

**Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**


Факультет безопасности информационных технологий

Дисциплина:
«Операционные системы»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №8
«Системы контроля доступа. Apparmor и SELinux»

Выполнил:

Рядовой Т.С., студент группы N3252



(подпись)

Проверил:

Чебунин Константин Олегович

(отметка о выполнении)

(подпись)

Санкт-Петербург
2023 г.

СОДЕРЖАНИЕ

Введение.....	4
1 Apparmor	5
1.1 Задание.....	5
1.2 Настройка Apparmor.....	5
2 SELinux	7
2.1 Задание.....	7
2.2 Настройка SELinux.....	7
Заключение.....	10
Список использованных источников	11

ВВЕДЕНИЕ

Цель работы – познакомиться с системами контроля доступа в Linux. Они оба предоставляют механизмы для ограничения прав доступа процессов к ресурсам и файлам в системе с целью повышения безопасности.

В обычном варианте:

- Настроить Apparmor для мониторинга приложения и продемонстрировать его работу при ограниченных правах;
- Настроить SELinux в режиме мандатного доступа и продемонстрировать работу в двухуровневой модели.

1 APPARMOR

1.1 Задание

Настроить Apparmor и показать его работу.

1.2 Настройка Apparmor

Напишем и скомпилируем программу на языке C, работу которой будем в дальнейшем ограничивать. В ней идет попытка связаться с DNS-сервером Google. Так как мы работаем в Ubuntu, то Apparmor изначально предустановлен в систему. Перейдем к созданию и настройке профиля для управления правами файла prog. Командой «`sudo aa-autodep /home/tim/_folder/lab8/prog`» создаем профиль и включаем его в режим обучения командой «`sudo aa-complain /home/tim/_folder/lab8/prog`». Принудительное включение профиля происходит командой «`sudo aa-enforce /home/tim/_folder/lab8/prog`».

Листинг 1 – Программа prog.c

```
#include <stdio.h>
#include <stdlib.h>

int main(){
    int result = system("ping 8.8.8.8 -c 5");
    if (result == 0){
        printf("-----All done!-----\n");
    }
    else{
        printf("-----Error!-----\n");
    }
    return 0;
}
```

Листинг 2 – Пример профиля

```
# Last Modified: Tue Jan  2 17:12:58 2024
abi <abi/3.0>,

include <tunables/global>

/home/tim/_folder/lab8/prog flags=(complain) {
    include <abstractions/base>

    deny /usr/bin/dash x,

    /home/tim/_folder/lab8/prog mr,
}
```

В профиле мы запрещаем выполнение файла в строке «`deny /usr/bin/dash x`». Включаем профиль и проверяем.

```

tim@tim-xubuntu ~/ folder/lab8$ sudo aa-complain /home/tim/_folder/lab8/prog
Setting /home/tim/_folder/lab8/prog to complain mode.
tim@tim-xubuntu ~/ folder/lab8$ ./prog
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=58 time=84.2 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=58 time=84.5 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=58 time=84.2 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=58 time=87.1 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=58 time=85.0 ms

--- 8.8.8.8 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 84.201/85.005/87.114/1.093 ms
-----All done!-----
tim@tim-xubuntu ~/ folder/lab8$ sudo aa-enforce /home/tim/_folder/lab8/prog
Setting /home/tim/_folder/lab8/prog to enforce mode.
tim@tim-xubuntu ~/ folder/lab8$ ./prog
-----Error!-----

```

Рисунок 1 – Запуск программы при разных настройках Apparmor.

Как видим, после включения профиля в программе не запускается команда «system», и на выходе имеем строку ошибки.

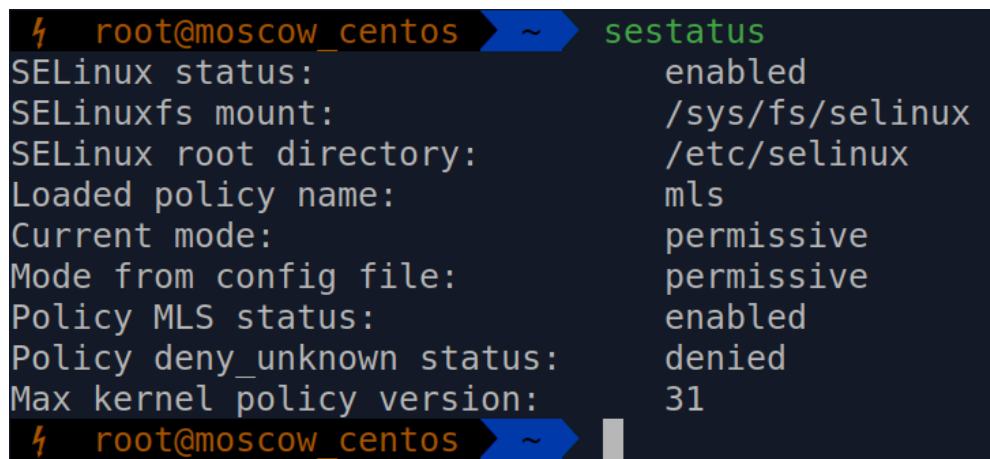
2 SELINUX

2.1 Задание

Настроить SELinux и показать его работу.

