

# **Лабораторная работа №6**

**Мандатное разграничение прав в Linux**

Панченко Денис Дмитриевич

# Содержание

1	Цель работы	3
2	Задачи	4
3	Выполнение лабораторной работы	5
4	Вывод	13

# 1 Цель работы

Развить навыки администрирования ОС Linux. Получить первое практическое знакомство с технологией SELinux. Проверить работу SELinux на практике совместно с веб-сервером Apache

## 2 Задачи

- Развить навыки администрирования ОС Linux.
- Получить первое практическое знакомство с технологией SELinux.
- Проверить работу SELinux.

### 3 Выполнение лабораторной работы

- 1) Убедимся, что SELinux работает в режиме enforcing политики targeted (рис. 3.1).

```
[root@derenchik ~]# getenforce
Enforcing
[root@derenchik ~]# sestatus
SELinux status:                enabled
SELinuxfs mount:              /sys/fs/selinux
SELinux root directory:       /etc/selinux
Loaded policy name:            targeted
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
```

Рис. 3.1: Режим

- 2) Обратимся к веб-серверу, запущенному на компьютере и после запустим его (рис. 3.2 - 3.3).

```
[root@derenchik ~]# service httpd status
Redirecting to /bin/systemctl status httpd.service
o httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
   Drop-In: /usr/lib/systemd/system/httpd.service.d
           └─php-fpm.conf
   Active: inactive (dead)
   Docs: man:httpd.service(8)

Apr 24 13:20:22 derenchik systemd[1]: Stopping The Apache HTTP Server...
Apr 24 13:20:23 derenchik systemd[1]: httpd.service: Deactivated successfully.
Apr 24 13:20:23 derenchik systemd[1]: Stopped The Apache HTTP Server.
Apr 24 13:20:23 derenchik systemd[1]: Starting The Apache HTTP Server...
Apr 24 13:20:23 derenchik httpd[46985]: Server configured, listening on: port 81
Apr 24 13:20:23 derenchik systemd[1]: Started The Apache HTTP Server.
Apr 24 13:36:45 derenchik systemd[1]: Stopping The Apache HTTP Server...
Apr 24 13:36:46 derenchik systemd[1]: httpd.service: Deactivated successfully.
Apr 24 13:36:46 derenchik systemd[1]: Stopped The Apache HTTP Server.
Apr 24 13:36:46 derenchik systemd[1]: httpd.service: Consumed 3.254s CPU time.
```

Рис. 3.2: Веб-сервер

```
[root@derenchik ~]# service httpd start
Redirecting to /bin/systemctl start httpd.service
```

Рис. 3.3: Запуск

3) Найдем веб-сервер Apache в списке процессов (рис. 3.4).

```
[root@derenchik ~]# ps auxZ | grep httpd
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 root 48958 0.0  0.0 221368 1860 pts/0 S+ 13:48  0:00
grep --color=auto httpd
```

Рис. 3.4: Список процессов

4) Посмотрим текущее состояние переключателей SELinux для Apache (рис. 3.5).

```
[root@derenchik ~]# sestatus httpd
SELinux status:                enabled
SELinuxfs mount:              /sys/fs/selinux
SELinux root directory:       /etc/selinux
Loaded policy name:            targeted
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
```

Рис. 3.5: Состояние SELinux

5) Посмотрим статистику по политике (рис. 3.6).

```
[root@derenchik ~]# seinfo
Statistics for policy file: /sys/fs/selinux/policy
Policy Version:          33 (MLS enabled)
Target Policy:           selinux
Handle unknown classes:  allow

Classes:                  135      Permissions:              457
Sensitivities:            1        Categories:              1024
Types:                    5135     Attributes:               259
Users:                    8         Roles:                   15
Booleans:                 357      Cond. Expr.:             390
Allow:                    65380    Neverallow:               0
Auditallow:               172      Dontaudit:               8647
Type_trans:               267809   Type_change:              94
Type_member:               37      Range_trans:             6164
Role allow:                39      Role_trans:              419
Constraints:              70      Validatetrans:           0
MLS Constrains:           72      MLS Val. Tran:           0
Permissives:               2       Polcap:                  6
Defaults:                 7        Typebounds:              0
Allowxperm:                0       Neverallowxperm:         0
Auditallowxperm:           0       Dontauditxperm:          0
Ibendportcon:              0       Ibpkeycon:               0
Initial SIDs:              27      Fs_use:                  35
Genfscon:                  109     Portcon:                 665
Netifcon:                  0       Nodecon:                 0
```

