Учебный курс

Безопасность сетей на базе Linux (Unix)

(лабораторные работы к курсу SecurL с ответами)

Автор: Лесковец В.В.

Версия: 2.0

г. Екатеринбург
2025

Оглавление

Глава 1. Введение.

1.1 Подготовка виртуалных машин.

1. Предполагается выполнение лабораторных работ на виртуальных машинах в среде VirtualBox или QEMU/KVM (libvirt).
2. Хостовая ОС может быть любой: Linux, Windows и пр.
3. Требования к хостовой машине:
    1. ОЗУ — от 16Гб.
    2. Не менее 6 ядер.
    3. Диск (SSD) — от 100Гб свободного места.
    4. Подготовленный шаблон виртуальных с установленной ОС Debian 12. Установлена по умолчанию. ОЗУ — 2Гб., 2 ядра, диск — 40Гб., каталог /home расположен на отдельном разделе, размером от 5Гб. Сеть назначенная для шаблона должна иметь выход в интернет и допускать взаимодействие ВМ между собой и с хостом.
4. Сделайте клон виртуальной машины с именем SRV1. В новой ВМ увеличьте количество памяти до 4Гб. В виртуальной машине должно быть создано два пользователя с именами sa и user. Пароль — lin123. Пользователь sa добавлен в группу sudo.
5. Запустите ВМ SRV1 и настройте имя гостевой машины на srv1. Не забудьте прописать новое имя в файле `/etc/hosts`.
6. Сделайте еще 2 клона виртуальной машины с именем CL1 и CL2.
7. Запустите ВМ CL1 и CL2 и настройте имя гостевой машины на cl1и cl2. Не забудьте прописать новые имена в файле `/etc/hosts`.

Глава 2. Обеспечение физической безопасности Linux-сервера.

2.1 Работа с ИБП.

Если имеется ИБП, то настройте управление им в вашей ОС. Данные о подключении к ИБП получите от преподавателя.

1. Установите пакеты nut и nut-client

```
$ sudo apt install nut nut-client
```

2. Установите режим работы NUT в netclient

```
# grep MODE= /etc/nut/nut.conf
MODE=netclient
```

3. Настройте мониторинг ИБП

```
# grep ^MONITOR /etc/nut/upsmon.conf
MONITOR powercom@172.27.255.26 1 upsmonuser lin123 secondary
```

4. Перезапустите службу nut-monitor.service проверьте ее состояние.

```
# systemctl restart nut-monitor.service
# systemctl status nut-monitor.service
```

5. Проверьте какие у ИБП: нагрузка, процент заряда, модель, общее состояние.

```
# upsc powercom@172.27.255.26
```

2.2 Настройка и управление загрузкой

1. Настройте пароль для изменения настроек в GRUB во время загрузки.

Создать хэш пароля

```
# grub-mkpasswd-pbkdf2
Введите пароль:
Повторно введите пароль:
Хэш PBKDF2 вашего пароля:
grub.pbkdf2.sha512.10000.BC259FB35201B07343A9FC3E917E8EA9A577BE0DB929CC9B1BC052A
9104E7F84A49AC614040C331B4DE710F3D33115F171C2129DA3EF75544E57924198AF0C61.CA042C
5A4D9596D43768E5D0B6C01670B160099608B9604B3EA13B83861DE959C20169CDC23DDCD3BD5B1F
1E9270590F7ACFB12EDEE2B6E357677FF35BD5437E
```

Далее внести строки для настройки пользователя и пароля в файл `/etc/grub.d/40_custom`:

```
#!/bin/sh
exec tail -n +3 $0
# This file provides an easy way to add custom menu entries.  Simply type the
# menu entries you want to add after this comment.  Be careful not to change
# the 'exec tail' line above.
set superusers="grub"
password_pbkdf2 grub grub.pbkdf2.sha512.10000.BC259FB35201B07343A9FC3E917E8EA9A5
77BE0DB929CC9B1BC052A9104E7F84A49AC614040C331B4DE710F3D33115F171C2129DA3EF75544E
57924198AF0C61.CA042C5A4D9596D43768E5D0B6C01670B160099608B9604B3EA13B83861DE959C
20169CDC23DDCD3BD5B1F1E9270590F7ACFB12EDEE2B6E357677FF35BD5437E
```

Одна строка: `password_pbkdf2 grub grub<..>437E`

В файле `/etc/grub.d/10_linux` изменить строку:

```
CLASS="--class gnu-linux --class gnu --class os"
```

на

```
CLASS="--class gnu-linux --class gnu --class os --unrestricted"
```

Обновить конфигурацию загрузчика

```
# update-grub
```

2. Опробуйте как работает пароль в GRUB.

Проверьте, что при простой загрузке пароль не спрашивает, но при попытке отредактировать пункт меню загрузки требует учетные данные.

3. Проведите процедуру сброса пароля.

Во время старта компьютера, в загрузчике, в строке выбора ОС нажмите клавишу «e».

Найдите строку, которая описывает какое ядро и с какими параметрами загружается. В конце строки напишите параметр `init=/bin/bash`. Затем нажмите `Ctrl-x`.

Командой `passwd` изменить пароль суперпользователю.

Выполнить команды `sync` и `mount -o remount,ro /`

Перезапустить систему.

2.3 Шифрование диска

На CL1:

1. Добавьте в виртуальную машину дисковый контроллер USB и подключите к нему диск.
2. Если пакет cryptsetup не установлен, то установите его.

```
# apt list cryptsetup
```

3. Определите что вы будете шифровать, это целый USB диск, поэтому нужно определить его имя

```
$ ls -l /sys/block/ | grep usb
lrwxrwxrwx 1 root root 0 фев 10 17:36 sdb ->
../devices/pci0000:00/0000:00:14.0/usb1/1-2/1-2:1.0/host3/target3:0:0/3:0:0:0/
block/sdb
```

4. Подготовка диска — заполнение диска случайными данными. Этот этап можно пропустить, но тогда данные будут уязвимыми.

```
# shred -v --iterations=1 /dev/sdb
```

5. Утилитой `cryptsetup` определите параметры шифрования.

```
# cryptsetup --cipher=aes-cbc-essiv:sha256 --key-size 256 \
luksFormat /dev/sdb

ПРЕДУПРЕЖДЕНИЕ!
========
Данные на /dev/sdb будут перезаписаны без возможности восстановления.

Вы уверены? (введите «yes» заглавными буквами): YES
Введите парольную фразу для /dev/sdb:
Парольная фраза повторно:
```

6. Создайте виртуальное устройство для шифрованного диска

```
# cryptsetup open /dev/sdb encsdb
Введите парольную фразу для /dev/sdb:

root@srv1:~# ls -l /dev/mapper/
итого 0
crw------- 1 root root 10, 236 фев  3 12:07 control
lrwxrwxrwx 1 root root       7 фев  3 12:16 encsdb -> ../dm-0
```

7. Создайте файловую систему

```
# mkfs.ext4 /dev/mapper/encsdb
```

8. Смонтируйте файловую систему и создайте в ней файл с именем `essential_data.txt` и строкой «Важные данные» внутри файла.

```
# mount /dev/mapper/encsdb /mnt/
# echo Важные данные > /mnt/essential_data.txt
```

9. Размонтируйте, закройте и отсоедините диск от ВМ.

```
# umount /mnt
# cryptsetup close encsdb
# ls -l /dev/mapper/
итого 0
crw------- 1 root root 10, 236 фев  3 12:07 control
```

На CL2

1. Подключите диск с шифрованными данными к CL2.
2. Если пакет cryptsetup не установлен, то установите его.

```
# apt list cryptsetup
```

3. Определите ка называется шифрованный USB диск

```
$ ls -l /sys/block/ | grep usb
lrwxrwxrwx 1 root root 0 фев 10 17:36 sdb ->
../devices/pci0000:00/0000:00:14.0/usb1/1-2/1-2:1.0/host3/target3:0:0/3:0:0:0/
block/sdb
```

4. Создайте виртуальное устройство для шифрованного диска

```
# cryptsetup open /dev/sdb encsdb
Введите парольную фразу для /dev/sdb:

root@srv1:~# ls -l /dev/mapper/
итого 0
crw------- 1 root root 10, 236 фев  3 12:07 control
lrwxrwxrwx 1 root root       7 фев  3 12:16 encsdb -> ../dm-0
```

5. Смонтируйте файловую систему и прочтите в ней файл с именем `essential_data.txt`.

```
# mount /dev/mapper/encsdb /mnt/
# cat /mnt/essential_data.txt
```

2.4 Управление внешними носителями.

1. Подключите к cl2 новый USB диск.
2. Создайте на нем файловую систему vfat и проверьте, что обычный пользователь может в графическом сеансе подключить этот диск и скопировать на него файл.

```
# mkfs.vfat /dev/sdb
```

Здесь используется диск sdb, в вашей системе он может называться по другому.

3. Создайте группу пользователей usb-users и добавьте пользователя sa в эту группу.

```
# groupadd usb-users
# gpasswd -a sa usb-users
```

4. Создайте правило UDEV, которое назначит группу usb-users на любой подключенный USB диск и только администратор сможет подключить диск.

```
# cat /etc/udev/rules.d/99zz-usbdisks.rules
ACTION=="add", SUBSYSTEM=="block", ENV{ID_USB_DRIVER}=="usb-storage",
RUN+="/usr/bin/chgrp usb-users /dev/%k", ENV{UDISKS_AUTO}="0",
ENV{UDISKS_SYSTEM}="1"
```

Одна строка

5. Перечитайте правила UDEV и проверьте как будет работать подключение USB носителя.

```
# udevadm control --reload
# udevadm test -a add /dev/sdb
```

6. Отключите и вновь подключите USB диск. Проверьте что sa может с ним работать, а user нет.
7. Проверьте, что пользователи могут открывать оптические диски. Подключите к ВМ любой iso файл.
8. В файл `/etc/modprobe.d/blockcdroms.conf` добавьте строку с блокировкой модуля sr_mod.

```
root@client:~# cat /etc/modprobe.d/blacklist.conf
blacklist sr_mod
```

9. Обновите все initrd файлы и перезагрузите машину.

```
# update-initramfs -k all -u
# reboot
```

10. Проверьте, что после перезагрузки оптические диски не открываются.

2.5 Отключение терминала VGA в процессе загрузки.

1. Выключите ВМ cl2 и добавьте в настройках ВМ COM порт №1, режим — TCP, путь/адрес — 54321, опцию подключения к существующему сокету — убрать. Запустите ВМ.
2. Подключитесь к порту 54321 на хосте по протоколу telnet.
3. Войдите на cl2 и активируйте консоль на первой последовательной линии.

```
# systemctl enable --now serial-getty@ttyS0
```

4. Убедитесь что в telnet сессии отобразилось приглашение на вход в систему.
5. Настройте перевод вывода на первое последовательное соединение.

```
root@srv1:~# grep -E 'serial|console' /etc/default/grub
GRUB_CMDLINE_LINUX_DEFAULT="quiet console=ttyS0"
GRUB_TERMINAL=serial
GRUB_SERIAL_COMMAND="serial --speed=9600 --unit=0 --word=8 --parity=no --stop=1"
```

6. Обновите файл конфигурации GRUB.

```
# grub-mkconfig -o /boot/grub/grub.cfg
```

7. Перезагрузите машину и проверьте, что теперь вывод процесса загрузки идет только на последовательную линию.

2.6 *Шифрование пользовательских данных (необязательное)

1. На CL1 проверьте, что установлен пакет cryptsetup.
2. Загрузите cl1 в однопользовательском режиме.
3. Сделайте архив пользовательских данных. Архив разместите в каталоге `/root`.
4. Настройте шифрование раздела, на котором находился каталог `/home`,
5. Опишите автоматическое подключение шифрованного раздела по паролю в `/etc/crypttab`.
6. Откройте шифрованный раздел.
7. Создайте файловую систему на открытом разделе.
8. Опишите подключение вновь созданной ФС в `/etc/fstab`.
9. Смонтируйте раздел и восстановите данные в /home.
10. Перезагрузите систему и проверьте что доступ к прежним данным пользователей восстановился.

Глава 3. Подключаемые модули аутентификации (PAM).

3.1 Настройка PAM

Задание выполняется на CL1.

1. Проверьте, что у вас установлен модуль pam_mkhomedir.so.

```
# ls /lib/x86_64-linux-gnu/security/pam_mkhomedir.so
```

2. Настройте PAM на автоматическое создание домашних каталогов пользователей, используя модуль pam_mkhomedir.so.

```
# pam-auth-update
```

Выбрать опцию «Create home directory on login».

3. Проверить в каком(их) файле произведена настройка.

```
# grep mkhome /etc/pam.d/*
/etc/pam.d/common-session:session   optional                pam_mkhomedir.so
```

4. Запустите команду `pam-auth-update` и проверьте, что опции для автоматической настройки модуля pam_umask нет.
5. Так же проверьте, что в файлах настройки PAM тоже нет упоминания модуля pam_umask.

```
# grep umask /etc/pam.d/*
```

6. Откройте справку по модулю и прочтите, как его настраивать.

```
# man pam_umask
```

7. В конце файла `/etc/pam.d/common-session` добавьте строку с подключением модуля.

```
# tail -1 /etc/pam.d/common-session
session optional pam_umask.so
```

8. Проверьте текущее значение umask пользователя user.

```
# su - user
$ umask
0022
$ exit
```

9. Измените описание учетной записи user, определив маску равную 0077.

```
# chfn -o umask=0077 user
# getent passwd user
user:x:1002:1002:,,,,umask=0077:/home/user:/bin/bash
```

10. Проверьте, что при новом входе пользователь получил значение umask, назначенное в свойствах учетной записи.

```
# su - user
$ umask
0077
$ exit
```

Глава 4. Аутентификация.

4.1 Учетные записи.

Задание выполняется на CL1.

1. Просмотрите файл /etc/shadow (с правами root). У всех ли пользователей содержимое второго поля выглядит приблизительно одинаково или же у некоторых пользователей имеются существенные отличия от записей для других пользователей?
2. Какие символы могут содержаться шифрованной строке пароля во втором поле файла /etc/shadow ?
3. Используя текстовые утилиты выведите таблицу, содержащую имена пользователей, их UID и GID, а также шифрованный пароль (необходимо иметь права root). Таблица должна содержать записи только для пользователей имеющих шифрованные пароли длиной более 10 символов.

```
# join --nocheck-order -t: /etc/passwd /etc/shadow | awk -F:
'length($8)>10{OFS=":"; print $1,$3,$4,$8}'
```

4. Зарегистрируйте пользователя test1, для которого запрещен вход в сеанс, имеющего домашний каталог /var/spool/mail, и являющегося членом групп users и mail. Пользователь должен иметь UID=1100.

```
# useradd -d /var/spool/mail -M -G users,mail -u 1100 -s /sbin/nologin test1
```

5. Создайте учетную запись для пользователя test2 с настройками по умолчанию, но без создания приватной группы. Проверьте создался ли домашний каталог пользователя, если да, то наполнен ли он файлами, и кому он принадлежит.

```
# useradd -N test2
```

Домашний каталог в Debian автоматически не создается.

6. Измените имя пользователя test2 на test3.

```
# usermod -l test3 test2
```

7. Откройте сеанс для пользователя test3 через su или sudo. При входе должен быть создан домашний каталог. Затем закройте сеанс.

```
# sudo -u test3 -i
Создание каталога /home/test2.
```

8. Какой каталог был создан для пользователя test3. Исправьте в настройках пользователя путь к домашнему каталогу.

```
# usermod -d /home/test3 test3
```

9. Получите идентификаторы пользователя test3.

```
# id test3
uid=1101(test3) gid=100(users) группы=100(users)
```

10. Удалите пользователя test3. И проверьте кому теперь принадлежат каталоги /home/test2 и /home/test3.

```
# userdel test3
```

```
# ls -ld /home/test*
drwxr-xr-x 2 1101 users 4096 фев 11 17:58 /home/test2
drwxr-xr-x 2 1101 users 4096 фев 11 18:01 /home/test3
```

11. В файле `/etc/default/useradd` измените значение переменной `SHELL` на `/bin/bash`
12. В файле `/etc/login.defs` создайте переменную которая настраивает автоматическое создание домашних каталогов пользователей при их добавлении.

```
# tail -1 /etc/login.defs
CREATE_HOME yes
```

13. Зарегистрируйте пользователя test4 с настройками по умолчанию. Какая оболочка установлена для test4. Проверьте появился ли домашний каталог. Кому принадлежат каталоги вида test*.

```
# useradd test4

# getent passwd test4
test4:x:1101:1101::/home/test4:/bin/bash

# ls -ld /home/test*
drwxr-xr-x 2 test4 users 4096 фев 11 17:58 /home/test2
drwxr-xr-x 2 test4 users 4096 фев 11 18:01 /home/test3
drwxr-xr-x 2 test4 test4 4096 фев 11 18:13 /home/test4
```

14. Получите идентификатор пользователя test4 и сравните его с тем, что был у test3.

```
# id test4
uid=1101(test4) gid=1101(test4) группы=1101(test4)
```

15. Установите пароль для test4 . Изучите содержимое соответствующей пользователю записи в файле `/etc/shadow`.

```
# passwd test4

# getent shadow test4
test4:$y$j9T$xSSg1k2mUC4rsCUHuVG5h0$JN2g.crSaMZcdzgPsmK1XDiOFDfjJDHAiygv26hU6UA:
20130:0:99999:7:::
```

16. Установите дату устаревания учетной записи test4 на 31 декабря текущего года. Проверьте, что изменилось в `/etc/shadow`.

```
# chage -l test4
Последний раз пароль был изменён                      : фев 11, 2025
Срок действия пароля истекает                         : никогда
Пароль будет деактивирован через                      : никогда
Срок действия учётной записи истекает                       : никогда
Минимальное количество дней между сменой пароля             : 0
Максимальное количество дней между сменой пароля            : 99999
Количество дней с предупреждением перед деактивацией пароля: 7

# chage -E 2025-12-31 test4

# chage -l test4
Последний раз пароль был изменён                      : фев 11, 2025
Срок действия пароля истекает                         : никогда
Пароль будет деактивирован через                      : никогда
Срок действия учётной записи истекает                       : дек 31, 2025
Минимальное количество дней между сменой пароля             : 0
Максимальное количество дней между сменой пароля            : 99999
```

```
Количество дней с предупреждением перед деактивацией пароля: 7

# getent shadow test4
test4:$y$j9T$xSSg1k2mUC4rsCUHuVG5h0$JN2g.crSaMZcdzgPsmK1XDiOFDfjJDHAiygv26hU6UA:
20130:0:99999:7::20453:
```

17. Удалите пароль пользователя и проверьте изменения в `/etc/shadow`.

```
# passwd --delete test4
passwd: пароль изменён.

# getent shadow test4
test4::20130:0:99999:7::20453:
```

18. Попробуйте войти пользователем test4. Пускает ли система без пароля?

```
# login test4
Linux client 6.1.0-25-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.106-3 (2024-08-26)
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.
Последний вход в систему: Вт фев 11 18:25:01 +05 2025 на pts/1
```

Да можно войти без пароля.

