

Мандатное разграничение прав в Linux

Соболев М. С.

15 октября 2022

Российский университет дружбы народов, Москва, Россия

Информация

- Соболев Максим Сергеевич
- Студент 4 курса, 1032192035
- Направление: Бизнес-информатика
- Российский университет дружбы народов
- sobolek322lorek@gmail.com

Вводная часть

- Изучим SELinux

- Изучим работу SELinux на практике совместно с веб-сервером Apache.

- Получение практических навыков работы с SELinux
-
- Отчет по ранее выполненной работе

Выполнение лабораторной работы

Входим в систему с полученными учётными данными и убеждаемся, что SELinux работает в режиме enforcing политики targeted с помощью команд `getenforce` и `sestatus`

```
[root@1032192035 1032192035_pfur.ru]# getenforce
Enforcing
[root@1032192035 1032192035_pfur.ru]# 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
```

Figure 1: 1

Проверяем, что apache работает: `systemctl status httpd`

```
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor pr>
   Active: active (running) since Fri 2022-10-14 17:19:48 MSK; 3min 24s ago
```

Figure 2: 2

Найдем веб-сервер Apache в списке процессов, определим его контекст безопасности.

```
[root@1032192035 1032192035_pfur.ru]# ps -eZ | grep httpd
system_u:system_r:httpd_t:s0      38852 ?        00:00:00 httpd
system_u:system_r:httpd_t:s0      38853 ?        00:00:00 httpd
system_u:system_r:httpd_t:s0      38854 ?        00:00:00 httpd
system_u:system_r:httpd_t:s0      38918 ?        00:00:00 httpd
system_u:system_r:httpd_t:s0      38919 ?        00:00:00 httpd
```

Figure 3: 3

Посмотрим текущее состояние переключателей SELinux для Apache с помощью команды `sestatus -bigrep httpd`

Обратим внимание, что многие из них находятся в положении «off».

```
[root@1032192035 1032192035_pfur.ru]# sestatus -b | grep httpd
httpd_anon_write off
httpd_builtin_scripting on
httpd_can_check_spam off
httpd_can_connect_ftp off
httpd_can_connect_ldap off
httpd_can_connect_mythtv off
httpd_can_connect_zabbix off
httpd_can_manage_courier_spool off
httpd_can_network_connect off
httpd_can_network_connect_cobbler off
httpd_can_network_connect_db off
httpd_can_network_memcache off
httpd_can_network_relay off
httpd_can_sendmail off
httpd_dbus_avahi off
httpd_dbus_sssd off
httpd_dontaudit_search_dirs off
```

Посмотрим статистику по политике с помощью команды `seinfo`, также определите множество пользователей, ролей, типов.

```
[root@1032192035 1032192035_pfur.ru]# seinfo
Statistics for policy file: /sys/fs/selinux/policy
Policy Version:          33 (MLS enabled)
Target Policy:           selinux
Handle unknown classes:  allow

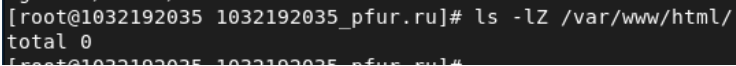
Classes:                  133      Permissions:              454
Sensitivities:            1        Categories:              1024
Types:                    4971     Attributes:               251
Users:                    8         Roles:                    14
Booleans:                 343       Cond. Expr.:             376
Allow:                    62517     Neverallow:               0
Auditallow:               163       Dontaudit:               8293
Type_trans:              247066    Type_change:              87
Type_member:              35        Range_trans:             5957
Role_allow:               37        Role_trans:               418
Constraints:              72       Validatetrans:            0
MLS Constrains:          72        MLS Val. Tran:            0
Permissives:              0         Polcap:                   5
Defaults:                 7         Typebounds:               0
Allowxperm:               0         Neverallowxperm:          0
Auditallowxperm:          0        Dontauditxperm:           0
```

Определим тип файлов и поддиректорий, находящихся в директории `/var/www/`, с помощью команды `ls -lZ /var/www/`

```
[root@1032192035 1032192035_pfur.ru]# ls -lZ /var/www/  
total 8  
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_script_exec_t:s0 4096 May 16 15:10 cgi-bin  
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_content_t:s0 4096 May 16 15:10 html
```

Figure 6: 6

Определим тип файлов, находящихся в директории /var/www/html/: `ls -lZ /var/www/html/`



```
[root@1032192035 1032192035_pfur.ru]# ls -lZ /var/www/html/  
total 0  
[root@1032192035 1032192035_pfur.ru]#
```

