

Отчёт по лабораторной работе №6

Аветисян Давид Артурович

11 Октября 2023

РУДН, Москва, Россия

Отчет по лабораторной работе №6

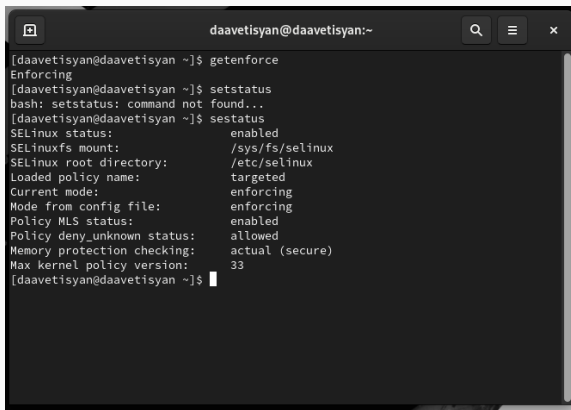
Цель работы: Развить навыки администрирования ОС Linux. Получить первое практическое знакомство с технологией SELinux. Проверить работу SELinux на практике совместно с веб-сервером Apache.

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 titled 'daavetisyan@daavetisyan:~' with search, menu, and close buttons in the title bar. The terminal shows the following commands and output:

```
[daavetisyan@daavetisyan ~]$ getenforce
Enforcing
[daavetisyan@daavetisyan ~]$ setstatus
bash: setstatus: command not found...
[daavetisyan@daavetisyan ~]$ 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
[daavetisyan@daavetisyan ~]$
```

Figure 1: Рисунок 1

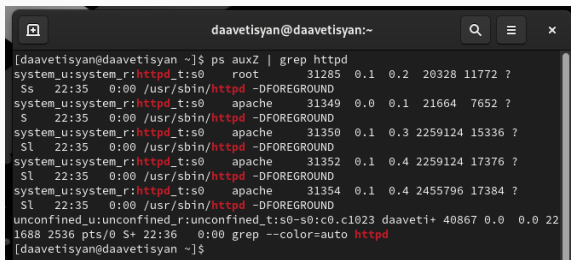
Обращаемся с помощью браузера к веб-серверу, запущенному на моем компьютере, и убеждаемся, что последний работает с помощью команды “service httpd status”.

```
[daavetisyan@daavetisyan ~]$ service httpd start
Redirecting to /bin/systemctl start httpd.service
[daavetisyan@daavetisyan ~]$ 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 Sat 2023-10-14 22:35:08 MSK; 3s ago
     Docs: man:httpd.service(8)
  Main PID: 31285 (httpd)
    Status: "Started, listening on: port 80"
     Tasks: 213 (limit: 24610)
    Memory: 37.6M
       CPU: 128ms
   CGroup: /system.slice/httpd.service
           └─31285 /usr/sbin/httpd -DFOREGROUND
             └─31349 /usr/sbin/httpd -DFOREGROUND
               └─31350 /usr/sbin/httpd -DFOREGROUND
                 └─31352 /usr/sbin/httpd -DFOREGROUND
                   └─31354 /usr/sbin/httpd -DFOREGROUND

окт 14 22:35:08 daavetisyan systemd[1]: Starting The Apache HTTP Server...
окт 14 22:35:08 daavetisyan httpd[31285]: AH00558: httpd: Could not reliably determine the server's fully qualified domain name, because the 'ServerName' directive lacks a 'FullyQualifiedDomainName' directive
окт 14 22:35:08 daavetisyan httpd[31285]: Server configured, listening on: port 80
окт 14 22:35:08 daavetisyan systemd[1]: Started The Apache HTTP Server.
lines 1-20/20 (END)...skipping...
```

Figure 2: Рисунок 2

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



```
daavetisyan@daavetisyan:~$ ps auxZ | grep httpd
system_u:system_r:httpd_t:s0  root      31285  0.1  0.2  20328  11772 ?
Ss  22:35   0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0  apache  31349  0.0  0.1  21664   7652 ?
S   22:35   0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0  apache  31350  0.1  0.3  2259124  15336 ?
Sl  22:35   0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0  apache  31352  0.1  0.4  2259124  17376 ?
Sl  22:35   0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0  apache  31354  0.1  0.4  2455796  17384 ?
Sl  22:35   0:00 /usr/sbin/httpd -DFOREGROUND
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 daaveti+ 40867 0.0  0.0  22
1688 2536 pts/0 S+ 22:36   0:00 grep --color=auto httpd
[daavetisyan@daavetisyan ~]$
```