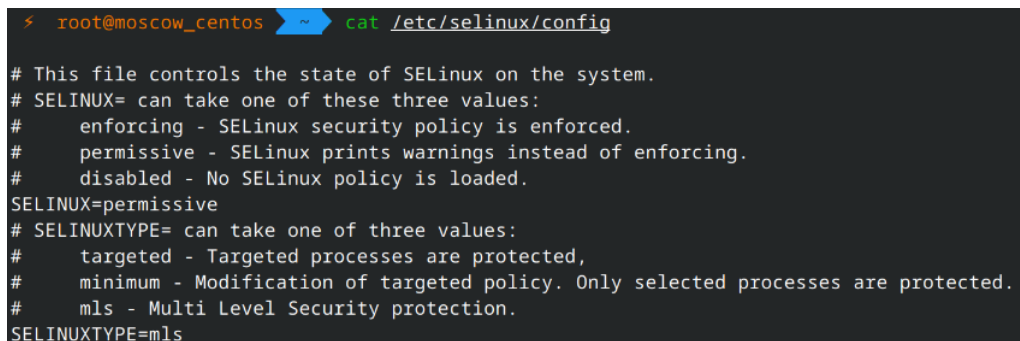
2.2 Настройка SELinux

Будем использовать сервер на CentOS 7. Сначала установим пакет безопасности и поддержки многозадачности (yum install selinux-policy-mls). Далее в файле-конфигурации системы SELinux («/etc/selinux/config») изменяем параметр работы на «permissive», а политику на «mls». Теперь проверим, что система работает в нужном режиме.



```
⚡ root@moscow_centos ~ sestatus
SELinux status:                enabled
SELinuxfs mount:              /sys/fs/selinux
SELinux root directory:       /etc/selinux
Loaded policy name:            mls
Current mode:                  permissive
Mode from config file:         permissive
Policy MLS status:             enabled
Policy deny_unknown status:    denied
Max kernel policy version:     31
⚡ root@moscow_centos ~
```

Рисунок 2 – Проверка SELinux



```
> root@moscow_centos ~ cat /etc/selinux/config
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#   enforcing - SELinux security policy is enforced.
#   permissive - SELinux prints warnings instead of enforcing.
#   disabled - No SELinux policy is loaded.
SELINUX=permissive
# SELINUXTYPE= can take one of three values:
#   targeted - Targeted processes are protected,
#   minimum - Modification of targeted policy. Only selected processes are protected.
#   mls - Multi Level Security protection.
SELINUXTYPE=mls
```

Рисунок 3 – Конфигурационный файл SELinux

Далее создаем двух пользователей.

```
⚡ root@moscow_centos ~ useradd -Z user_u newbie_1
⚡ root@moscow_centos ~ useradd -Z user_u newbie_2
⚡ root@moscow_centos ~ passwd newbie_1
Changing password for user newbie_1.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
⚡ root@moscow_centos ~ passwd newbie_2
Changing password for user newbie_2.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
⚡ root@moscow_centos ~
```

Рисунок 4 – Создание пользователей

У пользователя newbie_1 в корневом разделе создаем файл.

```
⚡ root@moscow_centos ~ su newbie_1
[newbie_1@moscow_centos root]$ cd
[newbie_1@moscow_centos ~]$ vim qwe.py
[newbie_1@moscow_centos ~]$ exit
```

Рисунок 5 – Создание файла у newbie_1

После залогинимся под newbie_2 и попробуем запустить этот файл. Получаем отказ.

```
[newbie_2@moscow_centos ~]$ cd /home/newbie_1
bash: cd: /home/newbie_1: Permission denied
[newbie_2@moscow_centos ~]$
```

Рисунок 6 – Попытка запуска программы

```
[newbie_1@moscow_centos ~]$ ls -l
total 4
-rw-rw-r--. 1 newbie_1 newbie_1 30 Jan  4 15:50 qwe.py
[newbie_1@moscow_centos ~]$ cat qwe.py
print("Hello from newbie_1!")
[newbie_1@moscow_centos ~]$ python qwe.py
Hello from newbie_1!
[newbie_1@moscow_centos ~]$ █
```

Рисунок 7 – Вторая попытка

ЗАКЛЮЧЕНИЕ

В ходе выполнения лабораторной работы мне удалось достигнуть поставленных целей:

Обычного варианта:

- Познакомиться с системами контроля доступа;
- Настроить Apparmor;
- Настроить SELinux.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. <https://losst.pro/nastrojka-apparmor-v-ubuntu->
2. <https://itsecforu.ru/2019/07/25/>
3. <https://losst.pro/nastrojka-selinux>
4. <https://phoenixnap.com/kb/enable-selinux-centos>