Figure 7: 7

Файлов нет

Определим круг пользователей, которым разрешено создание файлов в директории `/var/www/html/`.

```
[root@1032192035 1032192035_pfur.ru]# ls -la /var/www/  
total 16  
drwxr-xr-x.  4 root root 4096 Oct 14 17:17 .  
drwxr-xr-x. 21 root root 4096 Oct 14 17:17 ..  
drwxr-xr-x.  2 root root 4096 May 16 15:10 cgi-bin  
drwxr-xr-x.  2 root root 4096 May 16 15:10 html
```

Figure 8: 8

Создание файлов разрешено только пользователю root

Создадим от имени суперпользователя html-файл `/var/www/html/test.html` следующего содержания:

test



```
root@1032192035:/home/1032192035_pfur.ru
GNU nano 5.6.1 /var/www/html/test.html
<html>
  <body>
    test
  </body>
</html>
```

Figure 9: 9

Проверим контекст созданного нами файла.

```
[root@1032192035 1032192035_pfur.ru]# ls -lZ /var/www/html/test.html  
-rw-r--r--. 1 root root unconfined_u:object_r:httpd_sys_content_t:s0 39 Oct 14 17:27 /var/www/html/test.html
```

Figure 10: 10

Обращаемся к файлу через веб-сервер, введя в браузере адрес `http://localhost/test.html`. Убеждаемся, что файл был успешно отображён.

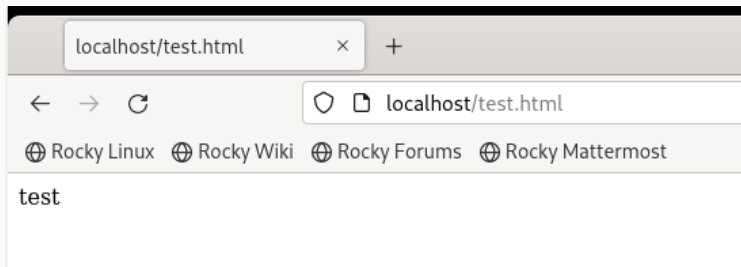


Figure 11: 11

Изучим справку `man httpd_selinux` и выясним, какие контексты файлов определены для `httpd`.

Изучили.

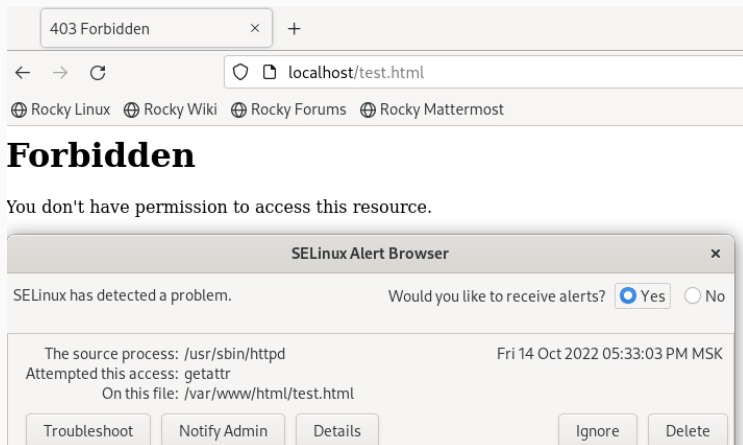
Изменяем контекст файла `/var/www/html/test.html` с `httpd_sys_content_t` на `samba_share_t`, к которому процесс `httpd` не должен иметь доступа: `chcon -t samba_share_t /var/www/html/test.html` `ls -Z /var/www/html/test.html`

После этого проверяем, что контекст поменялся.

```
[root@1032192035 1032192035_pfur.ru]# chcon -t samba_share_t /var/www/html/test.html
[root@1032192035 1032192035_pfur.ru]# ls -lZ /var/www/html/test.html
-rw-r--r--. 1 root root unconfined_u:object_r:samba_share_t:s0 39 Oct 14 17:27 /var/www/html/test.html
[root@1032192035 1032192035_pfur.ru]#
```

Figure 12: 13

Попробуем ещё раз получить доступ к файлу через веб-сервер, введя в браузере адрес `http://localhost/test.html`. Получаем сообщение об ошибке: Forbidden You don't have permission to access /test.html on this server.