Figure 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.

```
[daavetisyan@daavetisyan ~]$ sestatus -bigrep httpd
sestatus: invalid option -- 'i'

Usage: sestatus [OPTION]

    -v  Verbose check of process and file contexts.
    -b  Display current state of booleans.

Without options, show SELinux status.
[daavetisyan@daavetisyan ~]$ sestatus -b 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

Policy booleans:
abrt_anon_write                off
abrt_handle_event              off
abrt_upload_watch_anon_write   on
```


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

```
[daavetisyan@daavetisyan ~]$ 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:                    5100     Attributes:               258
Users:                    8         Roles:                    14
Booleans:                 353      Cond. Expr.:             384
Allow:                    65008    Neverallow:               0
Auditallow:               170      Dontaudit:                8572
Type_trans:               265344   Type_change:              87
Type_member:              35       Range_trans:              6164
Role allow:               38       Role_trans:               420
Constraints:              70      Validatetrans:            0
MLS Constrain:            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:                  660
Netifcon:                 0       Nodecon:                   0
```

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

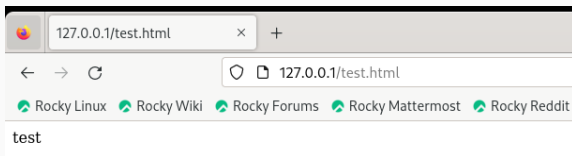


Figure 6: Рисунок 6

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.

```
[daavetisyan@daavetisyan ~]$ man httpd
[daavetisyan@daavetisyan ~]$ man selinux
[daavetisyan@daavetisyan ~]$ ls -Z /var/www/html/test.html
unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/test.html
[daavetisyan@daavetisyan ~]$ 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
[daavetisyan@daavetisyan ~]$ sudo chcon -t samba_share_t /var/www/html/test.html
[sudo] пароль для daavetisyan:
[daavetisyan@daavetisyan ~]$ ls -Z /var/www/html/test.html
unconfined_u:object_r:samba_share_t:s0 /var/www/html/test.html
[daavetisyan@daavetisyan ~]$
```

Figure 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).

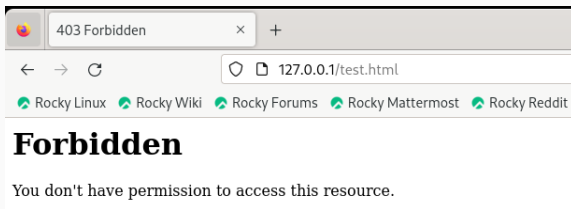
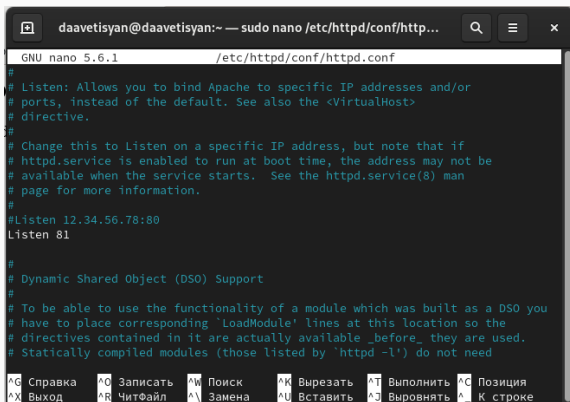


Figure 8: Рисунок 8

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



```
daavetisyan@daavetisyan:~ — sudo nano /etc/httpd/conf/http...
GNU nano 5.6.1 /etc/httpd/conf/httpd.conf
#
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, instead of the default. See also the <VirtualHost>
# directive.
#
# 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 not be
# 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
#
# To be able to use the functionality of a module which was built as a DSO you
# have to place corresponding 'LoadModule' lines at this location so the
# directives contained in it are actually available _before_ they are used.
# Statically compiled modules (those listed by 'httpd -l') do not need
```

^G Справка ^O Записать ^W Поиск ^K Вырезать ^T Выполнить ^C Позиция
^X Выход ^R ЧитФайл ^\ Замена ^U Вставить ^J Выворнять ^_ К строке

Figure 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.

```
[root@daavetisyan ~]# semanage port -a -t http_port_t -p tcp 81
ValueError: Порт tcp/81 уже определен
[root@daavetisyan ~]# 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@daavetisyan ~]# service httpd restart
Redirecting to /bin/systemctl restart httpd.service
[root@daavetisyan ~]#
```

Figure 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”.

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

Figure 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.