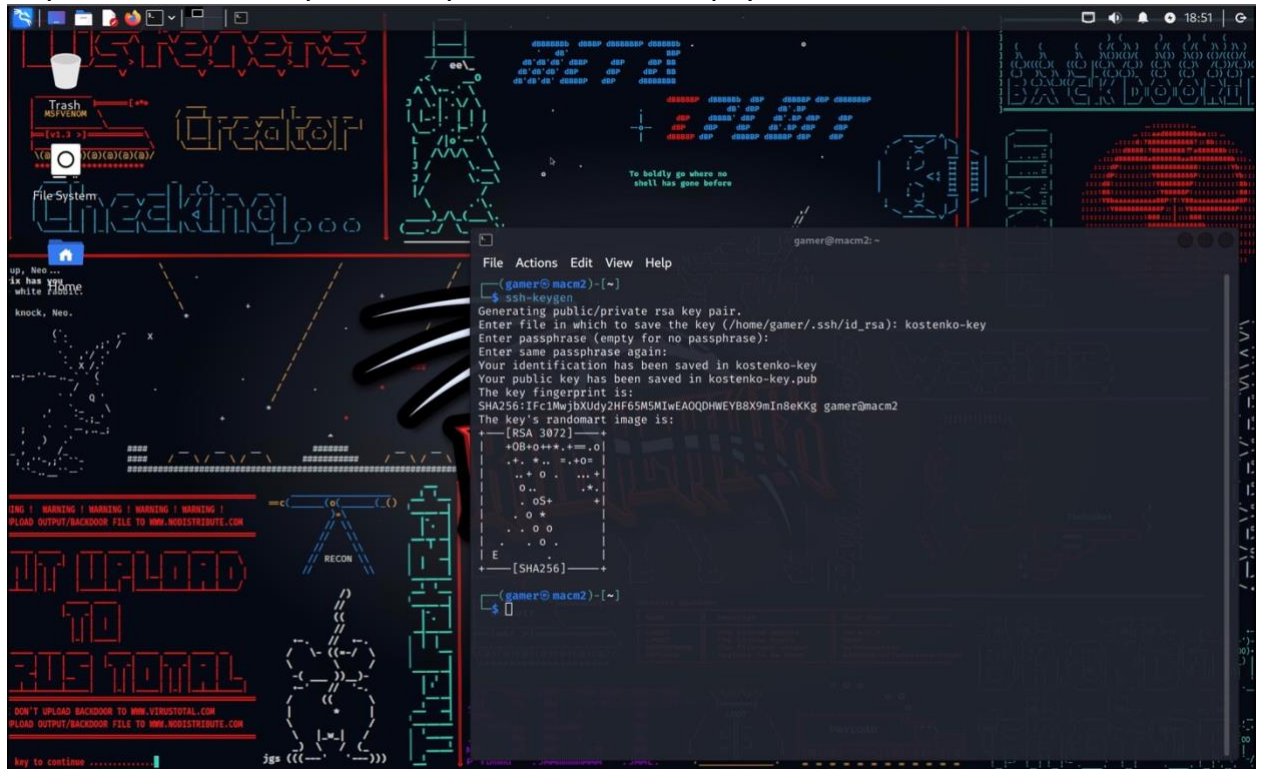


Практическое задание: безопасность ОС Linux, выполнил Костенко Евгений

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

Установка и запуск – `sudo apt install openssh-server / sudo systemctl start ssh`

Скрин ниже – генерация пары ключей на виртуальной машине kali linux



Поместил публичный ключ `./ssh/authorized_keys`

```
gamer@macm2: ~/.ssh
File Actions Edit View Help
(gamer@macm2)~[~/.ssh]
$ cat kostenko-key.pub > authorized_keys
(gamer@macm2)~[~/.ssh]
$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQgQC3H5In5gSnxkotmhWpZVh3JnP2Wqoh86eJC4LU85IgYftPu24j2wwvKfGUGMv9DeEI7MCPC
gJqRMDLYccZogZrR9TnFFlGmhuyXCJhRgJ8ugwUxnD0v6Q010e2MeXeq5Gq3DdVzL7nScwRmioKqkgnmeuN9idLDy000Pwx8+4G2xP3WptjW
XxgA25f1Hm0/WvcSg0LajEb1YNynu0lau46kbjm80hcy0ikJzy/9hAMTRKWUAGiriyHWZwB3LGstW93/1jyk5R5zKKd401DAugvtbKTGlkaS
dw0QcbqhaWLNkUx2J2L/PD0grA2Y2CX0769LEjYeIChKrUvy0VeExqEh35a6BWunC5tPDoM0ZxsNj9hLou2UJruE/c45Bz7koAlaosG7tzTRn
OPA5WNos0nkPhk8XLloawrSNrQC0M+Sk0YkCIQd7NshCTLVJzkW/kmd9MVBt3BzTj4om5rq7ZD1au5yMb7uo6iUA3g1/DBbcdlvwTF+bPqy7S
1sAAIE= gamer@macm2
(gamer@macm2)~[~/.ssh]
$
```

