

Лабораторная работа № 6

Мандатное разграничение прав в Linux

Артамонов Т. Е.

12 октября 2024

Российский университет дружбы народов, Москва, Россия

Информация

- Артамонов Тимофей Евгеньевич
- студент группы НКНбд-01-21
- Российский университет дружбы народов
- <https://github.com/teartamonov>



Развить навыки администрирования ОС Linux. Получить первое практическое знакомство с технологией SELinux¹. Проверить работу SELinux на практике совместно с веб-сервером Apache.

В SELinux права доступа определяются самой системой при помощи специально определённых политик. Политики работают на уровне системных вызовов и применяются самим ядром (но можно реализовать и на уровне приложения). SELinux действует после классической модели безопасности Linux: через SELinux нельзя разрешить то, что запрещено через права доступа пользователей или групп. Политики описываются при помощи специального гибкого языка описания правил доступа. В большинстве случаев правила SELinux «прозрачны» для приложений, и не требуется никакой их модификации. В состав некоторых дистрибутивов входят готовые политики, в которых права могут определяться на основе совпадения типов процесса (субъекта) и файла (объекта) — это основной механизм SELinux. Две других формы контроля доступа — доступ на основе ролей и на основе многоуровневой системы безопасности.

Выполнение лабораторной работы

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

```
[teartamonov@teartamonov conf]$ getenforce
Enforcing
[teartamonov@teartamonov conf]$ 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
[teartamonov@teartamonov conf]$
```

Figure 1: Все верно

Обратимся к веб-серверу и убедимся, что он работает. (рис. 2)

```
[teartamonov@teartamonov conf]$ service httpd status
Redirecting to /bin/systemctl status httpd.service
• httpd.service - The Apache HTTP Server
  Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
  Active: active (running) since Sun 2024-09-01 12:16:01 MSK; 16min ago
  Docs: man:httpd.service(8)
  Main PID: 41566 (httpd)
  Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec: 0;Uptime: 0s"
  Tasks: 177 (limit: 12190)
  Memory: 38.4M
  CPU: 4.044s
  CGroup: /system.slice/httpd.service
          └─41566 /usr/sbin/httpd -DFOREGROUND
             └─41567 /usr/sbin/httpd -DFOREGROUND
                └─41568 /usr/sbin/httpd -DFOREGROUND
                   └─41569 /usr/sbin/httpd -DFOREGROUND
                      └─41570 /usr/sbin/httpd -DFOREGROUND

сен 08 12:15:41 teartamonov systemd[1]: Starting The Apache HTTP Server...
сен 08 12:15:51 teartamonov httpd[41566]: AH00558: httpd: Could not reliably determine the server's fully qualified domain name, setting 'ServerName' to localhost.
сен 08 12:16:01 teartamonov httpd[41566]: Server configured, listening on: port 80
сен 08 12:16:01 teartamonov systemd[1]: Started The Apache HTTP Server.
lines 1-20/20 (END)
```

Figure 2: Веб-сервер работает

Найдем apache в списке процессов. (рис. 3)

```
[teartamonov@teartamonov conf]$ ps auxZ | grep httpd
system_u:system_r:httpd_t:s0 root 41566 0.0 0.5 20364 11528 ? Ss 12:15 0:00 /usr/sbin/
httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 41567 0.0 0.3 22096 7504 ? S 12:16 0:00 /usr/sbin/
httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 41568 0.1 0.7 2161168 15096 ? Sl 12:16 0:01 /usr/sbin/
httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 41569 0.1 0.6 2161168 13108 ? Sl 12:16 0:01 /usr/sbin/
httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 41570 0.1 1.1 2423376 23772 ? Sl 12:16 0:01 /usr/sbin/
httpd -DFOREGROUND
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 teartam+ 42638 0.0 0.1 221688 2432 pts/0 R+ 12:34 0:
00 grep --color=auto httpd
[teartamonov@teartamonov conf]$
```

Figure 3: Нашли с помощью команды `ps auxZ | grep httpd`

Посмотрим статистику по политике. (рис. 4)

```
[teartamonov@teartamonov conf]$ 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:                    5145     Attributes:               259
Users:                    8         Roles:                    15
Booleans:                 356      Cond. Expr.:             388
Allow:                    65504     Neverallow:               0
Auditallow:               176      Dontaudit:                8682
Type_trans:               271770   Type_change:              94
Type_member:              37        Range_trans:              5931
Role allow:               40        Role_trans:               417
Constraints:              70        Validatetrans:            0
MLS Constrain:            72        MLS Val. Tran:            0
Permissives:              4         Polcap:                   6
Defaults:                 7         Typebounds:               0
Allowxperm:               0         Neverallowxperm:          0
```

