

Отчёт по лабораторной работе №6

Дарижапов Тимур Андреевич

11 Октября 2023

РУДН, Москва, Россия

Отчет по лабораторной работе №6

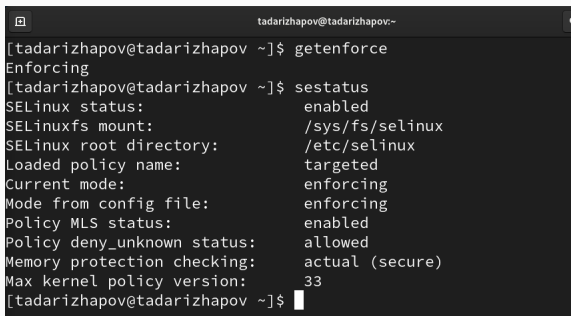
Цель работы: Развить навыки администрирования ОС Linux. Получить первое практическое знакомство с технологией SELinux. Проверить работу SELinux на практике совместно с веб-сервером Apache.

Теоретическое введение

SELinux (Security-Enhanced Linux) обеспечивает усиление защиты путем внесения изменений как на уровне ядра, так и на уровне пространства пользователя, что превращает ее в действительно «непробиваемую» операционную систему. Впервые эта система появилась в четвертой версии CentOS, а в 5 и 6 версии реализация была существенно дополнена и улучшена. SELinux имеет три основных режим работы:

- **Enforcing:** Режим по-умолчанию. При выборе этого режима все действия, которые каким-то образом нарушают текущую политику безопасности, будут блокироваться, а попытка нарушения будет зафиксирована в журнале.
- **Permissive:** В случае использования этого режима, информация о всех действиях, которые нарушают текущую политику безопасности, будут зафиксированы в журнале, но сами действия не будут заблокированы.
- **Disabled:** Полное отключение системы принудительного контроля доступа. Политика SELinux определяет доступ пользователей к ролям, доступ ролей к доменам и доступ доменов к типам. Контекст безопасности — все атрибуты SELinux

Входим в систему под своей учетной записью и убеждаемся, что SELinux работает в режиме enforcing политики targeted с помощью команд “getenforce” и “sestatus”.

A terminal window with a dark background and light text. The title bar shows 'tadarizhapov@tadarizhapov:~'. The user has entered two commands: 'getenforce' and 'sestatus'. The output of 'getenforce' is 'Enforcing'. The output of 'sestatus' is a multi-line status report.

```
tadarizhapov@tadarizhapov ~]$ getenforce
Enforcing
tadarizhapov@tadarizhapov ~]$ 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
tadarizhapov@tadarizhapov ~]$
```

Рис. 1: Рисунок 1

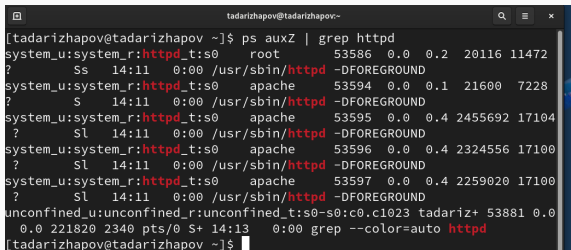
Обращаемся с помощью браузера к веб-серверу, запущенному на моем компьютере, и убеждаемся, что последний работает с помощью команды “service httpd status”.

```
[tadarizhapov@tadarizhapov ~]$ service httpd start
Redirecting to /bin/systemctl start httpd.service
[tadarizhapov@tadarizhapov ~]$ service httpd status
Redirecting to /bin/systemctl status httpd.service
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
   Active: active (running) since Fri 2023-10-13 22:27:53 MSK; 4s ago
     Docs: man:httpd.service(8)
    Main PID: 52999 (httpd)
      Status: "Started, listening on: port 80"
        Tasks: 213 (limit: 24670)
      Memory: 33.3M
         CPU: 50ms
    CGroup: /system.slice/httpd.service
            └─52999 /usr/sbin/httpd -DFOREGROUND
              └─53007 /usr/sbin/httpd -DFOREGROUND
                └─53008 /usr/sbin/httpd -DFOREGROUND
                  └─53009 /usr/sbin/httpd -DFOREGROUND
                    └─53010 /usr/sbin/httpd -DFOREGROUND

окт 13 22:27:53 tadarizhapov.localdomain systemd[1]: Starting The Apache HTTP Ser
окт 13 22:27:53 tadarizhapov.localdomain systemd[1]: Started The Apache HTTP Ser
окт 13 22:27:53 tadarizhapov.localdomain httpd[52999]: Server configured, listen
lines 1-19/19 (END)
```