Подключение с указанием файла приватного ключа – `ssh`
`gamer@192.168.3.69 -i kostenko-key`
Терминал zsh хостовой машины под MacOS

```
gamer@macm2: ~/.ssh
File Actions Edit View Help
(gamer@macm2)~[~/.ssh]
$ cat kostenko-key.pub > authorized_keys
(gamer@macm2)~[~/.ssh]
$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQgQC3H5In5gSnxkotmhWpZVh3JnP2Wqoh86eJC4LU85IgYftPu24j2wwvKfGUGMv9DeEI7MCPC
gJqRMDLYccZogZrR9TnFFlGmhuyXCJhRgJ8ugwUxnD0v6Q010e2MeXeq5Gq3DdVzL7nScwRmioKqkgnmeuN9idLDy000Pwx8+4G2xP3WptjW
XxgA25f1Hm0/WvcSg0LajEb1YNynu0lau46kbjm80hcy0ikJzy/9hAMTRKWUAGiriyHWZwB3LGstW93/1jyk5R5zKKd401DAugvtbKTGlkaS
dw0QcbqhaWLNkUx2J2L/PD0grA2Y2CX0769LEjYeIChKrUvy0VeExqEh35a6BWunC5tPDoM0ZxsNj9hLou2UJruE/c45Bz7koAlaosG7tzTRn
OPA5WNos0nkPhk8XLloawrSNrQC0M+Sk0YkCIQd7NshCTLVJzkW/kmd9MVBt3BzTj4om5rq7ZD1au5yMb7uo6iUA3g1/DBbcdlvwTF+bPqy7S
1sAAIE= gamer@macm2
(gamer@macm2)~[~/.ssh]
$ sudo systemctl is-active ssh
[sudo] password for gamer:
active
(gamer@macm2)~[~/.ssh]
$
```

```
gamer@macm2: ~ -- ssh gamer@192.168.3.69 -i kostenko-key -- 110x45
64 bytes from 192.168.3.69: icmp_seq=1 ttl=64 time=1.013 ms
64 bytes from 192.168.3.69: icmp_seq=2 ttl=64 time=1.095 ms
64 bytes from 192.168.3.69: icmp_seq=3 ttl=64 time=0.866 ms
64 bytes from 192.168.3.69: icmp_seq=4 ttl=64 time=1.056 ms
64 bytes from 192.168.3.69: icmp_seq=5 ttl=64 time=1.110 ms
64 bytes from 192.168.3.69: icmp_seq=6 ttl=64 time=1.082 ms
^C
--- 192.168.3.69 ping statistics ---
7 packets transmitted, 7 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 0.866/1.018/1.110/0.090 ms
kostenko.evgeny@ig7UhhMBAM2 Downloads % ssh -i kostenko-key 192.168.3.69
The authenticity of host '192.168.3.69 (192.168.3.69)' can't be established.
ED25519 key fingerprint is SHA256:WFHIZeSPCVi0mtUqDL5znLsWxOKDs13enB8/pok3P2Q.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '192.168.3.69' (ED25519) to the list of known hosts.
kostenko.evgeny@192.168.3.69's password:
Permission denied, please try again.
kostenko.evgeny@192.168.3.69's password:
Permission denied, please try again.
kostenko.evgeny@192.168.3.69's password:
zsh: suspended ssh -i kostenko-key 192.168.3.69
kostenko.evgeny@ig7UhhMBAM2 Downloads % ssh -help
ssh: illegal option -- h
usage: ssh [-46AacfgGkMnNqsTtVvXxyY] [-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 ...]]
kostenko.evgeny@ig7UhhMBAM2 Downloads % ssh gamer@192.168.3.69 -i kostenko-key
Enter passphrase for key 'kostenko-key':
Linux macm2 6.3.0-kali1-azm64 #1 SMP Debian 6.3.7-1kali1 (2023-06-29) aarch64

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

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
(gamer@macm2)~[~]
```