19. Запретите вход с пустыми паролями.

Необходимо убрать опцию nullok в настройках модуля pam_unix.so.

```
root@client:~# grep unix /etc/pam.d/common-auth
#auth [success=1 default=ignore]    pam_unix.so nullok
auth  [success=1 default=ignore]    pam_unix.so
```

20. Заблокируйте учетную запись test4.

```
# passwd -l test4
passwd: пароль изменён.

# getent shadow test4
test4:!:20130:0:99999:7::20453:
```

4.2 Качество паролей

Задание выполняется на CL1.

1. Установите пакеты libpwquality-tools и libpam-pwquality.

```
# apt install libpwquality-tools libpam-pwquality
```

2. Придумайте свой пароль и проверьте его на сложность.

```
# echo 'Q1w2e3r4#$%' | pwscore
Проверка сложности пароля завершилась неудачей:
 Пароль не прошел проверку орфографии - основан на слове из словаря
```

3. Попробуйте сгенерировать несколько паролей автоматически и сравните их сложность со своими паролями.

```
# pwmake 1
Warning: Value 1 is outside of the allowed entropy range, adjusting it.
gekf0v$yqOdI

# pwmake 10
Warning: Value 10 is outside of the allowed entropy range, adjusting it.
@h^3p@q3n%Oz

# pwmake 77
xAJzUpewnYjyR1Uk

# pwmake 100
ExypUthANyquhQysoKROP*
```

4. Проверьте в каких файлах PAM используется модуль pam_pwquality.

```
# grep pwquality /etc/pam.d/*
/etc/pam.d/common-password:password requisite                  pam_pwquality.so
retry=3
```

5. Настройте политику в отношении паролей: минимальная длина — 8, повторение имени пользователя не более чем в 4 символах, минимальное количество классов — 3, максимум повторений одного символа — 2, только для локальных пользователей.

```
# grep -v '^#' /etc/security/pwquality.conf
minlen = 6
minclass = 3
maxrepeat = 1
usersubstr = 4
local_users_only
```

6. Установите пароль lin123 от имени суперпользователя пользователю test4.

```
root@client:~# passwd test4
Новый пароль:
НЕУДАЧНЫЙ ПАРОЛЬ: Пароль содержит слишком меньше чем 3 символов различного типа
Повторите ввод нового пароля:
passwd: пароль успешно обновлён
```

7. Попробуйте изменить пароль пользователем test4 на tesQ1w2e3r4, test1234/ и Lin1234/

```
# su - test4
test4@cl1:~$ passwd
Смена пароля для test4.
Текущий пароль:
Новый пароль:  tesQ1w2e3r4
НЕУДАЧНЫЙ ПАРОЛЬ: Пароль не прошел проверку орфографии - основан на слове из
словаря
```

```
Новый пароль: test1234/
НЕУДАЧНЫЙ ПАРОЛЬ: Пароль содержит имя пользователя в той или иной форме
Новый пароль: Lin1234/
Повторите ввод нового пароля: Lin1234/
passwd: пароль успешно обновлён
```

4.3 Блокировка учетных записей.

1. Включите модуль pam_faillock. Когда будете редактировать файлы ориентируйтесь на уже имеющиеся строки, которые менять не надо: например в примере ниже параметры описываются относительно строки с модулем pam_unix.so.

```
# grep -A5 Primary /etc/pam.d/common-auth
# here are the per-package modules (the "Primary" block)
auth  required                        pam_faillock.so preauth
auth  [success=1 default=ignore]    pam_unix.so nullok
auth    [default=die]                   pam_faillock.so authfail
auth    sufficient                      pam_faillock.so authsucc
# here's the fallback if no module succeeds

# grep -A3 Primary /etc/pam.d/common-account
# here are the per-package modules (the "Primary" block)
account required                        pam_faillock.so
account     [success=1 new_authtok_reqd=done default=ignore]     pam_unix.so
# here's the fallback if no module succeeds
```

2. Настройте блокировку учетных записей на 30 мин при 5 неудачных попытках входа локального пользователя. Суперпользователь тоже должен блокироваться, но на 5 минут.

```
# grep -v '^#' /etc/security/faillock.conf
silent
local_users_only
deny = 5
unlock_time = 1800
even_deny_root
root_unlock_time = 300
```

3. Установите программу John The Ripper и подберите пароли для пользователя user. Используйте имеющийся словарь из пакета john-data. Пароль lin123 добавьте в словарь после строки qwerty.

```
# apt list installed john*
Вывод списка… Готово
john-data/stable,unstable,now 1.9.0-2 all [установлен]
john/stable,unstable,now 1.9.0-2 amd64 [установлен]

# unshadow /etc/passwd /etc/shadow | grep '^user:' > passwd

# cp /usr/share/john/password.lst .
# grep -A1 '^qwerty$' password.lst
qwerty
lin123

# john --format=crypt --wordlist=password.lst passwd

# john --show passwd

# cat .john/john.pot
```

4. *(Дополнительное). Установите пакет hydra. Установите пакет vsftpd. Запустите службу ftp. Отключите блокировку учетных записей и подберите пароль для пользователя user.

4.4 Группы.

1. Создайте группу пользователей xusers с GID 1010.

```
# groupadd -g 1010 xusers
```

2. Зарегистрируйте пользователя user в качестве участника группы xusers.

```
# usermod -aG xusers user
```

3. Используя утилиту `groupmod`, позволяющую изменять имена и GID групп. Измените имя группы на yusers.

```
# groupmod -n yusers xusers
```

```
# id user
uid=1002(user) gid=1002(user) группы=1002(user),100(users),1010(yusers)
```

4.5 Политики sudo.

1. Разрешите пользователю user выполнять команду passwd через утилиту sudo, для смены пароля любому пользователю кроме root.

```
# grep MYPASSWD /etc/sudoers
Cmnd_Alias MYPASSWD = /usr/bin/passwd ^[a-zA-Z0-9_]+$,!/usr/bin/passwd root
user ALL=MYPASSWD
```

Не редактируйте файл напрямую, используйте команду visudo.

```
# su - user
user@client:~$ sudo -l
[sudo] пароль для user:
Matching Defaults entries for user on client:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/
bin,
    use_pty

User user may run the following commands on client:
    (root) /usr/bin/passwd ^[a-zA-Z0-9_]+$, !/usr/bin/passwd root

$ sudo passwd test1
Новый пароль:
НЕУДАЧНЫЙ ПАРОЛЬ: Пароль должен содержать не менее 6 символов
Повторите ввод нового пароля:
passwd: пароль успешно обновлён

$ sudo passwd root
Sorry, user user is not allowed to execute '/usr/bin/passwd root' as root on
client.
```

4.6 Профили пользователей.

1. Измените значение umask на 027 для всех пользователей системы по умолчанию.

```
# cat /etc/profile.d/umask.sh
umask 027

# umask 022
# umask
0022

# su sa
$ umask
0022

# su - sa
$ umask
0027
```

2. Установите в собственном профиле оболочки псевдоним ll для команды ls -l.

```
$ grep alias\ ll .bashrc
alias ll='ls -l'
```

3. Каким образом сделать так, чтобы этот же псевдоним устанавливался и для всех вновь регистрируемых пользователей в системе, для которых оболочкой по умолчанию будет Bash?

Добавить алиас в /etc/bash.bashrc.

4. В каком файле удобнее всего добавить к переменной окружения PATH путь к каталогу bin, находящемуся в домашнем каталоге обычного пользователя?

```
~/.profile
```

5. Добавить на рабочий стол для всех новых пользователей ярлык для запуска браузера Firefox.

```
# mkdir /etc/skel/Desktop
# cp /usr/share/applications/mate-terminal.desktop /etc/skel/Desktop/
```

Для проверки удалить домашний каталог пользователя user и войти им в графику. Убедиться что ярлык появился на рабочем столе.

4.7 Активность пользователей.

1. Определите, когда была последний раз загружена система.

```
# last reboot | head -1
```

2. С помощью опции `-a` команды `who` получите подробную информацию о пользователях и статусе системы.
3. С помощью команды who получите список пользователей, входивших в сеанс ранее.

```
# who /var/log/wtmp
```

4. Сравните предыдущий список со списком, выводимым командой `last`.

last дополнительно показывает и перезагрузки системы.

5. Получите отчет по входам в сеанс суперпользователя с помощью `lastlog`.

```
# lastlog -u root
```

6. Кто входил в сеанс за последние пять дней?

```
# last -s $(date -d '-5 days' +'%Y-%m-%d')
```

7. Получите список активных сеансов.

```
# loginctl
SESSION  UID USER SEAT  TTY
      2 1000 sa         pts/0
     26 1000 sa   seat0
```

8. Получите подробную информацию о статусе одного из сеансов. Какие процессы там запущены?

```
# loginctl session-status 2
```

9. Заблокируйте и разблокируйте сеанс с графическим входом из другого терминала.

```
# loginctl lock-session 26
# loginctl unlock-session 26
```

10. Попробуйте заблокировать сеанс в текстовом терминале.

```
# loginctl lock-session 2
```

Блокировка не работает.

11. Завершите сессию в графическом терминале. После закрытия сессии посмотрите какая сессия откралась на этом терминале.

```
# loginctl terminate-session 26
```

Надо дождаться завершения сессии.

```
# loginctl
SESSION  UID USER    SEAT  TTY
      2 1000 sa            pts/0
     c5  108 lightdm seat0

# loginctl session-status c5
c5 - lightdm (108)
           Since: Tue 2025-02-11 22:14:14 +05; 1min 2s ago
          Leader: 6436 (lightdm)
```

```
   Seat: seat0; vc7
Display: :0
Service: lightdm-greeter; type x11; class greeter
  State: active
   Unit: session-c5.scope
         ├─6436 lightdm --session-child 17 20
         └─6460 /usr/sbin/lightdm-gtk-greeter
```

Глава 5. Доступ к файлам.

5.1 Создание папки для общего доступа

Задание выполняется на srv1.

1. От имени суперпользователя создайте каталог `/home/project`.
2. Добавьте пользователей user1, user2 и user3.

```
# adduser user1
# adduser user2
# adduser user3
```

3. Создайте группы RW_project и RO_project.

```
# groupadd  RW_project
# groupadd  RO_project
```

4. Добавьте пользователей user1 и user2 в RW_project, а пользователя user3 в RO_project.

```
# gpasswd -a юзер группа
```

5. Добавьте пользователя sa в группу adm.
6. Назначьте SGID на каталог `/home/project`, и права доступа 770.

```
# chmod 2770 /home/project
```

7. Сделайте группу adm владельцем каталога `/home/project`.

```
# chgrp adm /home/project
```

8. Пользователем sa создайте в каталог `/home/project/d1` и файл `f1` в нем.
9. Проверьте права доступа на содержимое `/home/project`.

```
$ id
uid=1000(sa) gid=1000(sa)
группы=1000(sa),4(adm),24(cdrom),25(floppy),27(sudo),29(audio),30(dip),44(video)
,46(plugdev),100(users),106(netdev),111(lpadmin),114(scanner),121(docker)
контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023

$ mkdir /home/project/d1

$ touch /home/project/d1/f1

$ ls -Rl /home/project/
/home/project/:
итого 4
drwxr-sr-x. 2 sa adm 4096 фев 12 14:12 d1

/home/project/d1:
итого 0
-rw-r--r--. 1 sa adm 0 фев 12 14:12 f1
```

10. Создайте ACL для каталога `/home/project` так чтобы пользователи в группе RW_project имели права rw, а RO_project — r. Остальные пользователи никаких прав иметь не должны.

```
# setfacl -m g:RW_project:rwx,g:RO_project:rx /home/project
```

11. Проверьте распространяются ли права доступа каталога `/home/project` на `d1` и `f1`.

```
# getfacl -R /home/project
getfacl: Removing leading '/' from absolute path names
# file: home/project
# owner: root
# group: adm
# flags: -s-
user::rwx
group::rwx
group:RW_project:rwx
group:RO_project:r-x
mask::rwx
other::---

# file: home/project/d1
# owner: sa
# group: adm
# flags: -s-
user::rwx
group::r-x
other::r-x

# file: home/project/d1/f1
# owner: sa
# group: adm
user::rw-
group::r--
other::r--
```

12. Перенесите ACL с каталога `/home/project` на все его текущее содержимое.

```
# setfacl -R -M<(getfacl /home/project) /home/project/*
getfacl: Removing leading '/' from absolute path names

# getfacl -R /home/project
getfacl: Removing leading '/' from absolute path names
# file: home/project
# owner: root
# group: adm
# flags: -s-
user::rwx
group::rwx
group:RW_project:rwx
group:RO_project:r-x
mask::rwx
other::---

# file: home/project/d1
# owner: sa
# group: adm
# flags: -s-
user::rwx
group::rwx
group:RW_project:rwx
group:RO_project:r-x
mask::rwx
other::---

# file: home/project/d1/f1
# owner: sa
# group: adm
```

```
user::rwx
group::rwx
group:RW_project:rwx
group:RO_project:r-x
mask::rwx
other::---
```

13. Пользователем user1 создайте в каталог `/home/project/d2` и файл `f2` в нем.
14. Получилось ли создать файлы? Если получилось, то какие права доступа получились у файлов и каталогов?

```
# su - user1
$ mkdir /home/project/d2
$ touch /home/project/d2/f2

$ getfacl -R /home/project/d2
getfacl: Removing leading '/' from absolute path names
# file: home/project/d2
# owner: user1
# group: adm
# flags: -s-
user::rwx
group::r-x
other::r-x

# file: home/project/d2/f2
# owner: user1
# group: adm
user::rw-
group::r--
other::r--
```

Вновь созданные файлы не наследуют ACL от каталога. Идет только наследование группы владельцев из-за установленного бита SGID.

15. Установите ACL по умолчанию (default ACL) на каталог `/home/project` так, чтобы члены групп RW_project, RO_project и adm получали правильные права доступа на вновь создаваемые объекты.

```
# setfacl -d -m g:RW_project:rwx,g:RO_project:rx,g::rwx,m::rwx /home/project/

# getfacl /home/project/
getfacl: Removing leading '/' from absolute path names
# file: home/project/
# owner: root
# group: adm
# flags: -s-
user::rwx
group::rwx
group:RW_project:rwx
group:RO_project:r-x
mask::rwx
other::---
default:user::rwx
default:group::rwx
default:group:RW_project:rwx
default:group:RO_project:r-x
default:mask::rwx
default:other::---
```

16. Пользователем user1 создайте в каталог `/home/project/d3` и файл `f3` в нем.