Файл не отображён, поскольку процесс httpd не имеет доступа к файлам с заданным процессом. SELinux не выдаёт мандат на чтение, таким образом запрещая чтение файла

```
type=AVC msg=audit(1665757983.560:204): avc: denied { getattr } for pid=38854 comm="httpd" path="/var/www/html/test.html" dev="sda2" ino=1048868 scontext=system_u:system_r:httpd_t:s0 tcontext=unconfined_u:object_r:samba_share_t:s0 tclass=file permissive=0
type=SYSCALL msg=audit(1665757983.560:204): arch=c000003e syscall=262 success=no exit=-13 a0=ffffff9c a1=7fe07800eaa0 a2=7fe07e7fb830 a3=0 items=0 ppid=38852 pid=38854 auid=4294967295 uid=48 gid=48 euid=48 suid=48 fsuid=48 egid=48 sgid=48 fsgid=48 tty=(none) ses=4294967295 comm="httpd" exe="/usr/sbin/httpd" subj=system_u:system_r:httpd_t:s0 key=(null)ARCH=x86_64 SYSCALL=newfstatat AUID="unset" UID="apache" GID="apache" EUID="apache" SUID="apache" FSUID="apache" EGID="apache" SGID="apache" FSGID="apache"
type=PROCTITLE msg=audit(1665757983.560:204): proctitle=2F7573722F7362696E2F6874747064002044464F524547524F554E44
type=AVC msg=audit(1665757983.560:205): avc: denied { getattr } for pid=38854 comm="httpd" path="/var/www/html/test.html" dev="sda2" ino=1048868 scontext=system_u:system_r:httpd_t:s0 tcontext=unconfined_u:object_r:samba_share_t:s0 tclass=file permissive=0
type=SYSCALL msg=audit(1665757983.560:205): arch=c000003e syscall=262 success=no exit=-13 a0=ffffff9c a1=7fe07800eb80 a2=7fe07e7fb830 a3=100 items=0 ppid=38852 pid=38854 auid=4294967295 uid=48 gid=48 euid=48 suid=48 fsuid=48 egid=48 sgid=48 fsgid=48 tty=(none) ses=4294967295 comm="httpd" exe="/usr/sbin/httpd" subj=system_u:system_r:httpd_t:s0 key=(null)ARCH=x86_64 SYSCALL=newfstatat AUID="unset" UID="apache" GID="apache" EUID="apache" SUID="apache" FSUID="apache" EGID="apache" SGID="apache" FSGID="apache"
type=PROCTITLE msg=audit(1665757983.560:205): proctitle=2F7573722F7362696E2F6874747064002044464F524547524F554E44
type=SERVICE_START msg=audit(1665757983.565:206): pid=1 uid=0 auid=4294967295 ses=4294967295 subj=system_u:system_r:init_t:s0 msg='unit=dbus-1.10-org.fedoraproject.Setroubleshootd@1 comm="systemd" exe="/usr/lib/systemd/systemd" hostname=? addr=? terminal=? res=success'UID="root" AUID="unset"
```

Figure 14: 15

Попробуем запустить веб-сервер Apache на прослушивание TCP-порта 81.

A screenshot of a text editor showing two lines of configuration code. The first line is commented out with a blue hash symbol and the text "#Listen 12.34.56.7". The second line is active and reads "Listen 81" in white text on a black background.

```
#Listen 12.34.56.7  
Listen 81
```

Figure 15: 16

Выполните перезапуск веб-сервера Apache. Произошёл сбой? Почему? Нет, не произошёл. Новые версии политик selinux позволяют httpd работать на разных

```
[root@1032192035 ~]# apachectl restart
[root@1032192035 ~]# systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)
   Active: active (running) since Fri 2022-10-14 17:39:11 MSK; 20s ago
     Docs: man:httpd.service(8)
  Main PID: 41268 (httpd)
    Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec:  0 B/sec"
    Tasks: 213 (limit: 61000)
   Memory: 39.1M
      CPU: 71ms
   CGroup: /system.slice/httpd.service
           └─41268 /usr/sbin/httpd -DFOREGROUND
             └─41269 /usr/sbin/httpd -DFOREGROUND
               └─41270 /usr/sbin/httpd -DFOREGROUND
                 └─41271 /usr/sbin/httpd -DFOREGROUND
                   └─41272 /usr/sbin/httpd -DFOREGROUND

Oct 14 17:39:11 1032192035.local systemd[1]: Starting The Apache HTTP Server...
Oct 14 17:39:11 1032192035.local httpd[41268]: Server configured, listening on: port 81
Oct 14 17:39:11 1032192035.local systemd[1]: Started The Apache HTTP Server.
```

портах, в т.ч. 81.