Содержимое sshd по умолчанию ssh сервер после запуска «из коробки» уже настроен на подключение по ключам на порту 22 расположенным в директории `./ssh/authorized_keys` тем не менее я прописал основные настройки вручную в файл, хотя этого не требуется в большинстве современных дистрибутивов – активные настройки выделены белым цветом на скрине ниже

```
GNU nano 7.2 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/bin:/usr/bin:/bin:/usr/games

# 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 22
#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

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
#PermitRootLogin prohibit-password
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
AuthorizedKeysFile .ssh/authorized_keys

#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication no
#PermitEmptyPasswords no

# Change to yes to enable challenge-response passwords (beware issues with
[ Wrote 122 lines ]

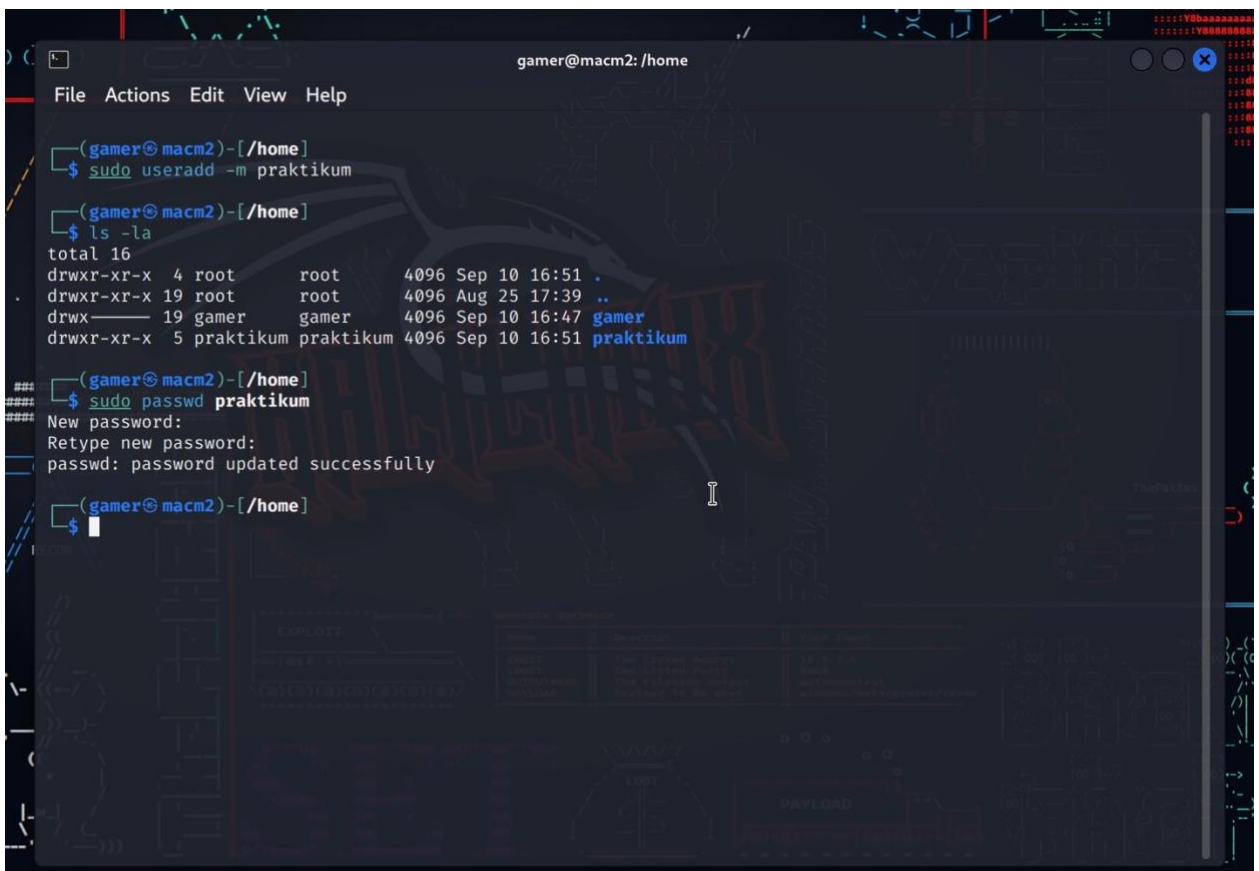
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line  M-E Redo
```