17. Какие права доступа получились у новых файлов и каталогов?

```
# su - user1
$ mkdir /home/project/d3
$ touch /home/project/d3/f3

$ getfacl -R /home/project/d3
getfacl: Removing leading '/' from absolute path names
# file: home/project/d3
# owner: user1
# group: adm
# flags: -s-
user::rwx
group::rwx
group:RW_project:rwx
group:RO_project:r-x
mask::rwx
other::---
default:user::rwx
default:group::rwx
default:group:RW_project:rwx
default:group:RO_project:r-x
default:mask::rwx
default:other::---

# file: home/project/d3/f3
# owner: user1
# group: adm
user::rw-
group::rwx              #effective:rw-
group:RW_project:rwx          #effective:rw-
group:RO_project:r-x          #effective:r--
mask::rw-
other::---
```

18. Проверьте есть ли наследуемые ACL на каталоге `/home/project/d1`.

```
$ getfacl  /home/project/d1
getfacl: Removing leading '/' from absolute path names
# file: home/project/d1
# owner: sa
# group: adm
# flags: -s-
user::rwx
group::rwx
group:RW_project:rwx
group:RO_project:r-x
mask::rwx
other::---
```

19. Перенесите ACL с каталога `/home/project` на все его текущее содержимое.

```
# setfacl -R -M<(getfacl /home/project) /home/project/*
```

20. Проверьте, что теперь обычные файлы имеют права на исполнение. Исправьте это.

```
# getfacl /home/project/d1/f1
getfacl: Removing leading '/' from absolute path names
# file: home/project/d1/f1
# owner: sa
# group: adm
```

```
user::rwx
group::rwx
group:RW_project:rwx
group:RO_project:r-x
mask::rwx
other::---

# find /home/project/ -type f | xargs setfacl -m m::rw

# getfacl /home/project/d1/f1
getfacl: Removing leading '/' from absolute path names
# file: home/project/d1/f1
# owner: sa
# group: adm
user::rwx
group::rwx              #effective:rw-
group:RW_project:rwx          #effective:rw-
group:RO_project:r-x          #effective:r--
mask::rw-
other::---
```

5.2 Специальные биты доступа.

1. Проверьте, какие биты прав доступа установлены на исполняемый файл команды passwd.

```
# stat $(which passwd)
  Файл: /usr/bin/passwd
  Размер: 68248         Блоков: 136        Блок В/В: 4096   обычный файл
Устройство: 8/17  Инода: 261754      Ссылки: 1
Доступ: (4755/-rwsr-xr-x)  Uid: (    0/    root)   Gid: (    0/    root)
Контекст: system_u:object_r:passwd_exec_t:s0
Доступ:        2025-02-07 22:23:35.322108838 +0500
Модифицирован: 2023-03-23 17:40:50.000000000 +0500
Изменён:       2025-02-04 16:43:52.500000000 +0500
Создан:        2024-09-21 15:50:21.543299000 +0500
```

2. С помощью `ps` и `awk` получите список всех процессов в системе, для которых RUID не равен EUID.

```
# ps -eo ruid,euid,tty,cmd | awk '$1!=$2'
 RUID  EUID TT       CMD
 1000     0 pts/0    sudo -i
 1000     0 pts/1    sudo -i
```

3. На какой-либо виртуальной консоли запустите команду passwd от имени обычного пользователя. Выполните ту же команду, что и в предыдущем пункте. Имеется ли в требуемом списке процесс passwd?

```
# ps -eo ruid,euid,tty,cmd | awk '$1!=$2'
 RUID  EUID TT       CMD
 1000     0 pts/0    sudo -i
 1000     0 pts/1    sudo -i
 1000     0 pts/2    passwd
```

4. Используя `find` запишите в файл `SUGID.txt` все имена файлов, на которые установлены биты SUID или SGID.

```
# find / -type f -perm /6000 2>/dev/null | xargs ls -l > SUGID.txt

# head -3 SUGID.txt
-rwsr-xr-x. 1 root root        377320 янв 11  2023
/opt/VBoxGuestAdditions-7.0.6/bin/VBoxDRMClient
-rwxr-sr-x. 1 root shadow       80376 мар 23  2023 /usr/bin/chage
-rwsr-xr-x. 1 root root         62672 мар 23  2023 /usr/bin/chfn
```

5. Найдите в `/home` все каталоги с установленными битами SGID или Sticky bit.

```
# find /home -type d -perm /3000 2>/dev/null | xargs ls -ld
```

5.3 Дополнительные атрибуты файлов.

1. Создайте в домашнем каталоге подкаталог dir с файлами 1.txt, 2.txt, 3.txt

```
$ mkdir dir
$ for VAR in {1..3} ; do echo $VAR > dir/${VAR}.txt ; done
```

2. Установите атрибут a для 1.txt, i для 2.txt и c для 3.txt

```
$ chattr +a dir/1.txt
chattr: Операция не позволена while setting flags on dir/1.txt

$ sudo chattr +a dir/1.txt
$ sudo chattr +i dir/2.txt
$ chattr +c dir/3.txt
```

3. Попробуйте выполнить с файлами следующие операции: добавить информацию в файлы, перезаписать содержимое, удалить файлы.

```
$ for VAR in {1..3} ; do echo new_$VAR > dir/${VAR}.txt ; done
-bash: dir/1.txt: Операция не позволена
-bash: dir/2.txt: Операция не позволена

$ for VAR in {1..3} ; do echo new_$VAR >> dir/${VAR}.txt ; done
-bash: dir/2.txt: Операция не позволена

 for VAR in {1..3} ; do rm dir/${VAR}.txt ; done
rm: невозможно удалить 'dir/1.txt': Операция не позволена
rm: невозможно удалить 'dir/2.txt': Операция не позволена

$ for VAR in {1..3} ; do sudo rm dir/${VAR}.txt ; done
rm: невозможно удалить 'dir/1.txt': Операция не позволена
rm: невозможно удалить 'dir/2.txt': Операция не позволена
rm: невозможно удалить 'dir/3.txt': Нет такого файла или каталога
```

Глава 6. Модули безопасности Linux.

6.1 SELinux.

Задание выполняется на srv1.

1. Установите пакеты selinux-basics, selinux-policy-default, selinux-utils и auditd.
2. Выполнить команду `selinux-activate`, которая настроит GRUB, PAM и создаст файл `/.autorelabel`.
3. Перезагрузите ОС.
4. Проверьте установку командой `check-selinux-installation`.
5. Проверьте состояние SELinux. Если оно выключено, то включите его или переведите в режим enforcing.

```
# sestatus
SELinux status:                 enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:             default
Current mode:                   enforcing
Mode from config file:          enforcing
Policy MLS status:              enabled
Policy deny_unknown status:     allowed
Memory protection checking:     actual (secure)
Max kernel policy version:      33
```

После изменения в файле `/etc/selinux/config` нужна перезагрузка.

6. Установите веб-сервер apache и проанализируйте как промаркированы файлы этой службы.

```
# apt install apache2

# ls -Z /var/www/html/index.html
unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/index.html

# ps -eZ | grep apache2
system_u:system_r:httpd_t:s0       1609 ?        00:00:00 apache2
system_u:system_r:httpd_t:s0       1610 ?        00:00:00 apache2
system_u:system_r:httpd_t:s0       1639 ?        00:00:00 apache2

# ls -Z /usr/sbin/apache2
system_u:object_r:httpd_exec_t:s0 /usr/sbin/apache2
```

7. В файл `/var/www/html/index.html` запишите одну строку: Default Site. Убедитесь, что веб-сервер показывает стартовую страницу.

```
# echo Default Site > /var/www/html/index.html

# curl http://127.0.0.1
Default Site
```

8. Создайте алиас для каталога `/home/myweb`, поместите в этот каталог файл `index.html` и проверьте доступна ли эта страница. На данном этапе страница не должна быть доступна.

```
# mkdir /home/myweb
# echo 'My test Web Page' > /home/myweb/index.html

# cat /etc/apache2/sites-available/myweb.conf
Alias /myweb /home/myweb
<Directory /home/myweb>
 AllowOverride none
 Require all granted
</Directory>

# a2ensite myweb
Enabling site myweb.
To activate the new configuration, you need to run:
  systemctl reload apache2

# systemctl reload apache2

# curl http://127.0.0.1/myweb/index.html
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>403 Forbidden</title>
</head><body>
<h1>Forbidden</h1>
<p>You don't have permission to access this resource.</p>
<hr>
<address>Apache/2.4.62 (Debian) Server at 127.0.0.1 Port 80</address>
</body></html>
```

9. Сделайте временную перемаркировку файлов для веб-сервера и проверьте доступ к странице алиаса.

```
# chcon -R  -t httpd_sys_content_t /home/myweb/
# curl http://127.0.0.1/myweb/index.html
My test Web Page
```

10. Восстановите маркировку по умолчанию для `/home`. Проверьте доступность страницы.

```
# restorecon -R /home/

# curl http://127.0.0.1/myweb/index.html
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>403 Forbidden</title>
</head><body>
<h1>Forbidden</h1>
<p>You don't have permission to access this resource.</p>
<hr>
<address>Apache/2.4.62 (Debian) Server at 127.0.0.1 Port 80</address>
</body></html>
```

11. Создайте постоянное правило маркировки примените эти правила к каталогу `/home/myweb`. Убедитесь, что правила маркировки отработали правильно и доступ к алиасу восстановился.

```
# semanage fcontext -a -t httpd_sys_content_t '/home/myweb(/.*)?'
libsemanage.add_user: user sddm not in password file

# restorecon -vR /home/myweb/
Relabeled /home/myweb from unconfined_u:object_r:user_home_dir_t:s0 to
unconfined_u:object_r:httpd_sys_content_t:s0
Relabeled /home/myweb/index.html from unconfined_u:object_r:user_home_t:s0 to
unconfined_u:object_r:httpd_sys_content_t:s0

# !curl
curl http://127.0.0.1/myweb/index.html
My test Web Page
```

12. Разрешите сетевое подключение веб-сервера к базам данных.

```
# getsebool httpd_can_network_connect_db
httpd_can_network_connect_db --> off
# setsebool httpd_can_network_connect_db on
# getsebool httpd_can_network_connect_db
httpd_can_network_connect_db --> on
```

13. Проанализируйте журнал аудита и примите решение по устранению отказов в доступе. Стоит ли их принять, проигнорировать или исправить политики или настройки политик. Если необходимо внесите нужные изменения.

Например:

```
grep 'avc:  denied' /var/log/audit/audit.log | audit2allow
grep 'avc:  denied' /var/log/audit/audit.log | audit2why
```

Или

```
grep httpd_t /var/log/audit/audit.log | audit2allow
grep httpd_t /var/log/audit/audit.log | audit2why
```

6.2 *Создание модуля SELinux. Правила RBAC (необязательное).

1. Создайте собственный модуль для SELinux. (По примеру в учебнике).
2. Скомпилируйте и установите модуль.
3. Включите модуль.
4. Удалите свой модуль.
5. Создайте нового Unix пользователя и сопоставьте его с SELinux пользователем `staff_u`. Проверьте какие операции может выполнять данный пользователь.
6. Создайте SE пользователя `webadm_u`, связанного с ролью `webadm_u`. Затем Unix пользователя webadm, связанного с `webadm_u` и проверьте его привилегии.
7. Измените SELinux пользователя для учетной записи администратора на `user_u`. Проверьте действует ли изменение на работу от имени обычного и суперпользователя.

После выполнения лабораторных работ отключите SELinux!

```
$ sudo setenforce 0
$ sudo selinux-activate disable
$ sudo reboot
```

6.3 AppArmor

Выполняется на cl1.

1. Установите пакеты apparmor, apparmor-utils, apparmor-profiles, apparmor-profiles-extra.
2. Скопируйте программу `ping` в домашний каталог пользователя.

```
$ sudo cp -a /bin/ping ping
```

3. Проверьте, что копия `ping` работает. Если не работает, то назначьте linux capabilities cap_net_raw=ep на этот файл.

```
sudo setcap cap_net_raw=ep ping
```

4. Создайте профиль AppArmor для копии программы `ping`.

```
$ sudo aa-autodep /home/sa/ping
Writing updated profile for /home/sa/ping.

$ sudo cat /etc/apparmor.d/home.sa.ping
# Last Modified: Tue May  3 11:36:43 2022
#include <tunables/global>

/home/sa/ping flags=(complain) {
  #include <abstractions/base>

  /home/sa/ping mr,

}
```

5. Проверьте, что копия `ping` работает.

```
$ ./ping -c1 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.051 ms

--- 127.0.0.1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.051/0.051/0.051/0.000 ms
```

6. Переключите профиль в принудительный режим. Теперь `ping` должен перестать работать.

```
$ sudo aa-enforce /etc/apparmor.d/home.sa.ping
Setting /etc/apparmor.d/home.sa.ping to enforce mode.

$ ./ping -c1 127.0.0.1
./ping: socket: Operation not permitted
```

7. С помощью `aa-genprof` настройте профиль, чтобы он разрешал использование копии `ping`.

```
$ sudo aa-complain /etc/apparmor.d/home.sa.ping
Setting /etc/apparmor.d/home.sa.ping to complain mode.

 sudo aa-genprof /home/sa/ping

Before you begin, you may wish to check if a
profile already exists for the application you
```

```
wish to confine. See the following wiki page for
more information:
https://gitlab.com/apparmor/apparmor/wikis/Profiles

Profiling: /home/sa/ping

Please start the application to be profiled in
another window and exercise its functionality now.

Once completed, select the "Scan" option below in
order to scan the system logs for AppArmor events.

For each AppArmor event, you will be given the
opportunity to choose whether the access should be
allowed or denied.

[(S)can system log for AppArmor events] / (F)inish
Reading log entries from /var/log/audit/audit.log.
Updating AppArmor profiles in /etc/apparmor.d.
Complain-mode changes:

Profile:    /home/sa/ping
Capability: net_raw
Severity:   8

 [1 - capability net_raw,]
(A)llow / [(D)eny] / (I)gnore / Audi(t) / Abo(r)t / (F)inish
Adding capability net_raw, to profile.

= Changed Local Profiles =

The following local profiles were changed. Would you like to save them?

 [1 - /home/sa/ping]
(S)ave Changes / Save Selec(t)ed Profile / [(V)iew Changes] / View Changes b/w
(C)lean profiles / Abo(r)t
Writing updated profile for /home/sa/ping.

Profiling: /home/sa/ping

Please start the application to be profiled in
another window and exercise its functionality now.

Once completed, select the "Scan" option below in
order to scan the system logs for AppArmor events.

For each AppArmor event, you will be given the
opportunity to choose whether the access should be
allowed or denied.

[(S)can system log for AppArmor events] / (F)inish

Reloaded AppArmor profiles in enforce mode.

Please consider contributing your new profile!
See the following wiki page for more information:
https://gitlab.com/apparmor/apparmor/wikis/Profiles

Finished generating profile for /home/sa/ping.
```

```
$ sudo aa-enforce /etc/apparmor.d/home.sa.ping
Setting /etc/apparmor.d/home.sa.ping to enforce mode.

$ sudo cat /etc/apparmor.d/home.sa.ping
# Last Modified: Tue May  3 11:41:33 2022
#include <tunables/global>

/home/sa/ping {
  #include <abstractions/base>

  capability net_raw,

  /home/sa/ping mr,

}

$ ./ping -c1 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.051 ms

--- 127.0.0.1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.051/0.051/0.051/0.000 ms
```

Глава 7. Мониторинг событий безопасности средствами ОС Linux.

7.1 Работа с журналами.

1. Проверьте установлен ли пакет rsyslog на машинах. Если нет, то установите его на всех ВМ. Далее почти все действия выполняются на srv1 на cl1 и cl2 настраивается только клиенты syslog.
2. Создайте в конфигурации демона rsyslog строку, которая заставит его записывать сообщения от источника auth с уровнем важности не ниже info в файл `/var/log/mylog` .
3. Добавьте в конце файла `/etc/rsyslog.conf` строку:

```
auth.info                               /var/log/mylog
```

4. Определите, какой сигнал позволяет демону rsyslog перечитать файл конфигурации и продолжить работу с измененными настройками. Пошлите демону этот сигнал и проверьте, создался ли файл `/var/log/mylog` .

Такого сигнала нет, сигнал HUP (1) — переоткрыть все файлы журналов:

```
# man rsyslogd
```

5. Проверьте, записываются ли в этот файл сообщения при входе в сеанс и выходе из него пользователей.
6. Создайте аналогичный журнал `/var/log/mylogpriv` для записи сообщений от источника authpriv . Проверьте его работоспособность. Записываются ли в него те же сообщения, что и в предыдущий журнал?

Аналогично настройкам `/var/log/mylog`

```
authpriv.info                                   /var/log/mylogpriv
```

7. Настройте прием сообщений по протоколу UDP на srv1 и запись таких сообщений в отдельные журналы для каждого узла.
8. Удобней будет создать отдельный файл конфигурации, например `/etc/rsyslog.d/udp.conf` и в нем описать следующие строки:

```
$ModLoad imudp

$template RemHost,"/var/log/network/%HOSTNAME%.log"

$RuleSet remote_udp

*.* ?RemHost

$InputUDPServerBindRuleset remote_udp
$UDPServerRun 514
```

В основном файле конфигурации необходимо перед первым правилом объявить RuleSet для локальных сообщений и сделать его дефолтным:

```
#### RULES ####
$Ruleset local
...

$DefaultRuleset local
```

Или в новом синтаксисе:

```
module(load="imudp")
input(type="imudp" port="514" ruleset="RemoteUDP514")

template(name="NetworkLog" type="list") {
    constant(value="/var/log/network/")
    constant(value="/")
    property(name="hostname")
    constant(value=".log")
}

ruleset(name="RemoteUDP514") {
        action(type="omfile" dynaFileCacheSize="1024" dynaFile="NetworkLog"
FileOwner="root" FileGroup="root" dirOwner="root" dirGroup="root"
FileCreateMode="0640" DirCreateMode="0755" flushOnTXEnd="off" asyncWriting="on"
flushInterval="10" ioBufferSize="64k")
}
```

9. На cl1 и cl2 настройте перенаправление событий на ваш srv1. И протестируйте работу службы.