Поменяем в конфиге порт на тот, который действительно не находится в списке разрешённых (8874), попробуем перезапустить `httpd`, получим ошибку, изучим логи.

Выполним команду `semanage port -a -t http_port_t -p tcp 8874` После этого проверим список портов командой `semanage port -l | grep http_port_t` Убеждаемся, что порт 8874 появился в списке.

```
[root@1032192035 ~]# semanage port -a -t http_port_t -p tcp 8874
[root@1032192035 ~]# semanage port -l | grep http_port_t
http_port_t      tcp      8874, 80, 81, 443, 488, 8008, 8009, 8443, 9000
```

Figure 16: 19

Да, поняли. Политика selinux не позволяла процессу прослушивать порт

```
[root@1032192035 ~]# apachectl restart
[root@1032192035 ~]# systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)
   Active: active (running) since Fri 2022-10-14 18:24:21 MSK; 4s ago
     Docs: man:httpd.service(8)
  Main PID: 42322 (httpd)
    Status: "Started, listening on: port 8874"
    Tasks: 213 (limit: 61000)
   Memory: 49.1M
      CPU: 63ms
   CGroup: /system.slice/httpd.service
           └─42322 /usr/sbin/httpd -DFOREGROUND
             └─42323 /usr/sbin/httpd -DFOREGROUND
               └─42324 /usr/sbin/httpd -DFOREGROUND
                 └─42325 /usr/sbin/httpd -DFOREGROUND
                   └─42326 /usr/sbin/httpd -DFOREGROUND

Oct 14 18:24:21 1032192035.local systemd[1]: Starting The Apache HTTP Server...
Oct 14 18:24:21 1032192035.local httpd[42322]: Server configured, listening on: port 8874
Oct 14 18:24:21 1032192035.local systemd[1]: Started The Apache HTTP Server.
```

Figure 17: 20

Вернем контекст `httpd_sys_content_t` к файлу `/var/www/html/test.html`: `chcon -t httpd_sys_content_t /var/www/html/test.html` После этого попробуем получить доступ к файлу через веб-сервер, введя в браузере адрес `http://localhost:8874/test.html`. Мы должны увидеть содержимое файла — слово «test».

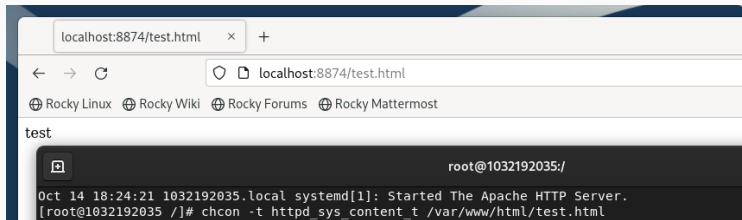
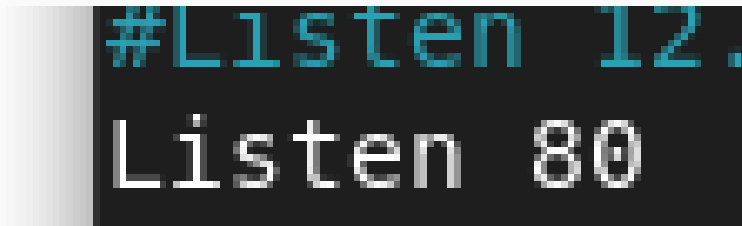


Figure 18: 21

Исправим обратно конфигурационный файл apache, вернув Listen 80



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

Figure 19: 22

Удалим привязку http_port_t к 8874 порту: semanage port -d -t http_port_t -p tcp 8874 и проверим, что порт 8874 удалён

```
[root@1032192035 ~]# semanage port -d -t http_port_t -p tcp 8874
```

Figure 20: 23

Удалим файл /var/www/html/test.html: `rm /var/www/html/test.html`

```
[root@1032192035 ~]# rm -rf /var/www/html/test.html
```

Figure 21: 24

Выводы

Мы развили навыки администрирования ОС Linux. Получили первое практическое знакомство с технологией SELinux

Проверили работу SELinux на практике совместно с веб-сервером Apache

...