Рис. 2: Рисунок 2

С помощью команды “ps auxZ | grep httpd” определяем контекст безопасности веб-сервера Apache - httpd_t.



```
tadarizhapov@tadarizhapov:~$ ps auxZ | grep httpd
system_u:system_r:httpd_t:s0 root 53586 0.0 0.2 20116 11472
? Ss 14:11 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 53594 0.0 0.1 21600 7228
? S 14:11 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 53595 0.0 0.4 2455692 17104
? Sl 14:11 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 53596 0.0 0.4 2324556 17100
? Sl 14:11 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 53597 0.0 0.4 2259020 17100
? Sl 14:11 0:00 /usr/sbin/httpd -DFOREGROUND
unconfined_u:unconfined_r:unconfined_t:s0-c0.c1023 tadariz+ 53881 0.0
0.0 221820 2340 pts/0 S+ 14:13 0:00 grep --color=auto httpd
[tadarizhapov@tadarizhapov ~]$
```

Рис. 3: Рисунок 3

С помощью команды “ls -lZ /var/www” посмотрим файлы и поддиректории, находящиеся в директории /var/www. Используя команду “ls -lZ /var/www/html”, определяем, что в данной директории файлов нет. Только владелец или суперпользователь может создавать файлы в директории /var/www/html.

```
[tadarizhapov@tadarizhapov ~]$ ls -lZ /var/www/
итого 0
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_script_exec_t:s0 6 мая
16 23:21 cgi-bin
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_content_t:s0 6 мая
16 23:21 html
[tadarizhapov@tadarizhapov ~]$ ls -lZ /var/www/html
итого 0
[tadarizhapov@tadarizhapov ~]$
```

Рис. 4: Рисунок 4

От имени суперпользователя создаём html-файл
/var/www/html/test.html. Контекст созданного файла -
httpd_sys_content_t.

```
[tadarizhapov@tadarizhapov ~]$ su -  
Пароль:  
[root@tadarizhapov ~]# touch /var/www/html/test.html  
[root@tadarizhapov ~]# nano /var/www/html/test.html  
[root@tadarizhapov ~]# cat /var/www/html/test.html  
<html>  
<body>test</body>  
</html>  
  
[root@tadarizhapov ~]# su - tadarizhapov  
[tadarizhapov@tadarizhapov ~]$ ls -lZ /var/www/html/  
итого 4  
-rw-r--r--. 1 root root unconfined_u:object_r:httpd_sys_content_t:s0 34 ок  
т 14 14:33 test.html  
[tadarizhapov@tadarizhapov ~]$
```

Рис. 5: Рисунок 5

Обращаемся к файлу через веб-сервер, введя в браузере адрес “http://127.0.0.1/test.html”. Файл был успешно отображен.

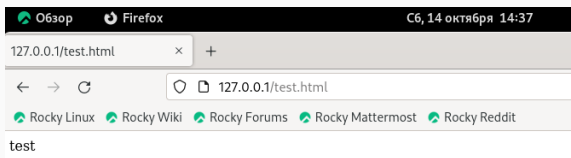


Рис. 6: Рисунок 6

Изучив справку `man httpd_selinux`, выясняем, что для `httpd` определены следующие контексты файлов: `httpd_sys_content_t`, `httpd_sys_script_exec_t`, `httpd_sys_script_ro_t`, `httpd_sys_script_rw_t`, `httpd_sys_script_ra_t`, `httpd_unconfined_script_exec_t`. Контекст моего файла - `httpd_sys_content_t` (в таком случае содержимое должно быть доступно для всех скриптов `httpd` и для самого демона). Изменяем контекст файла на `samba_share_t` командой “`sudo chcon -t samba_share_t /var/www/html/test.html`” и проверяем, что контекст поменялся.

```
[tadarizhapov@tadarizhapov ~]$ man httpd
[tadarizhapov@tadarizhapov ~]$ man selinux
[tadarizhapov@tadarizhapov ~]$ ls -Z /var/www/html/test.html
unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/test.html
[tadarizhapov@tadarizhapov ~]$ chcon -t samba_share_t /var/www/html/test.html
chcon: не удалось изменить контекст безопасности '/var/www/html/test.html'
на «unconfined_u:object_r:samba_share_t:s0»: Операция не позволена
[tadarizhapov@tadarizhapov ~]$ sudo chcon -t samba_share_t /var/www/html/test.html
[sudo] пароль для tadarizhapov:
[tadarizhapov@tadarizhapov ~]$ ls -Z /var/www/html/test.html
unconfined_u:object_r:samba_share_t:s0 /var/www/html/test.html
[tadarizhapov@tadarizhapov ~]$
```