Добавьте в конце файла `/etc/rsyslog.conf` строку:

```
*.*                     @srv1:514
```

Чтобы это работало, надо записать адрес srv1 в файле `/etc/hosts`.

10. Посмотрите журнал загрузки системы.

```
# journalctl -b
```

11. Попробуйте воспользоваться опцией -l видите ли вы разницу? Если нет, то как увидеть?

```
# journalctl | less
```

12. Настройте systemd вести постоянные журналы.

Создать каталог `/var/log/journal`

13. Создайте настройки ротации для журналов `/var/log/mylog` и `/var/log/mylogpriv`, созданных ранее. Файлы должны ротироваться ежедневно. Ротация первой копии должна осуществляться два раза. Ротируемые копии должны сжиматься утилитой gzip.

14. Создать новый файл `/etc/logrotate.d/mylog` со следующим содержимым:

```
/var/log/mylog
/var/log/mylogpriv
{
        daily
        rotate 2
        compress
        missingok
        notifempty
```

```
        sharedscripts
        postrotate
                /usr/lib/rsyslog/rsyslog-rotate
        endscript
}
```

15. Для файла `/var/log/mylog` установите в качестве дополнительного условия ротации достижение им размера 10 Кб.
16. Изменить файл `/etc/logrotate.d/mylog` следующим образом:

```
/var/log/mylog
{
        daily
        rotate 2
        size 10k
        compress
        missingok
        notifempty
        sharedscripts
        postrotate
                /usr/lib/rsyslog/rsyslog-rotate
        endscript
}
/var/log/mylogpriv
{
        daily
        rotate 2
        compress
        missingok
        notifempty
        sharedscripts
        postrotate
                /usr/lib/rsyslog/rsyslog-rotate
        endscript
}
```

17. Создайте правила ротации журналов с сообщениями, поступающими из сети. Параметры ротации — ежедневно, 7 копий, со сжатием 2 копии, суффикс в виде даты.

```
# cat /etc/logrotate.d/network
/var/log/network/*.log
{
        daily
        rotate 7
        dateext
        compress
        delaycompress
        missingok
        notifempty
        sharedscripts
        postrotate
                /usr/lib/rsyslog/rsyslog-rotate
        endscript
}
```

18. Настройте автоматическую ротацию журнала `/var/log/messages`, посредством исходящих сообщений rsyslog. Для ротации используйте logrotate с отдельным конфигурационным файлом. Размер журнала для ротации 1MB.

Создать файл со скритом:

```
touch /usr/local/sbin/log_rotation_script
chmod +x /usr/local/sbin/log_rotation_script
```

Содержимое скрипта:

```
#!/bin/bash
/sbin/logrotate /etc/logrotate.messages
```

В файле `/etc/logrotate.messages` параметры ротации, например:

```
/var/log/messages
{
    rotate 10
    size 1k
    missingok
    sharedscripts
    postrotate
        /usr/lib/rsyslog/rsyslog-rotate
    endscript
}
```

В конфигурации rsyslog:

```
$outchannel
mymessages,/var/log/messages,1048576,/usr/local/sbin/log_rotation_script

*.*                             :omfile:$mymessages
```

Перезапустить rsyslog

Проверка:

```
# for msg in test_msg_{1..99999}; do logger $msg; done&
# watch ls -l /var/log/messages
```

19. Настройте блокировку учетной записи пользователя, который выполняет не разрешенные команды sudo.

Найдите сообщения от системы авторизации связанные с использованием `sudo`. Просмотрите их видите ли вы события связанные с неправомерным использованием `sudo`. Как выглядят такие сообщения.

```
# grep sudoers /var/log/auth.log
2025-02-12T20:04:31.234347+05:00 srv1 sudo:    user1 : user NOT in sudoers ;
TTY=pts/2 ; PWD=/home/user1 ; USER=root ; COMMAND=/usr/bin/ls

# cat /usr/local/sbin/mylogparser
#!/bin/bash
export LANG=C
CDATE=$(date +'%b %_d %H:%M:%S')

BUSER=$(echo $@ | awk -F: '{print $1}' | tr -dc a-zA-Z0-9)

passwd -l ${BUSER}

echo "${CDATE} $(hostname) : User ${BUSER} locked" >> /var/log/notinsudo.log
```

```
loginctl terminate-user ${BUSER}

# grep tmpl1 /etc/rsyslog.conf
$template tmpl1,"%msg%"
if $programname == 'sudo' and $msg contains 'user NOT in sudoers' then
^/usr/local/sbin/mylogparser;tmpl1

$ ssh user1@172.27.2.134
user1@172.27.2.134's password:

$ sudo ls
[sudo] password for user1:
user1 is not in the sudoers file.

$ Connection to 172.27.2.134 closed by remote host.
Connection to 172.27.2.134 closed.
$ ssh user1@172.27.2.134
user1@172.27.2.134's password:

# getent shadow user2
user1:!$y$j9T$nj25dzr1PWNkHpXmGtYnn/$HRd3TFICWlVelkG3pVL.P/UzR3I8I3JKb9AOb/
VRNl.:19100:0:99999:7:::
```

Важное замечание!!!. Такой вариант не сработает при включенном SELinux.

20. Настройте rsyslog на запись IP адреса, с которого производится попытка входа по протоколу ssh пользователем root в файл `/var/log/badips.log`.

```
# grep programname /etc/rsyslog.conf
if $programname == 'sshd' and $msg contains 'Failed password for root from' then
^/usr/local/sbin/mylogparser

# cat /usr/local/sbin/mylogparser
#!/bin/bash
PATH=/usr/local/bin:/usr/bin:/bin:/sbin
CDATE=$(date -Iseconds)
remIP=$(echo $1 | sed -r 's/.*from ([0-9]+\.[0-9]+\.[0-9]+\.[0-9]+).*/\1/')
echo "${CDATE} : Attack detected from ${remIP}" >> /var/log/badips.log
```

21. *Установите пакет logwatch и просмотрите отчет о функционировании вашего компьютера.
22. *Настройте получение статистики от web-серверас помощью webalizer.
23. *Установите пакет webalizer. И изучите файл конфигурации /etc/webalizer/webalizer.conf. Обратите внимание на опции конфигурации LogFile и OutputDir.

7.2 Аудит.

1. Проверьте запущен ли демон аудита.

```
# systemctl is-active auditd.service
active
```

2. Проверьте его настройки.

Проверить содержимое каталога `/etc/audit` и файлов в нем.

3. Создайте правило для отслеживания обращений к файлу `/etc/shadow` и `/etc/passwd`.

```
# auditctl -w /etc/passwd -p wa -k passwd_access
# auditctl -w /etc/shadow -p wa -k shadow_access

# auditctl -l
-w /etc/passwd -p wa -k passwd_access
-w /etc/shadow -p wa -k shadow_access
```

4. Перезапустите аудит и проверьте, что правила не сохранились

```
# systemctl restart auditd.service
# auditctl -l
No rules
```

5. Опишите ваши правила в конфигурационном файле.

```
# cat /etc/audit/rules.d/my.rules
-w /etc/passwd -p wa -k passwd_access
-w /etc/shadow -p wa -k shadow_access

# systemctl restart auditd.service

# auditctl -l
-w /etc/passwd -p wa -k passwd_access
-w /etc/shadow -p wa -k shadow_access
```

6. Проверьте регистрируются ли события когда вы добавляете нового пользователя или меняете пароль.

```
# adduser newuser1

# egrep '(passwd|shadow)_access' /var/log/audit/audit.log
type=CONFIG_CHANGE msg=audit(1739373235.380:167): auid=1000 ses=2
subj=unconfined op=add_rule key="passwd_access" list=4 res=1AUID="sa"
type=CONFIG_CHANGE msg=audit(1739373247.654:168): auid=1000 ses=2
subj=unconfined op=add_rule key="shadow_access" list=4 res=1AUID="sa"
type=CONFIG_CHANGE msg=audit(1739373372.217:173): auid=4294967295 ses=4294967295
subj=unconfined op=remove_rule key="passwd_access" list=4 res=1AUID="unset"
type=CONFIG_CHANGE msg=audit(1739373372.217:174): auid=4294967295 ses=4294967295
subj=unconfined op=remove_rule key="shadow_access" list=4 res=1AUID="unset"
type=CONFIG_CHANGE msg=audit(1739373464.575:193): auid=4294967295 ses=4294967295
subj=unconfined op=add_rule key="passwd_access" list=4 res=1AUID="unset"
type=CONFIG_CHANGE msg=audit(1739373464.575:194): auid=4294967295 ses=4294967295
subj=unconfined op=add_rule key="shadow_access" list=4 res=1AUID="unset"
type=SYSCALL msg=audit(1739373501.873:199): arch=c000003e syscall=257
success=yes exit=5 a0=ffffff9c a1=55a12516cec0 a2=20902 a3=0 items=1 ppid=1551
pid=1558 auid=1000 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0
tty=pts1 ses=2 comm="useradd" exe="/usr/sbin/useradd" subj=unconfined
```

```
key="passwd_access"ARCH=x86_64 SYSCALL=openat AUID="sa" UID="root" GID="root"
EUID="root" SUID="root" FSUID="root" EGID="root" SGID="root" FSGID="root"
type=SYSCALL msg=audit(1739373501.978:200): arch=c000003e syscall=257
success=yes exit=10 a0=ffffff9c a1=55a12516e160 a2=20902 a3=0 items=1 ppid=1551
pid=1558 auid=1000 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0
tty=pts1 ses=2 comm="useradd" exe="/usr/sbin/useradd" subj=unconfined
key="shadow_access"ARCH=x86_64 SYSCALL=openat AUID="sa" UID="root" GID="root"
EUID="root" SUID="root" FSUID="root" EGID="root" SGID="root" FSGID="root"
type=SYSCALL msg=audit(1739373502.042:202): arch=c000003e syscall=82 success=yes
exit=0 a0=7fffe23ad9e0 a1=55a12516cec0 a2=7fffe23ad950 a3=100 items=5 ppid=1551
pid=1558 auid=1000 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0
tty=pts1 ses=2 comm="useradd" exe="/usr/sbin/useradd" subj=unconfined
key="passwd_access"ARCH=x86_64 SYSCALL=rename AUID="sa" UID="root" GID="root"
EUID="root" SUID="root" FSUID="root" EGID="root" SGID="root" FSGID="root"
type=SYSCALL msg=audit(1739373502.082:203): arch=c000003e syscall=82 success=yes
exit=0 a0=7fffe23ad9e0 a1=55a12516e160 a2=7fffe23ad950 a3=100 items=5 ppid=1551
pid=1558 auid=1000 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0
tty=pts1 ses=2 comm="useradd" exe="/usr/sbin/useradd" subj=unconfined
key="shadow_access"ARCH=x86_64 SYSCALL=rename AUID="sa" UID="root" GID="root"
EUID="root" SUID="root" FSUID="root" EGID="root" SGID="root" FSGID="root"
type=SYSCALL msg=audit(1739373507.552:207): arch=c000003e syscall=257
success=yes exit=4 a0=ffffff9c a1=562b66c25700 a2=20902 a3=0 items=1 ppid=1551
pid=1568 auid=1000 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0
tty=pts1 ses=2 comm="chfn" exe="/usr/bin/chfn" subj=unconfined
key="passwd_access"ARCH=x86_64 SYSCALL=openat AUID="sa" UID="root" GID="root"
EUID="root" SUID="root" FSUID="root" EGID="root" SGID="root" FSGID="root"
type=SYSCALL msg=audit(1739373507.596:208): arch=c000003e syscall=82 success=yes
exit=0 a0=7ffc8361b390 a1=562b66c25700 a2=7ffc8361b300 a3=100 items=5 ppid=1551
pid=1568 auid=1000 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0
tty=pts1 ses=2 comm="chfn" exe="/usr/bin/chfn" subj=unconfined
key="passwd_access"ARCH=x86_64 SYSCALL=rename AUID="sa" UID="root" GID="root"
EUID="root" SUID="root" FSUID="root" EGID="root" SGID="root" FSGID="root"

# ausearch -k passwd_access
----
time->Wed Feb 12 20:13:55 2025
type=PROCTITLE msg=audit(1739373235.380:167):
proctitle=617564697463746C002D77002F6574632F706173737764002D70007761002D6B007061
737377645F616363657373
type=SYSCALL msg=audit(1739373235.380:167): arch=c000003e syscall=44 success=yes
exit=1080 a0=4 a1=7fff809e5080 a2=438 a3=0 items=0 ppid=1047 pid=1374 auid=1000
uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=pts1 ses=2
comm="auditctl" exe="/usr/sbin/auditctl" subj=unconfined key=(null)
type=CONFIG_CHANGE msg=audit(1739373235.380:167): auid=1000 ses=2
subj=unconfined op=add_rule key="passwd_access" list=4 res=1
----
time->Wed Feb 12 20:16:12 2025
type=PROCTITLE msg=audit(1739373372.217:173):
proctitle=2F7362696E2F617564697463746C002D52002F6574632F61756469742F61756469742E
72756C6573
type=SOCKADDR msg=audit(1739373372.217:173): saddr=100000000000000000000000
type=SYSCALL msg=audit(1739373372.217:173): arch=c000003e syscall=44 success=yes
exit=1080 a0=3 a1=7ffc15ae8a10 a2=438 a3=0 items=0 ppid=1453 pid=1463
auid=4294967295 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0
tty=(none) ses=4294967295 comm="auditctl" exe="/usr/sbin/auditctl"
subj=unconfined key=(null)
type=CONFIG_CHANGE msg=audit(1739373372.217:173): auid=4294967295 ses=4294967295
subj=unconfined op=remove_rule key="passwd_access" list=4 res=1
----
```

```
time->Wed Feb 12 20:17:44 2025
type=PROCTITLE msg=audit(1739373464.575:193):
proctitle=2F7362696E2F617564697463746C002D52002F6574632F61756469742F61756469742E
72756C6573
type=SYSCALL msg=audit(1739373464.575:193): arch=c000003e syscall=44 success=yes
exit=1080 a0=3 a1=7ffdacf66470 a2=438 a3=0 items=0 ppid=1527 pid=1542
auid=4294967295 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0
tty=(none) ses=4294967295 comm="auditctl" exe="/usr/sbin/auditctl"
subj=unconfined key=(null)
type=CONFIG_CHANGE msg=audit(1739373464.575:193): auid=4294967295 ses=4294967295
subj=unconfined op=add_rule key="passwd_access" list=4 res=1
----
time->Wed Feb 12 20:18:21 2025
type=PROCTITLE msg=audit(1739373501.873:199):
proctitle=2F7362696E2F757365726164002D64002F686F6D652F6E65777573657231002D6700
31303032002D73002F62696E2F62617368002D750031303032006E65777573657231
type=PATH msg=audit(1739373501.873:199): item=0 name="/etc/passwd" inode=2355579
dev=08:01 mode=0100644 ouid=0 ogid=0 rdev=00:00 nametype=NORMAL cap_fp=0
cap_fi=0 cap_fe=0 cap_fver=0 cap_frootid=0
type=CWD msg=audit(1739373501.873:199): cwd="/root"
type=SYSCALL msg=audit(1739373501.873:199): arch=c000003e syscall=257
success=yes exit=5 a0=ffffff9c a1=55a12516cec0 a2=20902 a3=0 items=1 ppid=1551
pid=1558 auid=1000 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0
tty=pts1 ses=2 comm="useradd" exe="/usr/sbin/useradd" subj=unconfined
key="passwd_access"
----
time->Wed Feb 12 20:18:22 2025
type=PROCTITLE msg=audit(1739373502.042:202):
proctitle=2F7362696E2F757365726164002D64002F686F6D652F6E65777573657231002D6700
31303032002D73002F62696E2F62617368002D750031303032006E65777573657231
type=PATH msg=audit(1739373502.042:202): item=4 name="/etc/passwd" inode=2355712
dev=08:01 mode=0100644 ouid=0 ogid=0 rdev=00:00 nametype=CREATE cap_fp=0
cap_fi=0 cap_fe=0 cap_fver=0 cap_frootid=0
type=PATH msg=audit(1739373502.042:202): item=3 name="/etc/passwd" inode=2355579
dev=08:01 mode=0100644 ouid=0 ogid=0 rdev=00:00 nametype=DELETE cap_fp=0
cap_fi=0 cap_fe=0 cap_fver=0 cap_frootid=0
type=PATH msg=audit(1739373502.042:202): item=2 name="/etc/passwd+"
inode=2355712 dev=08:01 mode=0100644 ouid=0 ogid=0 rdev=00:00 nametype=DELETE
cap_fp=0 cap_fi=0 cap_fe=0 cap_fver=0 cap_frootid=0
type=PATH msg=audit(1739373502.042:202): item=1 name="/etc/" inode=2354689
dev=08:01 mode=040755 ouid=0 ogid=0 rdev=00:00 nametype=PARENT cap_fp=0 cap_fi=0
cap_fe=0 cap_fver=0 cap_frootid=0
type=PATH msg=audit(1739373502.042:202): item=0 name="/etc/" inode=2354689
dev=08:01 mode=040755 ouid=0 ogid=0 rdev=00:00 nametype=PARENT cap_fp=0 cap_fi=0
cap_fe=0 cap_fver=0 cap_frootid=0
type=CWD msg=audit(1739373502.042:202): cwd="/root"
type=SYSCALL msg=audit(1739373502.042:202): arch=c000003e syscall=82 success=yes
exit=0 a0=7fffe23ad9e0 a1=55a12516cec0 a2=7fffe23ad950 a3=100 items=5 ppid=1551
pid=1558 auid=1000 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0
tty=pts1 ses=2 comm="useradd" exe="/usr/sbin/useradd" subj=unconfined
key="passwd_access"
----
time->Wed Feb 12 20:18:27 2025
type=PROCTITLE msg=audit(1739373507.552:207):
proctitle=2F62696E2F6368666E006E65777573657231
type=PATH msg=audit(1739373507.552:207): item=0 name="/etc/passwd" inode=2355712
dev=08:01 mode=0100644 ouid=0 ogid=0 rdev=00:00 nametype=NORMAL cap_fp=0
cap_fi=0 cap_fe=0 cap_fver=0 cap_frootid=0
type=CWD msg=audit(1739373507.552:207): cwd="/root"
```

```
type=SYSCALL msg=audit(1739373507.552:207): arch=c000003e syscall=257
success=yes exit=4 a0=ffffff9c a1=562b66c25700 a2=20902 a3=0 items=1 ppid=1551
pid=1568 auid=1000 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0
tty=pts1 ses=2 comm="chfn" exe="/usr/bin/chfn" subj=unconfined
key="passwd_access"
----
time->Wed Feb 12 20:18:27 2025
type=PROCTITLE msg=audit(1739373507.596:208):
proctitle=2F62696E2F6368666E006E65777573657231
type=PATH msg=audit(1739373507.596:208): item=4 name="/etc/passwd" inode=2355570
dev=08:01 mode=0100644 ouid=0 ogid=0 rdev=00:00 nametype=CREATE cap_fp=0
cap_fi=0 cap_fe=0 cap_fver=0 cap_frootid=0
type=PATH msg=audit(1739373507.596:208): item=3 name="/etc/passwd" inode=2355712
dev=08:01 mode=0100644 ouid=0 ogid=0 rdev=00:00 nametype=DELETE cap_fp=0
cap_fi=0 cap_fe=0 cap_fver=0 cap_frootid=0
type=PATH msg=audit(1739373507.596:208): item=2 name="/etc/passwd+"
inode=2355570 dev=08:01 mode=0100644 ouid=0 ogid=0 rdev=00:00 nametype=DELETE
cap_fp=0 cap_fi=0 cap_fe=0 cap_fver=0 cap_frootid=0
type=PATH msg=audit(1739373507.596:208): item=1 name="/etc/" inode=2354689
dev=08:01 mode=040755 ouid=0 ogid=0 rdev=00:00 nametype=PARENT cap_fp=0 cap_fi=0
cap_fe=0 cap_fver=0 cap_frootid=0
type=PATH msg=audit(1739373507.596:208): item=0 name="/etc/" inode=2354689
dev=08:01 mode=040755 ouid=0 ogid=0 rdev=00:00 nametype=PARENT cap_fp=0 cap_fi=0
cap_fe=0 cap_fver=0 cap_frootid=0
type=CWD msg=audit(1739373507.596:208): cwd="/root"
type=SYSCALL msg=audit(1739373507.596:208): arch=c000003e syscall=82 success=yes
exit=0 a0=7ffc8361b390 a1=562b66c25700 a2=7ffc8361b300 a3=100 items=5 ppid=1551
pid=1568 auid=1000 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0
tty=pts1 ses=2 comm="chfn" exe="/usr/bin/chfn" subj=unconfined
key="passwd_access"
```