Figure 4: Видно количество разных сущностей, например, пользователей

Определим тип файлов и директорий в /var/www. (рис. 5)

```
[teartamonov@teartamonov conf]$ ls -lZ /var/www
итого 0
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_script_exec_t:s0 6 апр  8 19:30 cgi-bin
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_content_t:s0      6 апр  8 19:30 html
[teartamonov@teartamonov conf]$
```

Figure 5: httpd_sys_script_exec_t и httpd_sys_content_t

Посмотрим список пользователей. (рис. 6)

```
[teartamonov@teartamonov conf]$ seinfo -u  
  
Users: 8  
  guest_u  
  root  
  staff_u  
  sysadm_u  
  system_u  
  unconfined_u  
  user_u  
  xguest_u  
[teartamonov@teartamonov conf]$ seinfo -t
```

Figure 6: Список пользователей

Создадим от имени root файл в /var/www/html. (рис. 7)

```
[teartamonov@teartamonov ~]$ su -  
Пароль:  
[root@teartamonov ~]# touch /var/www/html/test.html  
[root@teartamonov ~]# mcedit /var/www/html/test.html  
bash: mcedit: команда не найдена...  
Установить пакет «mc», предоставляющий команду «mcedit»? [N/y] n  
  
[root@teartamonov ~]# emacs /var/www/html/test.html  
[root@teartamonov ~]# secon --file /var/www/html/test.html  
user: unconfined_u  
role: object_r  
type: httpd_sys_content_t  
sensitivity: s0  
clearance: s0  
mls-range: s0  
[root@teartamonov ~]# █
```

Figure 7: Успешно

Попробуем посмотреть файл в браузере. (рис. 8)

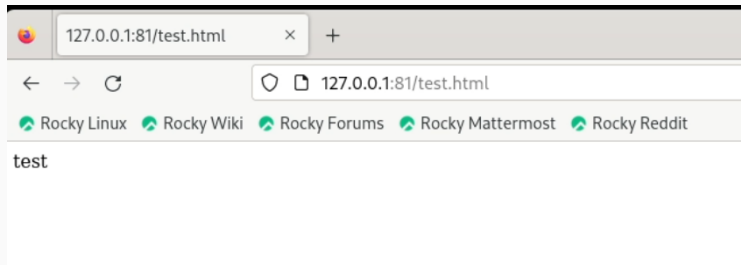


Figure 8: Успешно

Изменим контекст файла. (рис. 9)

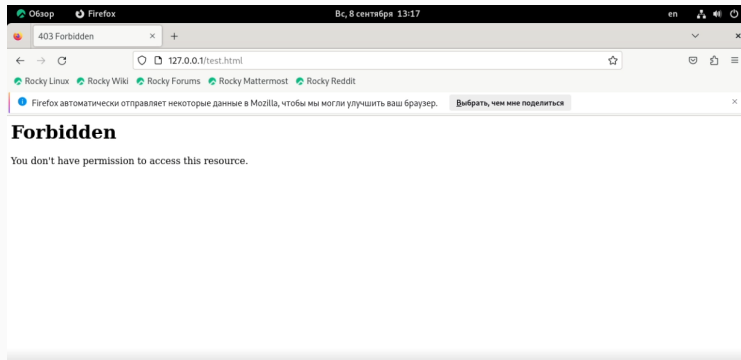


Figure 9: Успешно, теперь файл не отображается

Добавим на прослушивание порт 81. (рис. 10)

```
port 81 already defined, modifying instead
[teartamonov@teartamonov conf]$ sudo semanage port -l | grep http_port_t
http_port_t          tcp      1, 80, 81, 443, 488, 8008, 8009, 8443, 9000
pegasus_http_port_t  tcp      5988
[teartamonov@teartamonov conf]$ sudo systemctl restart httpd
```

Figure 10: Успешно, теперь происходит сбой, если не добовать 81 порт в список

Вернем все как было. (рис. 11)

```
valueError: Port tcp/81 is defined in policy, cannot be deleted
[teartamonov@teartamonov conf]$ sudo 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
[teartamonov@teartamonov conf]$
```

Figure 11: Успешно

Развили навыки администрирования ОС Linux. Получили первое практическое знакомство с технологией SELinux¹. Проверили работу SELinx на практике совместно с веб-сервером Apache.

1. SELinux [Электронный ресурс]. Wikimedia Foundation, Inc., 2024. URL: <https://ru.wikipedia.org/wiki/SELinux>.