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

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

Пользователя можно создать командой `adduser` или `useradd` – более простой и автоматизированный способ `adduser`

`sudo useradd -m praktikum` – создал пользователя с домашней директорией
`sudo passwd praktikum` – добавил пароль
на скрине ниже видно содержимое папки `/home`



```
gamer@macm2: /home
File Actions Edit View Help
(gamer@macm2)-[/home]
$ sudo useradd -m praktikum
(gamer@macm2)-[/home]
$ ls -la
total 16
drwxr-xr-x  4 root    root    4096 Sep 10 16:51 .
drwxr-xr-x 19 root    root    4096 Aug 25 17:39 ..
drwx----- 19 gamer   gamer   4096 Sep 10 16:47 gamer
drwxr-xr-x  5 praktikum praktikum 4096 Sep 10 16:51 praktikum
(gamer@macm2)-[/home]
$ sudo passwd praktikum
New password:
Retype new password:
passwd: password updated successfully
(gamer@macm2)-[/home]
$
```

`sudo visudo` – добавляю эту строку в файл `sudoers`

praktikum

ALL=NOPASSWORD:/sbin/route,/sbin/iptables,/usr/bin/nmap,/usr/sbin/hping3,/usr/bin/systemctl,/sbin/ifup,/sbin/ifdown

The screenshot shows a terminal window with two overlapping windows. The background window displays the output of the following commands:

```
$ whoami
praktikum
$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target    prot opt source               destination
Chain FORWARD (policy ACCEPT)
target    prot opt source               destination
Chain OUTPUT (policy ACCEPT)
target    prot opt source               destination
$
```

The foreground window shows the contents of the `/etc/sudoers` file, edited with `nano`. The file content is as follows:

```
GNU nano 7.2
# This allows running arbitrary commands, but so does
# different sudoers have their choice of editor resp
#Defaults:%sudo env_keep += "EDITOR"

# Completely harmless preservation of a user preference
#Defaults:%sudo env_keep += "GREP_COLOR"

# While you shouldn't normally run git as root, you
#Defaults:%sudo env_keep += "GIT_AUTHOR_* GIT_COMMIT"

# Per-user preferences; root won't have sensible values
#Defaults:%sudo env_keep += "EMAIL DEBEMAIL DEBFULLNAME"

# "sudo scp" or "sudo rsync" should be able to use y
#Defaults:%sudo env_keep += "SSH_AGENT_PID SSH_AUTH_SSH"

# Ditto for GPG agent
#Defaults:%sudo env_keep += "GPG_AGENT_INFO"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL
praktikum ALL=NOPASSWD:/sbin/route,/sbin/iptables,/usr/bin/nmap,/usr/sbin/hping3,/usr/bin/systemctl,/sbin/ifup,/sbin/ifdown
# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL
```

На скринах видно содержимое файла `sudoers` и выполнение нашим новым юзером нужной команды без пароля