Глава 8. Защита сетевых взаимодействий.

8.1 Подготовка ВМ.

1. Выключите ВМ SRV1.
2. Добавьте к SRV1 второй адаптер, который подключен во внутреннюю сеть с именем netXX, где XX — номер, который вам назначил преподаватель. После включите ВМ.
3. Определите (по MAC) как называются сетевые интерфейсы внутри ВМ SRV1.

```
$ ip -br link
lo               UNKNOWN      00:00:00:00:00:00 <LOOPBACK,UP,LOWER_UP>
enp0s3           UP           08:00:27:75:be:ef <BROADCAST,MULTICAST,UP,LOWER_UP>
enp0s8           UP           08:00:27:d6:68:ad <BROADCAST,MULTICAST,UP,LOWER_UP>
```

4. Удалите все соединения кроме lo в NetworkManager и настройте интерфейсы так, чтобы внешний интерфейс получал адрес от DHCP, а на внутреннем был назначен статический адрес 10.1.1.1/24.

```
# nmcli connection
NAME                UUID                                  TYPE      DEVICE
Wired connection 1  2962b6ba-461c-4e7e-b5ca-35c4e34bc1f5  ethernet  enp0s8
lo                  d1919858-97f1-41d8-b631-1552a7164ed1  loopback  lo

# nmcli connection delete Wired\ connection\ 1
Подключение «Wired connection 1» (2962b6ba-461c-4e7e-b5ca-35c4e34bc1f5) успешно
удалено.

# nmcli connection add con-name External type ethernet ifname enp0s3 ipv4.method
auto
Подключение «External» (acf76488-01e2-4517-af23-359f97ddfec9) успешно добавлено.

# nmcli connection add con-name Internal type ethernet ifname enp0s8 ipv4.method
manual ipv4.addresses 10.1.1.1/24
Подключение «Internal» (820385ab-30ce-4bd4-9ce9-668f5bc93425) успешно добавлено.

# ip -4 -br address
lo               UNKNOWN        127.0.0.1/8
enp0s3           UP             172.27.6.156/24
enp0s8           UP             10.1.1.1/24
```

8.2 Фильтрация пакетов

1. Проверьте текущие правила фильтрации

```
# nft list ruleset
```

Вывод команды должен быть пустой. Если это не так, то надо выяснить почему он не пустой и почистить таблицу фильтрации, например это может быть из-за включенной службы firewalld или из-за того что вы уже что-то настраивали командами.

2. Создайте таблицу для фильтрации ip пакетов.

```
# nft create table ip MyTable '{ comment "My first table"; }'
# nft list ruleset
table ip MyTable {
      comment "My first table"
}
```

3. Добавьте в таблицу базовую цепочку с именем input для входящего трафика.

```
# nft add chain MyTable input '{ type filter hook input priority filter; policy
accept; comment "Base chain for INPUT packets"; }'
```

4. Добавьте в таблицу регуляную цепочку с именем SSH-IN.

```
# nft add chain MyTable SSH-IN '{  comment "Regular chain for SSH input
control"; }'
```

5. Разрешите в цепочке input любые пакеты, которые относятся к установленным соединениям.

```
# nft add rule MyTable input ct state established,related accept
```

6. Разрешите любой входящий трафик на интерфейсе lo в цепочке input, который идет с адресов сети 127.0.0.0/8. И запретите любой другой трафик.

```
# nft add rule MyTable input iif "lo" ip saddr 127.0.0.0/8 accept
# nft add rule MyTable input iif "lo" ip saddr != 127.0.0.0/8 drop
```

7. Настройте прыжок из цепочки input в SSH-IN, когда приходит пакет на порт 22.

```
# nft add rule MyTable input tcp dport 22 jump SSH-IN
```

8. Разрешите входящий трафик для сервера SSH с адресов сетей 127.0.0.0/8 и 10.1.1.0/24. Остальные пакеты должны быть запрещены.

```
# nft add rule MyTable SSH-IN ip saddr 127.0.0.0/8 accept
# nft add rule MyTable SSH-IN ip saddr 10.1.1.0/24 accept
# nft add rule MyTable SSH-IN drop
```

9. Можете ли вы теперь открыть какую-нибудь веб страницу в интернете? И почему?

Все должно работать из-за правила, которое разрешает любые пакеты, которые относятся к установленным соединениям. Вторая причина политика, которая разрешает все по умолчанию.

10. Создайте файл с именем `nft.rules` записав в него строку flush ruleset.

```
# echo flush ruleset > nft.rules
```

11. Допишите в это же файл текущие правила.

```
# nft list ruleset >> nft.rules

# cat nft.rules
flush ruleset
table ip MyTable {
      comment "My first table"
      chain input {
```

```
            comment "Base chain for INPUT packets"
            type filter hook input priority filter; policy accept;
            ct state established,related accept
            iif "lo" ip saddr 127.0.0.0/8 accept
            iif "lo" ip saddr != 127.0.0.0/8 drop
            tcp dport 22 jump SSH-IN
      }

      chain SSH-IN {
            comment "Regular chain for SSH input control"
            ip saddr 127.0.0.0/8 accept
            ip saddr 10.1.1.0/24 accept
            drop
      }
}
```

12. Сделайте копию файла `nft.rules` и в копии за комментируйте строку с разрешением установленных соединений и измените политику цепочки input на drop. Потом примените правила. Можете ли вы открыть какой-либо сайт в браузере?

```
# cp nft.rules nft2.rules
# cat nft2.rules
flush ruleset
table ip MyTable {
      comment "My first table"
      chain input {
            comment "Base chain for INPUT packets"
            type filter hook input priority filter; policy drop;
            #ct state established,related accept
            iif "lo" ip saddr 127.0.0.0/8 accept
            iif "lo" ip saddr != 127.0.0.0/8 drop
            tcp dport 22 jump SSH-IN
      }

      chain SSH-IN {
            comment "Regular chain for SSH input control"
            ip saddr 127.0.0.0/8 accept
            ip saddr 10.1.1.0/24 accept
            drop
      }
}

# nft -f nft2.rules
```

Никакой сайт не откроется.

13. В исходном файле `nft.rules` добавьте счетчики к правилам в цепочке SSH-IN. Примените эти правила.

```
# nft -f nft.rules
root@srv1:~# cat nft.rules
flush ruleset
table ip MyTable {
      comment "My first table"
      chain input {
            comment "Base chain for INPUT packets"
            type filter hook input priority filter; policy accept;
            ct state established,related accept
            iif "lo" ip saddr 127.0.0.0/8 accept
            iif "lo" ip saddr != 127.0.0.0/8 drop
```

```
            tcp dport 22 jump SSH-IN
      }

      chain SSH-IN {
            comment "Regular chain for SSH input control"
            ip saddr 127.0.0.0/8 counter accept
            ip saddr 10.1.1.0/24 counter accept
            counter drop
      }
}
```

14. На SRV1 попробуйте подключиться по ssh к адресам 127.0.0.1 и 10.1.1.1. Оба ли подключения удались?

Подключение на адрес 10.1.1.1 не работает.

15. Добавьте еще одно правило, которое разрешает подключаться с адреса 10.1.1.1 на интерфейс lo.

```
# cat nft.rules
flush ruleset
table ip MyTable {
      comment "My first table"
      chain input {
            comment "Base chain for INPUT packets"
            type filter hook input priority filter; policy accept;
            ct state established,related accept
            iif "lo" ip saddr 127.0.0.0/8 accept
            iif "lo" ip saddr 10.1.1.1/32 accept
            iif "lo" ip saddr != 127.0.0.0/8 drop
            tcp dport 22 jump SSH-IN
      }

      chain SSH-IN {
            comment "Regular chain for SSH input control"
            ip saddr 127.0.0.0/8 counter accept
            ip saddr 10.1.1.0/24 counter accept
            counter drop
      }
}

# nft -f nft.rules
```

16. Проверьте состояние счетчиков.

```
# nft list ruleset
table ip MyTable {
      comment "My first table"
      chain input {
            comment "Base chain for INPUT packets"
            type filter hook input priority filter; policy accept;
            ct state established,related accept
            iif "lo" ip saddr 127.0.0.0/8 accept
            iif "lo" ip saddr 10.1.1.1 accept
            iif "lo" ip saddr != 127.0.0.0/8 drop
            tcp dport 22 jump SSH-IN
      }

      chain SSH-IN {
            comment "Regular chain for SSH input control"
```

```
            ip saddr 127.0.0.0/8 counter packets 0 bytes 0 accept
            ip saddr 10.1.1.0/24 counter packets 0 bytes 0 accept
            counter packets 0 bytes 0 drop
      }
}
```

Счетчики пакетов нулевые, хотя подключения были.

17. CL1 и CL2 подключите в ту же сеть в VirtualBox, что и второй адаптер SRV1. Настройте адреса соответственно 10.1.1.11/24 и 10.1.1.12/24. Шлюз по умолчанию 10.1.1.1. DNS сервер такой же как указан в файле `/etc/resolv.conf` на srv1.
18. Подключитесь с cl1 или cl2 на srv1 (10.1.1.1) по SSH. Проверьте, что счетчики начали увеличиваться согласно количеству попыток подключения.

```
# nft list ruleset  | grep counter
            ip saddr 127.0.0.0/8 counter packets 0 bytes 0 accept
            ip saddr 10.1.1.0/24 counter packets 1 bytes 60 accept
            counter packets 0 bytes 0 drop
```

19. Проверьте что ни cl1 ни cl2 не могут выйти в интернет.

```
$ ping mail.ru
ping: mail.ru: Временный сбой в разрешении имен
```

8.3 NAT.

1. Очистите все правила на srv1.

```
# nft flush ruleset
```

2. Включите пересылку пакетов на srv1.

```
# grep ip_forward /etc/sysctl.conf
net.ipv4.ip_forward=1

# sysctl -p /etc/sysctl.conf
net.ipv4.ip_forward = 1
```

3. Добавьте таблицу с именем NAT.

```
# nft add table ip NAT
```

4. В таблицу NAT цепочку типа nat, хуком postrouting и именем s-nat.

```
# nft create chain NAT s-nat '{ type nat hook postrouting priority 0 ; comment
"My SNAT chain" ; }'
```

5. В цепочку s-nat добавьте правило для маскарадинга всех пакетов из сети 10.1.1.0/24 и исходящих с внешнего интерфейса.

```
# nft add rule NAT s-nat oif enp0s3 ip saddr 10.1.1.0/24 masquerade
# nft list ruleset
table ip NAT {
      chain s-nat {
            comment "My SNAT chain"
            type nat hook postrouting priority filter; policy accept;
            oif "enp0s3" ip saddr 10.1.1.0/24 masquerade
      }
}
```

6. Проверьте, что cl1 и cl2 могут выходить в интернет.

```
$ ping -c1 mail.ru
PING mail.ru (94.100.180.200) 56(84) bytes of data.
64 bytes from mail.ru (94.100.180.200): icmp_seq=1 ttl=56 time=82.9 ms

--- mail.ru ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 82.874/82.874/82.874/0.000 ms
```

7. Опубликуйте службу SSH на cl1 и cl2 с портами соответственно 22011 и 22012 на srv1.

```
# nft create chain NAT d-nat '{ type nat hook prerouting priority 0 ; comment
"My DNAT chain" ; }'

#  nft add rule NAT s-nat oif enp0s3 ip saddr 10.1.1.11 tcp sport 22 snat to
:22011
# nft add rule NAT d-nat iif enp0s3 tcp dport 22011 dnat to 10.1.1.11:22

# nft list ruleset
table ip NAT {
      chain s-nat {
            comment "My SNAT chain"
            type nat hook postrouting priority filter; policy accept;
            oif "enp0s3" ip saddr 10.1.1.0/24 masquerade
            oif "enp0s3" ip saddr 10.1.1.11 tcp sport 22 snat to :22011
      }

      chain d-nat {
```

```
            comment "My DNAT chain"
            type nat hook prerouting priority filter; policy accept;
            iif "enp0s3" tcp dport 22011 dnat to 10.1.1.11:22
      }
}

$ ssh sa@172.27.6.156 -p 22011
sa@172.27.6.156's password:
Linux cl1 6.1.0-25-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.106-3 (2024-08-26)
x86_64
---

#  nft add rule NAT s-nat oif enp0s3 ip saddr 10.1.1.12 tcp sport 22 snat to
:22012
# nft add rule NAT d-nat iif enp0s3 tcp dport 22012 dnat to 10.1.1.12:22
```

8. Сохраните текущие правила фильтрации в файл `/etc/mynftd.rules`.

```
# nft list ruleset > /etc/mynftd.rules
```

9. Создайте скрипт, который будет управлять правилами фильтрации с именем `/usr/local/sbin/mynftd`, сделайте файл исполняемым.

```
# cat /usr/local/sbin/mynftd
#!/bin/bash
NFT="/sbin/nft"
CONFIG=/etc/mynftd.rules
test -r /etc/mynftd.rules || { echo "There is no config $CONFIG" ; exit 1; }
case $1 in
  start)
      $NFT flush ruleset
      $NFT -f $CONFIG
  ;;
  stop)
      $NFT flush ruleset
  ;;
  save)
      $NFT list ruleset > $CONFIG
  ;;
esac

# chmod a+x /usr/local/sbin/mynftd
```

10. Создайте службу systemd, которая будет запускать `/usr/local/sbin/mynftd` во время старта ОС.

```
# nano /etc/systemd/system/mynftd.service
# systemctl daemon-reload
# systemctl enable mynftd.service
Created symlink /etc/systemd/system/network.target.wants/mynftd.service →
/etc/systemd/system/mynftd.service.

# cat /etc/systemd/system/mynftd.service
[Unit]
Desctiption = Simple service for loading nft rules to kernel

[Service]
Type=oneshot
RemainAfterExit=yes
```

```
ExecStart=/usr/local/sbin/mynftd start
ExecStop=/usr/local/sbin/mynftd stop

[Install]
WantedBy=network.target

# reboot

# nft list ruleset
table ip NAT {
      chain s-nat {
            comment "My SNAT chain"
            type nat hook postrouting priority filter; policy accept;
            oif "enp0s3" ip saddr 10.1.1.0/24 masquerade
            oif "enp0s3" ip saddr 10.1.1.11 tcp sport 22 snat to :22011
            oif "enp0s3" ip saddr 10.1.1.12 tcp sport 22 snat to :22012
      }

      chain d-nat {
            comment "My DNAT chain"
            type nat hook prerouting priority filter; policy accept;
            iif "enp0s3" tcp dport 22011 dnat to 10.1.1.11:22
            iif "enp0s3" tcp dport 22012 dnat to 10.1.1.12:22
      }
}
```

8.4 *Firewalld (необязательное).

