

# Лабораторная работа №6

Мандатное разграничение прав в Linux

---

Буллет Т. А.

15 февраля 2025

Российский университет дружбы народов, Москва, Россия

## Информация

---

- Буллер Татьяна Александровна
- студент направления Бизнес-информатика
- Российский университет дружбы народов

## Вводная часть

---

- Операционная система linux, дистрибутив Kali
- Среда виртуализации VirtualBox
- Технология мандатного разграничения прав SELinux

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

- Процессор **pandoc** для входного формата Markdown
- Среда виртуализации VirtualBox
- SELinux
- Apache2

## Выполнение лабораторной работы

---



## Выполнение лабораторной работы

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

```
(tabuller@jordi)-[~]  
$ sestatus  
SELinux status:                enabled  
SELinuxfs mount:              /sys/fs/selinux  
SELinux root directory:       /etc/selinux  
Loaded policy name:            default  
Current mode:                  permissive  
Mode from config file:         permissive  
Policy MLS status:             enabled  
Policy deny_unknown status:    allowed  
Memory protection checking:    actual (secure)  
Max kernel policy version:     33
```

Рис. 1: sestatus

Предварительно запустив сервис `apache2`, проверим, работает ли `httpd`.

```
(tabuller@jordi)-[~]  
$ ps aux | grep httpd  
tabuller  9480  0.0  0.0  6332  2200 pts/2    S+   08:40   0:00 grep --color=auto httpd
```

Рис. 2: httpd

Далее с помощью команды 'ps auxZ | grep httpd' определим контекст безопасности веб-сервера Apache. Строка unconfined говорит о том, что никакие дополнительные ограничения не установлены.

```
(tabuller@ Jordi)~$ ps auxZ | grep httpd
system_u:system_r:httpd_t:s0 root 4300 0.0 0.4 207860 22816 ? Ss 08:23 0:00 /usr/s
bin/apache2 -k start
system_u:system_r:httpd_t:s0 www-data 4302 0.0 0.2 208564 13556 ? S 08:23 0:00 /usr/s
bin/apache2 -k start
system_u:system_r:httpd_t:s0 www-data 4303 0.0 0.2 208564 13556 ? S 08:23 0:00 /usr/s
bin/apache2 -k start
system_u:system_r:httpd_t:s0 www-data 4304 0.0 0.2 208596 11936 ? S 08:23 0:00 /usr/s
bin/apache2 -k start
system_u:system_r:httpd_t:s0 www-data 4305 0.0 0.2 208596 11936 ? S 08:23 0:00 /usr/s
bin/apache2 -k start
system_u:system_r:httpd_t:s0 www-data 4306 0.0 0.2 208596 11936 ? S 08:23 0:00 /usr/s
bin/apache2 -k start
system_u:system_r:httpd_t:s0 www-data 5601 0.0 0.2 208596 11936 ? S 08:28 0:00 /usr/s
bin/apache2 -k start
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0-c1023 tabuller 9585 0.0 0.0 6332 2188 pts/2 S+ 08:40 0
:00 grep --color=auto httpd
```

Рис. 3: Контекст безопасности веб-сервера Apache

## Выполнение лабораторной работы

Посмотрим текущее состояние переключателей SELinux для Apache с помощью команды `sestatus -b | grep httpd`. Видим настройки по умолчанию: все переключатели разрешений для httpd в режиме off.

```
(tabuller@jordi)-[~]  
$ sestatus -b | grep httpd  
allow_httpd_anon_write off  
allow_httpd_apcupsd CGI_script_anon_write off  
allow_httpd_awstats_script_anon_write off  
allow_httpd_collectd_script_anon_write off  
allow_httpd_cvs_script_anon_write off  
allow_httpd_lightsquid_script_anon_write off  
allow_httpd_man2html_script_anon_write off  
allow_httpd_mediawiki_script_anon_write off  
allow_httpd_mod_auth_pam off  
allow_httpd_mojomojo_script_anon_write off  
allow_httpd_munin_script_anon_write off  
allow_httpd_nagios_script_anon_write off  
allow_httpd_nutups CGI_script_anon_write off  
allow_httpd_prewikka_script_anon_write off  
allow_httpd_smokeping CGI_script_anon_write off  
allow_httpd_squid_script_anon_write off
```