Рис. 3.6: Программа

- 6) Определим тип файлов и поддиректорий, находящихся в директории /var/www (рис. 3.7).

```
[root@derenchik ~]# ls -lZ /var/www
total 0
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_script_exec_t:s0 6 Oct 28 12:35 cgi-bin
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_content_t:s0      6 Oct 28 12:35 html
```

Рис. 3.7: Типы файлов

- 7) Определим тип файлов, находящихся в директории /var/www/html (рис. 3.8).

```
[root@derenchik ~]# ls -lZ /var/www/html
total 0
```

Рис. 3.8: Типы файлов

- 8) Создадим html-файл (рис. 3.9 - 3.10).

```
[root@derenchik ~]# nano /var/www/html/test.html
```

Рис. 3.9: Создание

```
GNU nano 5.6.1 /var  
<html>  
<body>test</body>  
</html>
```

Рис. 3.10: html-файл

9) Обратимся к файлу через веб-сервер (рис. 3.11).

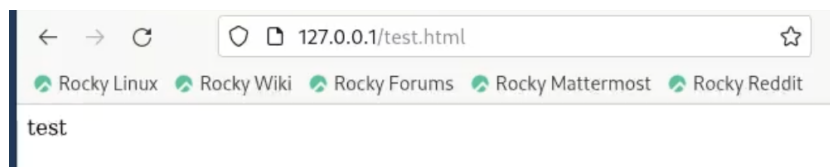


Рис. 3.11: Браузер

10) Проверим контекст файла (рис. 3.12).

```
[root@derenchik ~]# ls -Z /var/www/html/test.html  
unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/test.html
```

Рис. 3.12: Контекст

11) Изменим контекст файла и проверим это (рис. 3.13 - 3.14).

```
[root@derenchik ~]# chcon -t samba_share_t /var/www/html/test.html
```

Рис. 3.13: Контекст файла

```
[root@derenchik ~]# ls -Z /var/www/html/test.html  
unconfined_u:object_r:samba_share_t:s0 /var/www/html/test.html
```

Рис. 3.14: Контекст

12) Попробуем ещё раз получить доступ к файлу через веб-сервер (рис. 3.15).  
Не получилось.



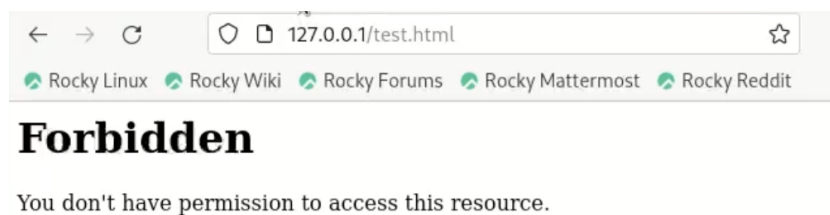


Рис. 3.15: Браузер

13) Проанализируем ситуацию (рис. 3.16 - 3.17).

```
[root@derenchik ~]# ls -l /var/www/html/test.html
-rw-r--r--. 1 root root 33 Apr 24 13:50 /var/www/html/test.html
```

Рис. 3.16: Права доступа

```
[root@derenchik ~]# tail /var/log/messages
Apr 24 13:53:18 derenchik systemd[1]: Starting SETroubleshoot daemo
...
Apr 24 13:53:18 derenchik systemd[1]: Started SETroubleshoot daemon
Apr 24 13:53:18 derenchik setroubleshoot[49319]: failed to retrieve
tml':
Apr 24 13:53:18 derenchik systemd[1]: Started dbus-1.1-org.fedorap
e.
Apr 24 13:53:19 derenchik setroubleshoot[49319]: SELinux is prevent
```

Рис. 3.17: log-файл

14) Попробуем запустить веб-сервер Apache на прослушивание TCP-порта 81 (рис. 3.18 - 3.19).

```
[root@derenchik ~]# nano /etc/httpd/conf/httpd.conf
```

Рис. 3.18: Изменение

```
#Listen 12.34.56.78:80
Listen 81
```

Рис. 3.19: Изменение

15) Выполним перезапуск веб-сервера Apache (рис. 3.20). Произошел сбой.