Вывод файла `passwd`

```

(gamer@macm2)-[/etc]
$ cat passwd
root:x:0:0:root:/root:/usr/bin/zsh
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:Mail List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
_apt:x:42:65534::/nonexistent:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:998:998:systemd Network Management:/:/usr/sbin/nologin
mysql:x:100:107:MySQL Server,,,:/nonexistent:/bin/false
tss:x:101:108:TPM software stack,,,:/var/lib/tpm:/bin/false
strongswan:x:102:65534::/var/lib/strongswan:/usr/sbin/nologin
systemd-timesync:x:997:997:systemd Time Synchronization:/:/usr/sbin/nologin
redsocks:x:103:109::/var/run/redsocks:/usr/sbin/nologin
rwhod:x:104:65534::/var/spool/rwho:/usr/sbin/nologin
_gophish:x:105:111::/var/lib/gophish:/usr/sbin/nologin
iodine:x:106:65534::/run/iodine:/usr/sbin/nologin
messagebus:x:107:112::/nonexistent:/usr/sbin/nologin
miredo:x:108:65534::/var/run/miredo:/usr/sbin/nologin
redis:x:109:115::/var/lib/redis:/usr/sbin/nologin
usbmux:x:110:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
mosquitto:x:111:117::/var/lib/mosquitto:/usr/sbin/nologin
tcpdump:x:112:119::/nonexistent:/usr/sbin/nologin
sshd:x:113:65534::/run/sshd:/usr/sbin/nologin
_rpc:x:114:65534::/run/rpcbind:/usr/sbin/nologin
dnsmasq:x:115:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
statd:x:116:65534::/var/lib/nfs:/usr/sbin/nologin
avahi:x:117:123:Avahi mDNS daemon,,,:/run/avahi-daemon:/usr/sbin/nologin
stunnel4:x:996:996:stunnel service system account:/var/run/stunnel4:/usr/sbin/nologin
Debian-snmp:x:118:124::/var/lib/snmp:/bin/false
_gvm:x:119:125::/var/lib/opensvas:/usr/sbin/nologin
speech-dispatcher:x:120:29:Speech Dispatcher,,,:/run/speech-dispatcher:/bin/false
sslh:x:121:126::/nonexistent:/usr/sbin/nologin
postgres:x:122:127:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
pulse:x:123:129:PulseAudio daemon,,,:/run/pulse:/usr/sbin/nologin
inetsim:x:124:131::/var/lib/inetsim:/usr/sbin/nologin
lightdm:x:125:132:Light Display Manager:/var/lib/lightdm:/bin/false
geoclue:x:126:133::/var/lib/geoclue:/usr/sbin/nologin
saned:x:127:135::/var/lib/saned:/usr/sbin/nologin
polkitd:x:994:994:polkit:/nonexistent:/usr/sbin/nologin
rtkit:x:128:136:RealtimeKit,,,:/proc:/usr/sbin/nologin
colord:x:129:137:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
nm-openvpn:x:130:138:NetworkManager OpenVPN,,,:/var/lib/openvpn/chroot:/usr/sbin/nologin
nm-openconnect:x:131:139:NetworkManager OpenConnect plugin,,,:/var/lib/NetworkManager:/usr/sbin/nologin
gamer:x:1000:1000:gamer,,,:/home/gamer:/usr/bin/zsh
praktikum:x:1002:1002::/home/praktikum:/bin/sh

```

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

Добавил minlen=8 в параметр password requisite


```
gamer@macm2: /etc/pam.d
File Actions Edit View Help
-rw-r--r-- 1 root root 84 Aug 4 00:45 samba
-rw-r--r-- 1 root root 2133 Jul 19 17:49 sshd
-rw-r--r-- 1 root root 2259 Aug 11 12:52 su
-rw-r--r-- 1 root root 137 Aug 11 12:52 su-l
-rw-r--r-- 1 root root 185 Jul 19 18:31 sudo
-rw-r--r-- 1 root root 170 Jul 19 18:31 sudo-i

(gamer@macm2)-[/etc/pam.d]
$ sudo nano common-password

(gamer@macm2)-[/etc/pam.d]
$ cat common-password
#
# /etc/pam.d/common-password - password-related modules common to all services
#
# This file is included from other service-specific PAM config files,
# and should contain a list of modules that define the services to be
# used to change user passwords. The default is pam_unix.

# Explanation of pam_unix options:
# The "yescrypt" option enables
# hashed passwords using the yescrypt algorithm, introduced in Debian
# 11. Without this option, the default is Unix crypt. Prior releases
# used the option "sha512"; if a shadow password hash will be shared
# between Debian 11 and older releases replace "yescrypt" with "sha512"
# for compatibility. The "obscure" option replaces the old
# 'OBSOLETE_CHECKS_ENAB' option in login.defs. See the pam_unix manpage
# for other options.

# As of pam 1.0.1-6, this file is managed by pam-auth-update by default.
# To take advantage of this, it is recommended that you configure any
# local modules either before or after the default block, and use
# pam-auth-update to manage selection of other modules. See
# pam-auth-update(8) for details.

# here are the per-package modules (the "Primary" block)
password [success=1 default=ignore] pam_unix.so obscure yescrypt
# here's the fallback if no module succeeds
password requisite pam_deny.so minlen=8
# prime the stack with a positive return value if there isn't one already;
# this avoids us returning an error just because nothing sets a success code
# since the modules above will each just jump around
password required pam_permit.so
# and here are more per-package modules (the "Additional" block)
password optional pam_gnome_keyring.so
# end of pam-auth-update config

(gamer@macm2)-[/etc/pam.d]
$
```

4. Установить на сервер пакеты Java.