1. Остановите сервис mynftd. И запретите его запуск.
2. Если сервис firewalld еще не установлен, то установите и запустите его.
3. Посмотрите какие имеются зоны. Какие интерфейсы в этих зонах и какие службы могут сейчас работать.
4. Переопределите внутренний интерфейс в зону work. Какие теперь службы разрешены?
5. Переведите внешний интерфейс в зону external.
6. Дайте разрешение для всех служб и портов, которые вам необходимы.
7. Настройте NAT маскарадинг. Проверьте, что cl1 и cl2 могут выходить в интернет.
8. Остановите и запретите запуск службы firewalld.
9. Вновь включите и запустите службу mynftd. Проверьте, что клиенты могут выходить в интернет.

8.5 Прокси сервер squid.

1. Остановите службу mynftd. Проверьте, что cl1 и cl2 не могут выходить в интернет.

```
# systemctl stop mynftd.service
```

2. Установите прокси сервер squid на srv1.

```
# apt install squid
```

3. Сохраните исходный файл конфигурации squid и создайте новый удалив все строки с комментариями и повторяющиеся строки.

```
# cp /etc/squid/squid.conf /etc/squid/squid.conf.orig
# grep -v '^#' /etc/squid/squid.conf.orig | uniq > /etc/squid/squid.conf
```

4. Создайте ACL для адресов источника с именем mynet и адресом 10.1.1.0/24 и предоставьте доступ к интернету для этого списка. Удалите ACL localnet.

```
# egrep 'mynet|localnet|http_access' /etc/squid/squid.conf
acl mynet src 10.1.1.0/24
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports
http_access allow localhost manager
http_access deny manager
http_access allow localhost
http_access allow mynet
http_access deny all

# squid -k reconfigure
```

5. Настройте клиентов для работы с прокси сервером.

```
$ sudo cat /etc/apt/apt.conf.d/70proxy
Acquire::HTTPS::proxy "http://10.1.1.1:3128";
Acquire::HTTP::proxy "http://10.1.1.1:3128";
Acquire::FTP::proxy "http://10.1.1.1:3128";

$ sudo apt update
Пол:1 http://security.debian.org/debian-security bookworm-security InRelease
[48,0 kB]
Сущ:2 http://deb.debian.org/debian bookworm InRelease
Пол:3 http://deb.debian.org/debian bookworm-updates InRelease [55,4 kB]
Пол:4 http://security.debian.org/debian-security bookworm-security/main amd64
Packages [245 kB]
Пол:5 http://security.debian.org/debian-security bookworm-security/main
Translation-en [146 kB]
Получено 494 kB за 2с (286 kB/s)
Чтение списков пакетов… Готово
Построение дерева зависимостей… Готово
Чтение информации о состоянии… Готово
Может быть обновлён 151 пакет. Запустите «apt list --upgradable» для показа

$ cat ~/.wgetrc
https_proxy = http://10.1.1.1:3128/
http_proxy = http://10.1.1.1:3128/
ftp_proxy = http://10.1.1.1:3128/
use_proxy = on

$ wget http://mail.ru -O /dev/null
--2025-02-13 16:08:16--  http://mail.ru/
Подключение к 10.1.1.1:3128... соединение установлено.
```

```
Proxy-запрос отправлен. Ожидание ответа… 301 Moved Permanently
Адрес: https://mail.ru/ [переход]
--2025-02-13 16:08:16--  https://mail.ru/
Подключение к 10.1.1.1:3128... соединение установлено.
Proxy-запрос отправлен. Ожидание ответа… 200 OK
Длина: нет данных [text/html]
Сохранение в: «/dev/null»

/dev/null               [  <=>               ] 376,55K  1,72MB/s    за 0,2s

2025-02-13 16:08:16 (1,72 MB/s) - «/dev/null» сохранён [385586]
```

Для браузера настройка прокси производится в меню настройек программы.

6. Определите URL ACL с регулярным выражением google. Запретите доступ к страницам подпадающим под действие созданного ACL

```
# grep google /etc/squid/squid.conf
acl google url_regex google
http_access deny google

# squid -k reconfigure

sa@cl1:~$ wget http://google.com -O /dev/null
--2025-02-13 16:16:53--  http://google.com/
Подключение к 10.1.1.1:3128... соединение установлено.
Proxy-запрос отправлен. Ожидание ответа… 403 Forbidden
2025-02-13 16:16:53 ОШИБКА 403: Forbidden.
```

7. Установите максимальный размер запроса клиента равным 10кб.

```
# grep request /etc/squid/squid.conf
request_body_max_size 10 KB
```

8. Установите ограничение скорости получения информации для файлов с расширением zip на величину 100кбит/с.

```
# egrep 'zip|delay' /etc/squid/squid.conf
acl zip_files rep_mime_type -i ^application/zip$
delay_pools 1
delay_class 1 3
delay_access 1 allow zip_files
delay_access 1 deny all
delay_parameters 1 -1/-1 -1/-1 12500/80000

sa@cl1:~$ wget http://172.27.255.26/ftp/pub/aldpro_2.1.0/ALDPro_docs.zip
--2025-02-13 16:40:08--
http://172.27.255.26/ftp/pub/aldpro_2.1.0/ALDPro_docs.zip
Подключение к 10.1.1.1:3128... соединение установлено.
Proxy-запрос отправлен. Ожидание ответа… 200 OK
Длина: 53631011 (51M) [application/zip]
Сохранение в: «ALDPro_docs.zip»

ALDPro_docs.zip       0%[                    ] 455,74K  11,7KB/s    ост 63m 47s^
```

9. Настройте базовую аутентификацию на вашем сервере.

```
# htpasswd -c /etc/squid/squid.passwd squiduser
New password:
Re-type new password:
Adding password for user squiduser

# egrep 'password|auth' /etc/squid/squid.conf
```

```
auth_param basic program /usr/lib/squid/basic_ncsa_auth /etc/squid/squid.passwd
acl password proxy_auth REQUIRED
http_access allow password

sa@cl1:~$ cat .wgetrc
https_proxy = http://squiduser:123@10.1.1.1:3128/
http_proxy = http://squiduser:123@10.1.1.1:3128/
ftp_proxy = http://squiduser:123@10.1.1.1:3128/
use_proxy = on

sa@cl1:~$ wget http://google.com -O /dev/null
--2025-02-13 16:45:43--  http://google.com/
Подключение к 10.1.1.1:3128... соединение установлено.
Proxy-запрос отправлен. Ожидание ответа… 301 Moved Permanently
Адрес: http://www.google.com/ [переход]
--2025-02-13 16:45:43--  http://www.google.com/
Повторное использование соединения с 10.1.1.1:3128.
Proxy-запрос отправлен. Ожидание ответа… 200 OK
Длина: нет данных [text/html]
Сохранение в: «/dev/null»

/dev/null               [ <=>                ]  20,99K  --.-KB/s    за 0,05s

2025-02-13 16:45:44 (450 KB/s) - «/dev/null» сохранён [21496]
```

10. Отключите аутентификацию, запрет google и пулы задержки.

```
# egrep '^#' /etc/squid/squid.conf
#auth_param basic program /usr/lib/squid/basic_ncsa_auth /etc/squid/squid.passwd
#acl google url_regex google
#acl zip_files rep_mime_type -i ^application/zip$
#delay_pools 1
#delay_class 1 3
#delay_access 1 allow zip_files
#delay_access 1 deny all
#delay_parameters 1 -1/-1 -1/-1 12500/80000
#acl password proxy_auth REQUIRED
#http_access allow password
#http_access deny google

# squid -k reconfigure
```

11. Запустите службу mynftd. Проверьте, что cl1 и cl2 могут выходить в интернет.

```
# systemctl start mynftd.service

sa@cl1:~$ rm .wgetrc
sa@cl1:~$ wget http://google.com -O /dev/null
--2025-02-13 16:50:31--  http://google.com/
Подключение к 10.1.1.1:3128... соединение установлено.
Proxy-запрос отправлен. Ожидание ответа… 301 Moved Permanently
Адрес: http://www.google.com/ [переход]
--2025-02-13 16:50:31--  http://www.google.com/
Повторное использование соединения с 10.1.1.1:3128.
Proxy-запрос отправлен. Ожидание ответа… 200 OK
Длина: нет данных [text/html]
Сохранение в: «/dev/null»

/dev/null               [ <=>                ]  20,96K  --.-KB/s    за 0,04s

2025-02-13 16:50:32 (522 KB/s) - «/dev/null» сохранён [21460]
```

8.6 SSH.

1. Проверьте установлена ли и запущена служба SSH.

```
sa@srv1:~$ systemctl status ssh
```

2. Проверьте работоспособность команд `ssh` и `scp`.

```
sa@srv1:~$ ssh 10.1.1.11
sa@10.1.1.11's password:
Linux cl1 6.1.0-25-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.106-3 (2024-08-26)
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.
You have new mail.
Last login: Thu Feb 13 15:57:00 2025 from 10.1.1.1
sa@cl1:~$ exit

sa@srv1:~$ echo test > test.txt
sa@srv1:~$ scp test.txt 10.1.1.11:/home/sa
sa@10.1.1.11's password:
test.txt                                      100%    5     0.5KB/s   00:00
```

3. Создайте пару ключей RSA и поместите публичный ключ на желаемый сервер. Оставьте парольную фразу пустой - в таком случае при успешном обмене ключами вводить пароль для доступа к удаленному узлу не буде требоваться (но сам ключ будет не защищен).

```
sa@cl1:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/sa/.ssh/id_rsa): Enter passphrase
(empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/sa/.ssh/id_rsa
Your public key has been saved in /home/sa/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:uS92XctdaxSzU/k0WKhYuSrTPIjXovPQUdvj2ZNgxcY sa@cl1
The key's randomart image is:
+---[RSA 3072]----+
|            . .  |
|           oo. . |
|          .o oEo .|
|         .oooo. =o|
|       ..S..=   .B|
|      ..*.Bo = o+o|
|      .o.= .+ *.o+|
|     o. o.. . +o.|
|      oo o.   .  |
+----[SHA256]-----+

sa@cl1:~$ grep srv1 /etc/hosts
10.1.1.1    srv1

sa@cl1:~$ ssh-copy-id srv1
```

```
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed:
"/home/sa/.ssh/id_rsa.pub"
The authenticity of host 'srv1 (10.1.1.1)' can't be established.
ED25519 key fingerprint is SHA256:Q138Pxa07MEo+jqRq8ndjbdNa1wdou0R0Vmc60nMTDs.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:1: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are
prompted now it is to install the new keys
sa@srv1's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'srv1'"
and check to make sure that only the key(s) you wanted were added.

sa@cl1:~$ ssh srv1
Linux srv1 6.1.0-31-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.128-1 (2025-02-07)
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: Thu Feb 13 15:56:51 2025 from 192.168.97.242

sa@srv1:~$ exit
```

4. Защитите созданный ключ паролем.

```
sa@cl1:~$ ssh-keygen -p
Enter file in which the key is (/home/sa/.ssh/id_rsa):
Key has comment 'sa@cl1'
Enter new passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved with the new passphrase.

sa@cl1:~$ ssh srv1
Enter passphrase for key '/home/sa/.ssh/id_rsa':
---
```

5. Проверьте, что теперь при каждом входе на сервер у вас будут спрашивать пароль для разблокирования ключа.
6. Войдите в систему в графический сеанс и запустите эмулятор терминала. Добавьте частный ключ в связку ключей ssh агента. Пароль будет запрошен только один раз в момент добавления ключа.

```
sa@cl1:~$ ssh-add
Enter passphrase for /home/sa/.ssh/id_rsa:
Identity added: /home/sa/.ssh/id_rsa (sa@cl1)
sa@cl1:~$ ssh-add -l
3072 SHA256:uS92XctdaxSzU/k0WKhYuSrTPIjXovPQUdvj2ZNgxcY sa@cl1 (RSA)
```

7. Войдите в систему не используя графического сеанса и проверьте, что агент в этом случае недоступен.

```
sa@cl1:~$ ssh-add -l
Could not open a connection to your authentication agent.
```

8. Настройте запуск ssh агента:

```
sa@cl1:~$ cat /etc/systemd/user/ssh-agent.service
[Unit]
Description=SSH key agent

[Service]
Type=simple
Environment=SSH_AUTH_SOCK=%t/ssh-agent.socket
ExecStart=/usr/bin/ssh-agent -D -a $SSH_AUTH_SOCK

[Install]
WantedBy=default.target

sa@cl1:~$ cat /etc/profile.d/ssh-agent.sh
export SSH_AUTH_SOCK="$XDG_RUNTIME_DIR/ssh-agent.socket"

user@cl1:~$ systemctl --user status ssh-agent.service
○ ssh-agent.service - SSH key agent
     Loaded: loaded (/etc/xdg/systemd/user/ssh-agent.service; disabled; preset:>
     Active: inactive (dead)

фев 13 18:33:49 cl1 systemd[1912]: ssh-agent.service: Failed to open /etc/xdg/s>
фев 13 18:34:15 cl1 systemd[1912]: ssh-agent.service: Failed to open /etc/xdg/s>
user@cl1:~$ systemctl --user enable --now ssh-agent.service
Created symlink /home/user/.config/systemd/user/default.target.wants/ssh-
agent.service → /etc/xdg/systemd/user/ssh-agent.service.
user@cl1:~$ systemctl --user status ssh-agent.service
● ssh-agent.service - SSH key agent
     Loaded: loaded (/etc/xdg/systemd/user/ssh-agent.service; enabled; preset: >
     Active: active (running) since Thu 2025-02-13 18:35:15 +05; 2s ago
   Main PID: 2140 (ssh-agent)
      Tasks: 1 (limit: 2284)
     Memory: 864.0K
        CPU: 22ms
     CGroup: /user.slice/user-1002.slice/user@1002.service/app.slice/ssh-agent.>
             └─2140 /usr/bin/ssh-agent -D -a /run/user/1002/ssh-agent.socket

фев 13 18:35:15 cl1 systemd[2048]: Started ssh-agent.service - SSH key agent.
фев 13 18:35:15 cl1 ssh-agent[2140]: SSH_AUTH_SOCK=/run/user/1002/ssh-agent.soc>
фев 13 18:35:15 cl1 ssh-agent[2140]: echo Agent pid 2140;

user@cl1:~$ ssh-add
Enter passphrase for /home/user/.ssh/id_rsa:
Identity added: /home/user/.ssh/id_rsa (user@cl1)
user@cl1:~$ ssh-add -l
3072 SHA256:ynt4JoIMUs9+eFgb9GuwIexPi7PKmMtUskDnWrEqmAY user@cl1 (RSA)
```

9. Запретите вход пользователю root с помощью пароля.

```
sa@srv1:~$ grep Root /etc/ssh/sshd_config
PermitRootLogin prohibit-password
```

10. Измените порт 22 на порт 33322.

```
sa@srv1:~$ grep ^Port /etc/ssh/sshd_config
Port 33322
```

11. Настройте клиентскую машину так, чтобы она по молчанию подключалась к серверу по порту 33322.

```
sa@cl1:~$ cat .ssh/config
Host srv1
      HostName 10.1.1.1
      Port 33322
```

12. *Установите пакет fail2ban и настройте блокировку адресов, которые пытаются подобрать пароль пользователей по ssh.

```
sa@srv1:~# apt install fail2ban

root@srv1:~# nano /etc/fail2ban/jail.local
[DEFAULT]
banaction = nftables-multiport
banaction_allports = nftables[type=allports]

[sshd]
backend=systemd
enabled = true

root@srv1:~# systemctl restart fail2ban.service
```

Сделать несколько (по умолчанию 5) попыток войти с неправильным паролем на srv1. Потом снять блокировку адреса.

```
sa@cl1:~$ ssh nouser@srv1

root@srv1:~# fail2ban-client status
Status
|- Number of jail:      1
`- Jail list:     sshd
root@srv1:~# fail2ban-client banned
[{'sshd': ['10.1.1.11']}]

root@srv1:~# fail2ban-client unban 10.1.1.11
```

Добавить дополнительную защиту и настройки.

```
root@srv1:~# nano /etc/fail2ban/jail.local
[DEFAULT]
bantime.increment = true
bantime.multipliers = 1 5 30 60 300 720 1440 2880

ignoreip = 127.0.0.1/8 ::1
maxretry = 3

banaction = nftables-multiport
banaction_allports = nftables[type=allports]

[sshd]
backend=systemd
enabled = true

root@srv1:~# systemctl restart fail2ban.service
root@srv1:~# fail2ban-client status
Status
|- Number of jail:      1
`- Jail list:     sshd
```

Глава 9. Инфраструктура открытых ключей на основе openssl.