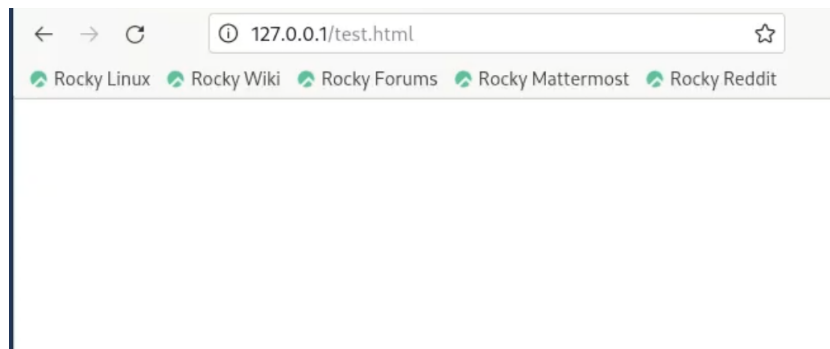


Рис. 3.20: Браузер

16) Проанализируем лог-файлы (рис. 3.21).

```
[root@derenchik ~]# tail -n1 /var/log/messages  
Apr 24 13:55:07 derenchik systemd[1]: Started The Apache HTTP Server.
```

Рис. 3.21: Лог-файлы

17) Выполним команду и проверку (рис. 3.22 - 3.23).

```
[root@derenchik ~]# semanage port -a -t http_port_t -p tcp 81
```

Рис. 3.22: Команда

```
[root@derenchik ~]# semanage port -l | grep http_port_t  
http_port_t          tcp      80, 81, 443, 488, 8008, 8009, 8443, 9000  
pegasus_http_port_t  tcp      5988
```

Рис. 3.23: Проверка

18) Попробуем запустить веб-сервер Apache ещё раз (рис. 3.24). Получилось.

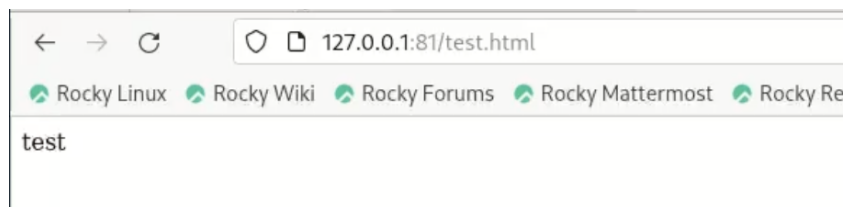


Рис. 3.24: Браузер

- 19) Вернем контекст `httpd_sys_content_t` к файлу и попробуем получить доступ к веб-серверу (рис. 3.25 - 3.26).

```
[root@derenchik ~]# chcon -t httpd_sys_content_t /var/www/html/test.html
```

Рис. 3.25: Контекст

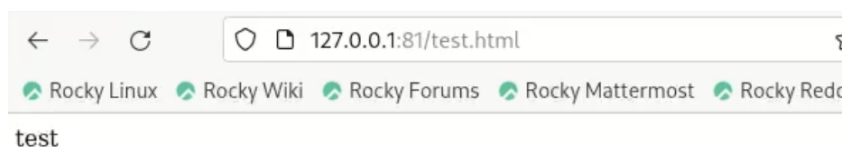


Рис. 3.26: Браузер

- 20) Исправим обратно конфигурационный файл `apache` (рис. 3.27 - 3.28).

```
[root@derenchik ~]# nano /etc/httpd/conf/httpd.conf
```

Рис. 3.27: Контекст

```
# available when the service starts. See the
# page for more information.
#
#Listen 12.34.56.78:80
Listen 80
```

Рис. 3.28: Браузер

- 21) Удалим привязку `http_port_t` к 81 порту (рис. 3.29).

```
[root@derenchik ~]# semanage port -d -t http_port_t -p tcp 81
```

Рис. 3.29: Порт

- 22) Удалим файл `/var/www/html/test.html` (рис. 3.30).

```
[root@derenchik ~]# rm /var/www/html/test.html  
rm: remove regular file '/var/www/html/test.html'? y
```

Рис. 3.30: Удаление

## 4 Вывод

Я развил навыки администрирования ОС Linux, получил первое практическое знакомство с технологией SELinux и проверил работу SELinux на практике совместно с веб-сервером Apache.