sudo apt install default-jdk

```
(gamer@macm2)-[/etc/pam.d]
$ java -version
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
openjdk version "17.0.8" 2023-07-18
OpenJDK Runtime Environment (build 17.0.8+7-Debian-1)
OpenJDK 64-Bit Server VM (build 17.0.8+7-Debian-1, mixed mode, sharing)

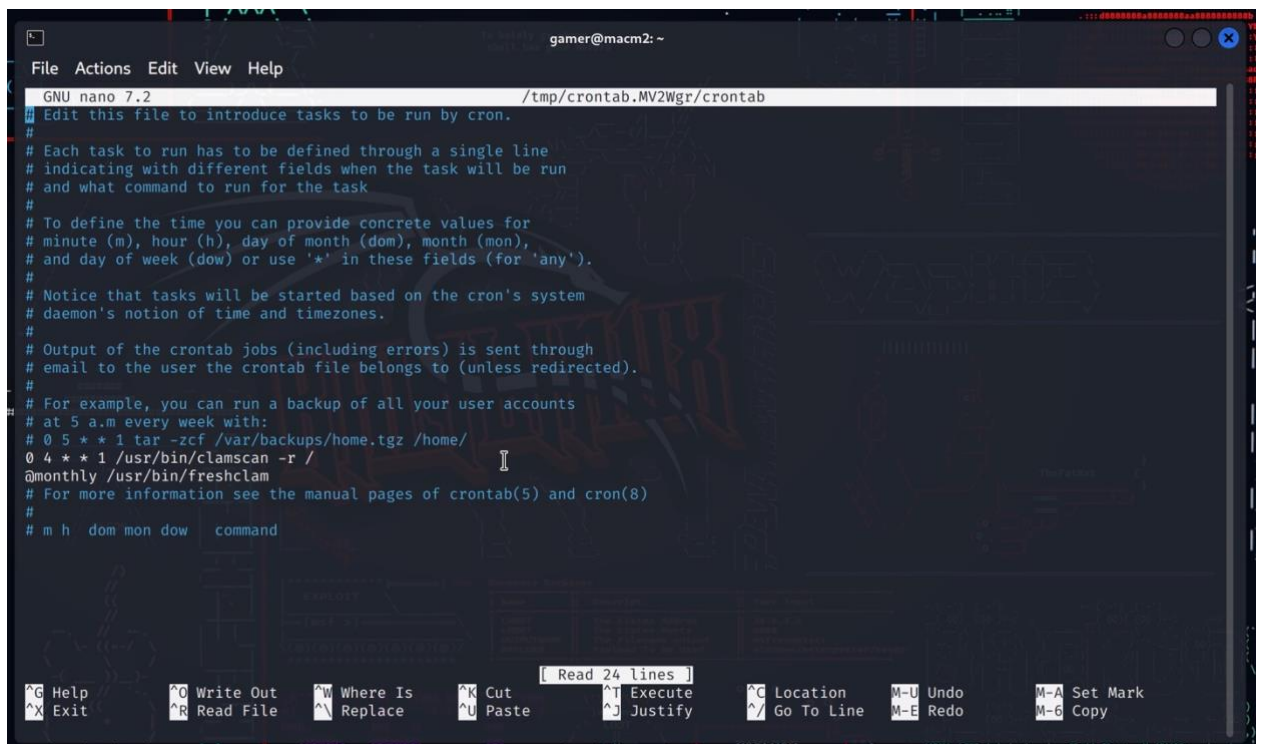
(gamer@macm2)-[/etc/pam.d]
$
```

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

Открываю редактор кронтаб `sudo crontab -e`

и прописываю задачи `0 4 * * 1 /usr/bin/clamscan -r /` - сканирование в 4 утра каждый понедельник

`@monthly /usr/bin/freshclam` – ежемесячно обновлять базу



```
GNU nano 7.2 /tmp/crontab.MV2Wgr/crontab
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# 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/
0 4 * * 1 /usr/bin/clamscan -r /
@monthly /usr/bin/freshclam
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
```

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

Устанавливаю правила `sudo iptables -P INPUT DROP` и `iptables -P OUTPUT DROP`


```
gamer@macm2: ~  
File Actions Edit View Help  
(gamer@macm2)-[~]  
$ sudo iptables -P INPUT DROP  
[sudo] password for gamer:  
(gamer@macm2)-[~]  
$ sudo iptables -P INPUT OUTPUT DROP  
Bad argument 'DROP'  
Try 'iptables -h' or 'iptables --help' for more information.  
(gamer@macm2)-[~]  
$ sudo iptables -P OUTPUT DROP  
(gamer@macm2)-[~]  
$ clear
```

Устанавливаю применение правил после перезагрузки

sudo apt install iptables-persistent