9.1 Создание CA.

1. В качестве корневого CA будет использоваться srv1.
2. Создайте сертификат корневого CA.

Создаем в `/srv/pki/CA` структуру каталогов и нужные файлы

```
root@srv1:~# mkdir -p /srv/pki/CA && cd /srv/pki/CA
root@srv1:/srv/pki/CA# mkdir certs crl newcerts private
root@srv1:/srv/pki/CA# echo "01" > serial
root@srv1:/srv/pki/CA# > index.txt
root@srv1:/srv/pki/CA# cd ..
```

Копируем в каталог CA конфигурационный файл `openssl` – `openssl.cnf`

```
root@srv1:/srv/pki# cp /etc/ssl/openssl.cnf .
```

Вносим изменения в `openssl.cnf` (например, исправляем в секции `[CA_default]` переменную `dir` – `dir = ./CA`). Обратите внимание и на следующие опции: `countryName_default`, `stateOrProvinceName_default`, `localityName_default`, `0.organizationName_default`, `organizationalUnitName_default`.

Так же настраиваем опции в файле `openssl.cnf` следующим образом:

```
[ CA_default ]

dir             = /srv/pki/CA           # Where everything is kept
certs           = $dir/certs            # Where the issued certs are kept
crl_dir         = $dir/crl              # Where the issued crl are kept
database        = $dir/index.txt        # database index file.
#unique_subject = no                    # Set to 'no' to allow creation of
                                        # several certs with same subject.
new_certs_dir   = $dir/newcerts         # default place for new certs.

certificate     = $dir/certs/ca.crt             # The CA certificate
serial          = $dir/serial           # The current serial number
crlnumber       = $dir/crlnumber        # the current crl number
                                        # must be commented out to leave a V1
CRL
crl             = $dir/crl.pem          # The current CRL
private_key     = $dir/private/ca.key   # The private key

x509_extensions = usr_cert              # The extensions to add to the cert
copy_extensions = copyall               # ОЧЕНЬ ОПАСНО!!!

# Пропущено несколько строк
[ policy_match ]
countryName             = match
stateOrProvinceName     = optional
```

```
organizationName          = optional
organizationalUnitName    = optional
commonName                = supplied
emailAddress              = optional

# Пропущено несколько строк
[ req_distinguished_name ]
countryName                     = Country Name (2 letter code)
countryName_default             = RU
countryName_min                 = 2
countryName_max                 = 2

stateOrProvinceName             = State or Province Name (full name)
stateOrProvinceName_default     = Sverdlovsk

localityName                    = Locality Name (eg, city)
й
0.organizationName              = Organization Name (eg, company)
0.organizationName_default      = IT Cloud

# we can do this but it is not needed normally :-)
#1.organizationName             = Second Organization Name (eg, company)
#1.organizationName_default     = World Wide Web Pty Ltd

organizationalUnitName          = Organizational Unit Name (eg, section)
organizationalUnitName_default  = Class SecurL
```

Создаем приватный ключ агентства:

```
root@srv1:/srv/pki# openssl genrsa -aes256 -out CA/private/ca.key 2048
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
```

Далее создаем самоподписанный сертификат агентства. Так как мы не используем каталог по умолчанию определенный для данной версии, то нам стоит определить переменную OPENSSL_CONF, которая укажет на собственный конфигурационный файл:

```
root@srv1:/srv/pki# openssl version -d
OPENSSLDIR: "/usr/lib/ssl"
root@srv1:/srv/pki# ls -l /usr/lib/ssl/openssl.cnf
lrwxrwxrwx. 1 root root 20 окт 27 19:16 /usr/lib/ssl/openssl.cnf ->
/etc/ssl/openssl.cnf

root@srv1:/srv/pki# export OPENSSL_CONF=/srv/pki/openssl.cnf

root@srv1:/srv/pki# openssl req -new -x509 -days 36525 -key CA/private/ca.key
-out CA/certs/ca.crt
Enter pass phrase for CA/private/ca.key:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [RU]:
State or Province Name (full name) [Sverdlovsk]:
Locality Name (eg, city) [Ykaterinburg]:
Organization Name (eg, company) [IT Cloud]:
```

```
Organizational Unit Name (eg, section) [Class SecurL]:
Common Name (e.g. server FQDN or YOUR name) []:My CA cert
Email Address []:
```

Проверяем сертификат:

```
root@srv1:/srv/pki# openssl x509 -in CA/certs/ca.crt -text -noout
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            37:b8:96:9c:cc:be:7b:df:0d:6c:c8:5e:0e:ca:eb:6b:42:da:a4:09
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: C = RU, ST = Sverdlovsk, L = Ykaterinburg, O = IT Cloud, OU =
Class SecurL, CN = My CA cert
        Validity
            Not Before: Feb  6 17:02:51 2025 GMT
            Not After : Feb  7 17:02:51 2125 GMT
        Subject: C = RU, ST = Sverdlovsk, L = Ykaterinburg, O = IT Cloud, OU =
Class SecurL, CN = My CA cert
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:c8:3d:01:cc:9d:c2:02:c2:11:23:9d:6e:2e:c3:
                    f3:5d:b9:01:48:ff:1a:80:03:d2:0b:42:a0:54:f8:
                    c3:6a:32:e2:8d:5c:bd:f4:e7:17:56:7d:1d:5b:09:
                    73:60:6d:99:fc:b3:d5:3e:82:3a:50:fb:dd:64:09:
                    94:65:21:31:81:1c:af:32:9f:90:b8:9d:57:4d:28:
                    91:df:67:2a:df:89:c7:60:ca:7e:79:66:f6:ed:5c:
                    49:1e:6f:f0:d1:14:08:44:fd:bf:6d:d4:02:5e:54:
                    9f:78:7d:61:2e:45:be:18:be:24:17:d8:0e:0d:3d:
                    a6:ae:f4:97:b9:91:9d:16:84:ef:50:8a:d6:ad:83:
                    b4:df:af:29:47:bf:32:c2:fe:bf:46:ce:81:eb:d3:
                    18:79:38:3a:83:72:58:05:c8:82:09:8a:ab:47:d5:
                    19:22:93:8a:84:be:fd:b1:e5:c4:a2:e2:5f:e9:ea:
                    55:36:2e:f1:c1:88:de:4f:d3:f0:83:34:d9:a7:0e:
                    a2:6a:56:a4:16:97:05:5f:ea:c2:3b:91:77:c0:f8:
                    f5:c5:2a:a6:d6:a8:b9:0e:3b:71:cc:d9:99:1d:1e:
                    fd:fb:b9:ef:b4:4d:76:68:f6:ad:41:7c:51:6d:8c:
                    1a:93:90:1d:fa:cf:a1:2f:f0:ae:8a:b8:1b:ad:3b:
                    09:f1
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Subject Key Identifier:
                FD:A4:25:32:14:F0:0A:3A:E9:69:B1:2F:F3:3A:86:DC:1C:09:38:1F
            X509v3 Authority Key Identifier:
                FD:A4:25:32:14:F0:0A:3A:E9:69:B1:2F:F3:3A:86:DC:1C:09:38:1F
            X509v3 Basic Constraints: critical
                CA:TRUE
    Signature Algorithm: sha256WithRSAEncryption
    Signature Value:
        c2:f7:b4:65:30:94:64:c6:bd:fa:f1:54:cc:4c:9a:b1:11:87:
        7b:39:87:9a:f3:1d:8a:b4:79:9b:14:23:88:32:ce:28:7d:a4:
        17:fb:f2:dc:49:c3:cf:0c:8a:dd:93:16:3d:df:1c:f2:e7:f3:
        8b:14:24:b5:09:9f:36:60:03:ef:18:41:08:aa:e4:29:0a:6b:
        5a:f0:40:de:61:fa:f1:7f:b3:f0:eb:c4:25:2d:e2:e9:c9:49:
        7b:69:26:03:68:88:62:f8:74:bf:06:00:4c:4a:43:62:06:81:
```

```
        f5:7a:d4:59:24:cc:08:6f:a8:55:dc:4d:f5:5d:e3:d0:92:64:
        f7:80:ec:55:eb:6a:d6:d8:89:27:ed:23:c7:f9:57:ce:61:42:
        ef:df:e1:70:01:27:78:c5:e7:dc:9f:d4:6b:7a:2f:cf:a6:e7:
        cc:d0:4f:6f:7e:a7:68:13:f5:f4:dd:5d:ef:f6:8d:73:4b:ed:
        ab:19:f9:f2:22:80:09:28:98:e1:ef:56:a9:a8:12:04:2f:39:
        8b:32:ae:44:35:7d:a5:7f:6a:d4:79:ff:a7:51:35:b2:01:70:
        b3:9b:47:88:07:4e:f2:ad:9a:70:7c:b1:57:94:27:88:32:86:
        b6:2c:bd:1a:6d:9f:70:7f:6e:92:5b:3e:df:47:fb:ab:10:c0:
        f9:68:83:58
```

3. Добавьте сертификат вашего CA в список доверенных

```
root@srv1:/srv/pki# mkdir /usr/local/share/ca-certificates/myca
root@srv1:/srv/pki# cp CA/certs/ca.crt /usr/local/share/ca-certificates/myca
root@srv1:/srv/pki# update-ca-certificates
Updating certificates in /etc/ssl/certs...
rehash: warning: skipping ca-certificates.crt,it does not contain exactly one
certificate or CRL
1 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
```

Проверим попал ли сертификат в список доверенных:

```
root@srv1:/srv/pki# openssl crl2pkcs7 -nocrl -certfile /etc/ssl/certs/ca-
certificates.crt | openssl pkcs7 -print_certs -noout | grep subject | grep 'My
CA cert'
subject=C = RU, ST = Sverdlovsk, L = Ykaterinburg, O = IT Cloud, OU = Class
SecurL, CN = My CA cert
```

9.2 Использование сертификатов.

1. Создайте частный ключ для будущего сертификата.

```
root@srv1:/etc/ssl# openssl genrsa -out /etc/ssl/private/srv1.key 2048
```

2. Запросите сертификат для srv1 с именами srv1, srv1.class.itcloud и для IP адреса 10.1.1.1.

Создаем запрос на получение сертификата:

```
root@srv1:/etc/ssl# SAN='DNS:srv1,DNS:srv1.class.itcloud,IP:10.1.1.1' openssl
req -new -key /etc/ssl/private/srv1.key -out /tmp/srv1.csr -extensions v3_req
-reqexts v3_req -config <(cat /etc/ssl/openssl.cnf; echo -e '[ v3_req ]\n
subjectAltName = ${ENV::SAN}\n' )
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [RU]:
State or Province Name (full name) [Свердловская область]:
Locality Name (eg, city) [Екатеринбург]:
Organization Name (eg, company) [АйТи Клауд]:
Organizational Unit Name (eg, section) [Класс SecurL]:
Common Name (e.g. server FQDN or YOUR name) []:srv1.class.itcloud
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

Проверяем созданный запрос:

```
root@srv1:/etc/ssl# openssl req -noout -text -in srv1.csr
Certificate Request:
    Data:
        Version: 1 (0x0)
        Subject: C = RU, ST =
\C3\90\C2\A1\C3\90\C2\B2\C3\90\C2\B5\C3\91\C2\80\C3\90\C2\B4\C3\90\C2\BB\C3\90\C
2\BE\C3\90\C2\B2\C3\91\C2\81\C3\90\C2\BA\C3\90\C2\B0\C3\91\C2\8F
\C3\90\C2\BE\C3\90\C2\B1\C3\90\C2\BB\C3\90\C2\B0\C3\91\C2\81\C3\91\C2\82\C3\91\C
2\8C, L =
\C3\90\C2\95\C3\90\C2\BA\C3\90\C2\B0\C3\91\C2\82\C3\90\C2\B5\C3\91\C2\80\C3\90\C
2\B8\C3\90\C2\BD\C3\90\C2\B1\C3\91\C2\83\C3\91\C2\80\C3\90\C2\B3, O =
\C3\90\C2\90\C3\90\C2\B9\C3\90\C2\A2\C3\90\C2\B8
\C3\90\C2\9A\C3\90\C2\BB\C3\90\C2\B0\C3\91\C2\83\C3\90\C2\B4, OU =
\C3\90\C2\9A\C3\90\C2\BB\C3\90\C2\B0\C3\91\C2\81\C3\91\C2\81 SecurL, CN =
srv1.class.itcloud
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:ae:35:d5:4d:98:29:66:69:a7:da:09:2b:ea:2f:
                    0e:86:08:6a:56:8c:f6:d6:d9:66:76:92:59:81:fd:
                    29:a0:7b:98:74:69:91:36:33:3f:1b:73:a8:a5:e5:
```

```
                    e9:3b:2a:12:b6:e4:7c:1a:82:14:de:32:1a:c3:d3:
                    8a:f5:4e:92:3e:68:b8:c0:f0:8f:4b:23:4a:95:d9:
                    2e:61:0b:8f:40:94:4c:1f:2e:95:8b:63:e7:fd:1b:
                    49:c6:cf:0d:62:ef:42:0d:65:5e:c4:4d:d5:85:c5:
                    f0:86:6a:12:bc:7c:bd:aa:10:ae:ec:6f:00:64:84:
                    ea:f0:e0:ab:b4:f2:fc:d3:8a:c0:c1:9b:a0:8b:cf:
                    b2:7b:14:c9:6f:b0:ab:22:ae:ab:0b:a3:6d:c0:af:
                    f8:cc:00:e8:66:aa:aa:ed:6d:b5:d7:66:42:08:4c:
                    bd:33:72:b5:fb:51:cf:0c:78:88:bf:3b:a4:90:4e:
                    cf:a6:4b:10:18:e7:51:5f:17:80:13:e9:f8:cd:77:
                    44:78:09:ef:f6:1d:d6:29:1e:d8:a2:df:ea:20:e5:
                    96:57:37:ea:f6:54:3b:01:b6:a1:18:93:a3:81:9b:
                    f4:ca:72:59:5d:5a:69:d8:34:88:4b:14:ab:07:49:
                    2a:b2:dd:97:79:b5:26:ce:5f:6f:59:02:db:f0:99:
                    75:f9
                Exponent: 65537 (0x10001)
        Attributes:
            Requested Extensions:
                X509v3 Basic Constraints:
                    CA:FALSE
                X509v3 Key Usage:
                    Digital Signature, Non Repudiation, Key Encipherment
                X509v3 Subject Alternative Name:
                    IP Address:10.1.1.1, DNS:srv1.class.itcloud,
DNS:www.class.itcloud, DNS:srv1
    Signature Algorithm: sha256WithRSAEncryption
    Signature Value:
        04:56:7a:c2:3e:a8:32:cb:7d:8c:0d:45:88:3a:85:3e:55:85:
        d4:cf:35:81:94:d8:5b:be:4d:26:9e:0b:a6:2e:4c:02:23:e7:
        35:d7:9e:5a:57:25:d6:b1:8d:6d:1f:93:19:65:6c:f2:7c:85:
        52:cf:c4:54:49:77:da:14:09:e0:af:26:6f:18:83:0b:d0:8a:
        be:33:d7:9c:5f:dd:f2:9c:c8:ef:b3:fa:18:b8:14:46:c3:30:
        e1:4b:12:d6:b5:a6:2f:f9:e0:e1:12:60:d5:f9:c1:5b:4b:43:
        bb:17:ff:4b:1b:96:33:9d:13:d8:54:bf:4b:68:a3:73:57:4a:
        64:b9:a3:75:51:e8:61:ec:3c:e2:0e:dc:5b:38:03:09:20:ea:
        5a:d0:fb:8b:75:c4:d8:c6:67:85:09:ff:e2:76:e8:4d:c6:3d:
        47:21:eb:4a:ee:59:56:57:16:5d:c8:90:78:c5:56:11:a4:bc:
        54:77:cf:00:a2:4e:d0:01:9f:1a:99:f0:bc:6c:33:ef:18:95:
        17:b9:6e:58:bd:cd:52:06:1a:81:0c:55:9a:14:df:ac:5c:ea:
        2e:77:75:41:a6:ff:19:db:b5:6a:ae:08:b6:a0:46:65:18:47:
        1c:21:c0:7a:13:e3:84:90:9e:7e:f7:09:a3:7a:07:2d:39:ea:
        e5:ca:4f:5b
```

Копируем запрос в CA:

```
root@srv1:/etc/ssl# cp srv1.csr /srv/pki/
```

Подписываем сертификат:

```
root@srv1:/srv/pki# openssl ca -in srv1.csr -out CA/certs/srv1.crt
-create_serial -days 3653 -config /srv/pki/openssl.cnf
```

3. Создайте сертификаты для cl1 и cl2. Альтернативные имена cl1 и cl2 и IP адреса 10.1.1.11 и 10.1.1.12.
4. Настройте защищенное с помощью stunnel соединение между cl1 и cl2 с использованием ранее полученных сертификатов. Cl1 — сервер.

Установите пакет stunnel4.

Настройка сервера

```
root@cl1:stunnel# cat /etc/stunnel/pppsrv.conf
cert = /etc/ssl/certs/cl1.crt
key = /etc/ssl/private/cl1.key
pid = /tmp/pppstunnel.pid
socket = l:TCP_NODELAY=1
socket = r:TCP_NODELAY=1
debug = 7
output = /var/log/pppstunnel.log
client = no
verify = 2
CAfile = /usr/local/share/ca-certificates/myca/ca.crt

[ppp]
 client = no
 accept = 31234
 exec = /usr/sbin/pppd
 execargs = local noauth 10.2.2.1:10.2.2.2
 pty = yes
```

Перезапустите сервер stunnel4:

```
root@cl1:/etc/ssl# systemctl restart stunnel4.service
```

Настройка клиента.