Рис. 7: Рисунок 7

Попробуем еще раз получить доступ к файлу через веб-сервер, введя в браузере адрес “http://127.0.0.1/test.html” и получаем сообщение об ошибке(т.к. к установленному ранее контексту процесс httpd не имеет доступа).

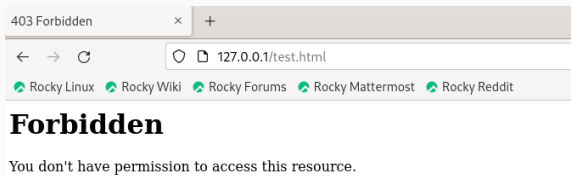
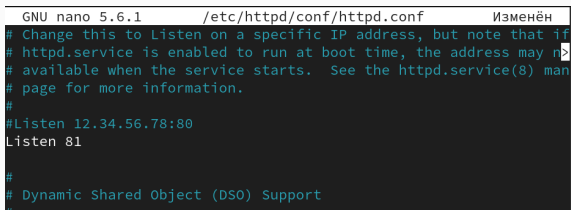


Рис. 8: Рисунок 8

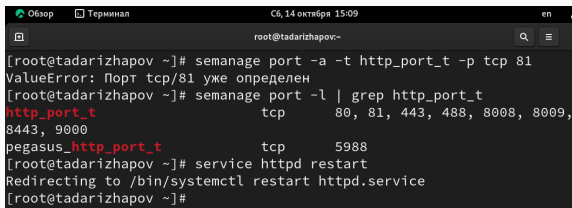
В файле `/etc/httpd/conf/httpd.conf` заменяем строчку “Listen 80” на “Listen 81”, чтобы установить веб-сервер Apache на прослушивание TCP-порта 81.



```
GNU nano 5.6.1 /etc/httpd/conf/httpd.conf Изменён
# Change this to Listen on a specific IP address, but note that if
# httpd.service is enabled to run at boot time, the address may n>
# available when the service starts. See the httpd.service(8) man
# page for more information.
#
#Listen 12.34.56.78:80
Listen 81
#
# Dynamic Shared Object (DSO) Support
#
```

Рис. 9: Рисунок 9

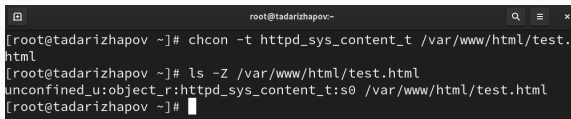
Выполняем команду “semanage port -a -t http_port_t -p tcp 81” и убеждаемся, что порт TCP-81 установлен. Проверяем список портов командой “semanage port -l | grep http_port_t”, убеждаемся, что порт 81 есть в списке и запускаем веб-сервер Apache снова.



```
Обзор Терминал Сб, 14 октября 15:09 en
root@tadarizhapov:~
[root@tadarizhapov ~]# semanage port -a -t http_port_t -p tcp 81
ValueError: Порт tcp/81 уже определен
[root@tadarizhapov ~]# 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
[root@tadarizhapov ~]# service httpd restart
Redirecting to /bin/systemctl restart httpd.service
[root@tadarizhapov ~]#
```

Рис. 10: Рисунок 10

Вернём контекст “httpd_sys_content_t” файлу “/var/www/html/test.html” командой “chcon -t httpd_sys_content_t /var/www/html/test.html” и после этого пробуем получить доступ к файлу через веб-сервер, введя адрес “http://127.0.0.1:81/test.html”, в результате чего увидим содержимое файла - слово “test”.

A terminal window with a dark background and light text. The title bar shows 'root@tadarizhapov:~'. The terminal contains the following commands and output:

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

Рис. 11: Рисунок 11

- В ходе выполнения данной лабораторной работы я развил навыки администрирования ОС Linux, получил первое практическое знакомство с технологией SELinux и проверил работу SELinux на практике совместно с веб-сервером Apache.