```
root@cl2:~# cat ~/pppclnt.conf
cert = /etc/ssl/certs/cl2.crt
key = /etc/ssl/private/cl2.key
pid = /tmp/pppstunnel.pid
socket = l:TCP_NODELAY=1
socket = r:TCP_NODELAY=1
debug = 0
output = /var/log/pppstunnel.log
client = yes
verify = 2
CAfile = /usr/local/share/ca-certificates/myca/ca.crt
foreground = no
connect = 10.1.1.11:31234
```

Запустить клиента командой `pppd`:

```
root@cl2:~# pppd passive updetach noauth pty "stunnel pppclnt.conf
```

5. Настройте работу веб сервера apache на srv1 для работы с ранее полученными сертификатами.

Создать цепочку сертификатов:

```
root@srv1:~# mkdir /etc/apache2/ssl.crt
root@srv1:~# cat /etc/ssl/certs/srv1.crt
/usr/local/share/ca-certificates/myca/ca.crt  > /etc/apache2/ssl.crt/srv1-ca.crt
```

Описать сертификат веб сервера в `/etc/apache2/sites-available/default-ssl.conf`

```
root@srv1:~# grep SSLCertificate /etc/apache2/sites-available/default-ssl.conf |
grep -v '#'
      SSLCertificateFile      /etc/ssl/certs/srv1.crt
```

```
      SSLCertificateKeyFile   /etc/ssl/private/srv1.key
      SSLCertificateChainFile /etc/apache2/ssl.crt/srv1-ca.crt
```

Включить модуль ssl:

```
root@srv1:~# a2enmod ssl
```

Включить сайт SSL:

```
root@srv1:~# a2ensite default-ssl.conf
```

Перезапустить веб-сервер и проверить, что он функционирует после перезапуска.

```
root@srv1:~# systemctl restart apache2
root@srv1:~# systemctl status apache2
root@srv1:~# ss -tlnp sport 443
```

Глава 10. Безопасность уровня приложений.

10.1 Контейнеры Docker.

Задание выполняется на cl1.

1. Установите инструменты для работы с контейнерами Docker.

```
# apt install docker.io
```

2. Добавьте пользователя sa в группу docker, чтобы разрешить ему управлять контейнерами.

```
# gpasswd -a sa docker
```

Не забудьте, что членство в группе начинает работать при повторном входе.

3. Найдем образ с веб-сервером (Nginx или Apache).

```
$ docker search --filter is-official=true httpd
```

4. Запустите это приложение и проверьте работу веб-сервера.

```
$ docker run -p 8080:80 --rm -d httpd

$ docker container ls

CONTAINER ID   IMAGE     COMMAND              CREATED          STATUS
PORTS                                  NAMES
3c0097b47d8a   httpd     "httpd-foreground"   49 seconds ago   Up 46 seconds
0.0.0.0:8080->80/tcp, :::8080->80/tcp   epic_bhaskara

$ curl http://localhost:8080
<html><body><h1>It works!</h1></body></html>

$ docker container stop epic_bhaskara
epic_bhaskara
$ docker container ls
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES

$ docker image ls
REPOSITORY   TAG       IMAGE ID       CREATED       SIZE
httpd        latest    4d98e80840bb   2 weeks ago   148MB
```

5. Настройте запуск контейнера так, чтобы он показывал содержимое каталога `~/web8080`. Контейнер должен удаляться после остановки.

```
$ mkdir ~/web8080
$ echo '<html><body><h1>My Web Srv on 8080!</h1></body></html>' >
~/web8080/index.html

$ docker run -p 8080:80 --rm -d -h websrv --name websrv8080-docker -m 1024M -v
 ~/web8080:/usr/local/apache2/htdocs httpd
```

6. Сделайте еще один контейнер, который запускает веб сервер на порту 8888 и работает с каталогом `~/web8888`.

```
$ cp -r ~/web8080 ~/web8888

$ echo '<html><body><h1>My Web Srv on 8888!</h1></body></html>' >
~/web8888/index.html
```

```
$ docker run -p 8888:80 --rm -d -h websrv --name websrv8888-docker -m 1024M -v
 ~/web8888:/usr/local/apache2/htdocs httpd

$ docker container ls
CONTAINER ID   IMAGE     COMMAND              CREATED          STATUS
PORTS                                  NAMES
ce45a6d9b4ca   httpd     "httpd-foreground"   55 seconds ago   Up 53 seconds
0.0.0.0:8888->80/tcp, :::8888->80/tcp   websrv8888-docker
e1a310186864   httpd     "httpd-foreground"   5 minutes ago    Up 5 minutes
0.0.0.0:8080->80/tcp, :::8080->80/tcp   websrv8080-docker
```

10.2 Контейнеры LXD.

Задание выполняется на cl2.

1. Установите пакеты для работы LXD.

```
# apt install lxd lxd-tools
```

2. Произведите первоначальную настройку.

```
# lxd init
Would you like to use LXD clustering? (yes/no) [default=no]:
Do you want to configure a new storage pool? (yes/no) [default=yes]:
Name of the new storage pool [default=default]:
Would you like to connect to a MAAS server? (yes/no) [default=no]:
Would you like to create a new local network bridge? (yes/no) [default=yes]:
What should the new bridge be called? [default=lxdbr0]:
What IPv4 address should be used? (CIDR subnet notation, “auto” or “none”)
[default=auto]:
What IPv6 address should be used? (CIDR subnet notation, “auto” or “none”)
[default=auto]:
Would you like the LXD server to be available over the network? (yes/no)
[default=no]: yes
Address to bind LXD to (not including port) [default=all]:
Port to bind LXD to [default=8443]:
Trust password for new clients:
Again:
Would you like stale cached images to be updated automatically? (yes/no)
[default=yes]:
Would you like a YAML "lxd init" preseed to be printed? (yes/no) [default=no]: y
config:
  core.https_address: '[::]:8443'
  core.trust_password: lin123
networks:
- config:
    ipv4.address: auto
    ipv6.address: auto
  description: ""
  name: lxdbr0
  type: ""
  project: default
storage_pools:
- config: {}
  description: ""
  name: default
  driver: dir
profiles:
- config: {}
  description: ""
  devices:
    eth0:
      name: eth0
      network: lxdbr0
      type: nic
    root:
      path: /
      pool: default
      type: disk
  name: default
projects: []
```

```
cluster: null
```

3. Проверьте источники, с которых вы можете получить образы.

```
# lxc remote list -f csv
images,https://images.linuxcontainers.org,simplestreams,none,YES,NO,NO
local (current),unix://,lxd,file access,NO,YES,NO
ubuntu,https://cloud-images.ubuntu.com/releases,simplestreams,none,YES,YES,NO
ubuntu-daily,https://cloud-images.ubuntu.com/daily,simplestreams,none,YES,YES,NO

# lxc image list images: -f csv
```

Ничего нет. Image server access is being phased out for LXD users, see here for details.

```
# lxc image list ubuntu: -f csv | head -3
a (5 more),2d53824fdf89,yes,ubuntu 17.10 amd64 (release)
(20180706),x86_64,CONTAINER,169.51MB,"Jul 6, 2018 at 12:00am (UTC)"
a (5 more),34bae4293007,yes,ubuntu 17.10 amd64 (release)
(20180706),x86_64,VIRTUAL-MACHINE,307.06MB,"Jul 6, 2018 at 12:00am (UTC)"
a/arm64 (2 more),9807825fcc6a,yes,ubuntu 17.10 arm64 (release)
(20180706),aarch64,VIRTUAL-MACHINE,286.44MB,"Jul 6, 2018 at 12:00am (UTC)"
```

4. Добавьте еще один удаленный репозиторий.

```
# lxc remote add canonical-imgs https://images.lxd.canonical.com --protocol
simplestreams
# lxc image list canonical-imgs: -f csv | wc -l
320
# lxc image list canonical-imgs: -f csv | grep debian | wc -l
24
```

Для сравнения:

```
# lxc image list ubuntu: -f csv | wc -l
11601
# lxc image list ubuntu: -f csv | grep ubuntu | wc -l
11601
```

5. Запустите контейнер с ОС Debian 12 и именем deb12-container.

```
# lxc launch canonical-imgs:'debian/12' deb12-container
Creating deb12-container
Starting deb12-container

# lxc list -c n4st
+-----------------+-----------------------+---------+-----------+
|      NAME       |         IPV4          |  STATE  |   TYPE    |
+-----------------+-----------------------+---------+-----------+
| deb12-container | 10.201.160.126 (eth0) | RUNNING | CONTAINER |
+-----------------+-----------------------+---------+-----------+
```

6. Войдите в контейнер и создайте в нем файл.

```
# lxc exec deb12-container /bin/bash
root@deb12-container:~# echo ABC > test.txt
root@deb12-container:~# exit
exit
```

7. Выключите контейнер и найдите файл, который вы создали внутри контейнера.

```
# lxc stop deb12-container

# lxc list -c n4st
+-----------------+------+---------+-----------+
|      NAME       | IPV4 |  STATE  |   TYPE    |
```

```
+-----------------+------+---------+-----------+
| deb12-container |      | STOPPED | CONTAINER |
+-----------------+------+---------+-----------+

# lxc storage list -f csv
default,dir,/var/lib/lxd/storage-pools/default,,2,CREATED

# cat /var/lib/lxd/storage-pools/default/containers/deb12-container/rootfs/
root/test.txt
ABC

# echo 123 >>
/var/lib/lxd/storage-pools/default/containers/deb12-container/rootfs/root/
test.txt
```

8. Вновь запустите контейнер и проверьте содержимое файла.

```
# lxc start deb12-container
# lxc exec deb12-container cat test.txt
ABC
```

9. Установите на контейнер deb12-container ограничения: 1Гб памяти и 1 ядро.

```
# lxc config set deb12-container limits.memory 1GB
# lxc config set deb12-container limits.cpu 1
```

10. Проверьте конфигурацию контейнера.

```
# lxc config show deb12-container
```

11. Перезагрузите cl2 и проверьте запустится ли контейнер во время старта хостовой ОС.

```
# reboot
# lxc list -f csv -c ns,limits.cpu,limits.memory,boot.autostart
deb12-container,RUNNING,1,1GB,
```

Контейнер запустился, но явного указания на автоматический старт нет.

12. Установите параметр boot.autostart в значение true.

```
# lxc config set deb12-container boot.autostart true
# lxc list -f csv -c ns,limits.cpu,limits.memory,boot.autostart
deb12-container,RUNNING,1,1GB,true
```

Глава 11. Поддержание системы в актуальном состоянии.

11.1 Установка обновлений.

1. Обновите локальный кэш репозиториев. Посмотрите сколько пакетов может быть обновлено.

```
# apt update
Пол:1 http://security.debian.org/debian-security bookworm-security InRelease
[48,0 kB]
Сущ:2 http://deb.debian.org/debian bookworm InRelease
Пол:3 http://deb.debian.org/debian bookworm-updates InRelease [55,4 kB]
Пол:4 http://security.debian.org/debian-security bookworm-security/main amd64
Packages [245 kB]
Получено 348 kB за 2с (231 kB/s)
Чтение списков пакетов… Готово
Построение дерева зависимостей… Готово
Чтение информации о состоянии… Готово
Может быть обновлено 149 пакетов. Запустите «apt list --upgradable» для их
показа.
```

2. Получите список пакетов, которые можно обновить.

```
# apt list --upgradable
```

3. Проверьте имеются ли обновления для ядра.

```
# apt list --upgradable linux*
```

4. Установите обновления без удаления пакетов.

```
# apt upgrade
```

5. Закончите установку обновление с удалением пакетов.

```
# apt full-upgrade
```

6. Если обновлялось ядро, то перезагрузите систему.

Глава 12. Контроль целостности.

12.1 Rkhunter.

1. Установите rkhunter.
2. Проверьте систему на наличие руткитов.

```
# rkhunter --check
```

12.2 Samhain.

Задание выполнять на cl1.

1. Установите samhain.

```
sa@cl1:~$ wget http://la-samhna.de/samhain/samhain-current.tar.gz
sa@cl1:~$ tar xf samhain-current.tar.gz
sa@cl1:~$ tar xf samhain-4.5.2.tar.gz
sa@cl1:~$ cd samhain-4.5.2/
sa@cl1:~/samhain-4.5.2$ ./configure
sa@cl1:~/samhain-4.5.2$ make
sa@cl1:~/samhain-4.5.2$ sudo make install
sa@cl1:~/samhain-4.5.2$ sudo make install-boot
```

2. Создайте baseline.

```
sa@cl1:~$ sudo samhain -t init
```

3. Внесите изменения в файл `passwd` (добавьте нового пользователя) и проверьте систему.

```
sa@cl1:~$ sudo systemctl start samhain.service
sa@cl1:~$ sudo systemctl status samhain.service

sa@cl1:~$ sudo useradd newtestuser

sa@cl1:~$ sudo samhain -t check --foreground 2>/dev/stdout | grep passwd
```

И/или

```
sa@cl1:~$ sudo grep passwd /var/log/samhain_log
```

12.3 AIDE.

Задание выполнять на cl2.

1. Установите aide.
2. Создайте и активируйте базу.

```
# aide -c /etc/aide/aide.conf -i
# mv /var/lib/aide/aide.db.new /var/lib/aide/aide.db
```

3. Внесите изменения в файл `hosts` и проверьте систему.

```
# echo 1.2.3.4 somehost >> /etc/hosts

# aide --check -c /etc/aide/aide.conf
```

Глава 13. Контроль защищенности Linux – систем.

13.1 Scap-workbench.

1. Установите scap-workbench.

```
$ wget https://github.com/OpenSCAP/scap-workbench/releases/download/1.2.1/scap-workbench-1.2.1.tar.bz2

$ sudo apt install build-essential openssh-client libopenscap-dev
libqt5xmlpatterns5-dev ssh-askpass pkg-config asciidoc libpolkit-agent-1-0 cmake

$ tar xf scap-workbench-1.2.1.tar.bz2
$ cd scap-workbench-1.2.1/

$ sed -i 's/-Wall//' CMakeLists.txt
$ sed -i 's/-Werror//' CMakeLists.txt

$ mkdir build
$ cd build/

$ cmake ../

$ make
$ sudo make install
```

2. Установите scap-security-guide .

```
$ sudo apt purge ssg-base

$ wget https://deb.debian.org/debian/pool/main/s/scap-security-guide/ssg-debian_0.1.74-1_all.deb
$ wget https://deb.debian.org/debian/pool/main/s/scap-security-guide/ssg-base_0.1.74-1_all.deb

$ sudo apt install ./ssg-base_0.1.74-1_all.deb ./ssg-debian_0.1.74-1_all.deb
```

3. Проведите сканирование системы.

Перед запуском сканирования отметьте чек бокс «Fetch remote resources».

13.2 oscap.

1. Установите пакеты openscap-scanner, openscap-utils и bzip2.
2. Скачайте документ с описанием рекомендованных настроек системы, в формате OVAL (Open Vulnerability and Assessment Language).

```
# wget https://www.debian.org/security/oval/oval-definitions-bookworm.xml.bz2
# bunzip2 oval-definitions-bookworm.xml.bz2
```

3. Запустите проверку:

```
# oscap oval eval --report /tmp/eval_result.html oval-definitions-bookworm.xml
---
Definition oval:org.debian:def:100025136735225569795784532702130753406: false
Definition oval:org.debian:def:100020194192621893181231895146832483613: false
Evaluation done.
```

4. Проверьте результат сканирования.

```
$ firefox /tmp/eval_result.html
```