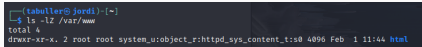
Рис. 4: sestatus

Команда `seinfo` позволяет посмотреть статистику по политике. Видим 3983 типа, 9 пользователей и 15 ролей.

```
(tabuller@jordi)-[~]  
$ seinfo  
Statistics for policy file: /sys/fs/selinux/policy  
Policy Version:          33 (MLS enabled)  
Target Policy:           selinux  
Handle unknown classes:  allow  
Classes:                 135      Permissions:          431  
Sensitivities:           1        Categories:          1024  
Types:                   3983     Attributes:           231  
Users:                   9        Roles:                15  
Booleans:               322      Cond. Expr.:         351  
Allow:                   98715    Neverallow:           0  
Auditallow:             21       Dontaudit:           14679  
Type_trans:             9388     Type_change:          72  
Type_member:            16       Range_trans:          69  
Role allow:             32       Role_trans:           362  
Constraints:            133     Validatetrans:         0  
MLS Constrain:          110     MLS Val. Tran:         0  
Permissives:            0       Polcap:               5
```

Рис. 5: `seinfo`

Командой `ls -lZ` определим типы файлов, находящихся в директории `/var/www` и то же самое для `/var/www/html`. Видим, что для всех файлов и поддиректорий установлены права, позволяющие только владельцу (суперпользователю) осуществлять запись в них, потому дальнейшую работу необходимо будет вести через `sudo`.



```
(tabuller@ jordi)~  
$ ls -lZ /var/www  
total 4  
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_content_t:s0 4096 Feb  1 11:44 html
```

Рис. 6: Типы файлов, находящихся в директории `/var/www`

От имени суперпользователя пишем в директории `/var/www/html` короткий файл `test.html`, при компиляции которого на экран будет выведено только слово `test`. Этот файл должен быть доступен по адресу `127.0.0.1/test.html`, так как на машине запущен сервис `apache`. Перейдя по адресу и не указывая имя файла мы встретим приветственную страницу `apache` (*it works!*), далее, уточнив имя файла `test.html`, увидим вывод созданного нами файла.

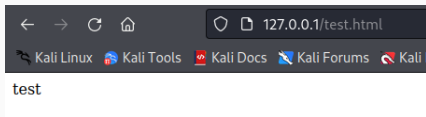
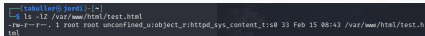


Рис. 7: test.html

В контексте созданного файла, который проверим снова командой `ls -lZ`, встречаем строку `unconfined_u` - свободный пользователь, роль `object_r` используемая по умолчанию для файлов на «постоянных» носителях и на сетевых файловых системах, и тип `httpd_sys_content_t`, позволяющий процессу `httpd` получить доступ к этому файлу.



```
(tabuller@jordi)~$ ls -lZ /var/www/html/test.html
-rw-r--r--. 1 root root unconfined_u:object_r:httpd_sys_content_t:s0 33 Feb 15 08:43 /var/www/html/test.html
```

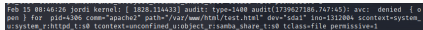
Рис. 8: Контекст test.html

Теперь командой `chcon -t samba_share_t /var/www/html/test.html` изменим контекст файла. Теперь процесс `samba` может получить доступ к файлу, а `httpd` - нет, поэтому при повторной попытке получить файл через браузер мы столкнемся с ошибкой. Соответствующее предупреждение можем видеть и в системных логах.



```
(tabuller@jordi)~$ sudo chcon -t samba_share_t /var/www/html/test.html
(tabuller@jordi)~$ ls -lZ /var/www/html/test.html
-rw-r--r--. 1 root root unconfined_u:object_r:samba_share_t:s0 33 Feb 15 08:43 /var/www/html/test.html
```

Рис. 9: Изменение контекста файла



```
Feb 15 08:46:20 jordi kernel: [ 1838.114433] audit: type=1400 audit(1739627386.747145): avc: denied { open } for pid=6386 comm="apache2" path="/var/www/html/test.html" dev="sda1" ino=1312004 scontext=system_u:system_r:httpd_t:s0 tcontext=unconfined_u:object_r:samba_share_t:s0 tclass=file permissive=1
```

Рис. 10: Сообщение об ошибке из системного лога



## Выполнение лабораторной работы

Перезапустим веб-сервер на прослушивание TCP порта 81 вместо 80, стоящего по умолчанию. Для этого в файле `/etc/apache/port.conf` заменим строчку `Listen 80` на `Listen 81`. При попытке перезапустить сервер после внесенных изменений, однако, снова столкнемся с ошибкой, потому что в конфигурацию SELinux соответствующие разрешения и изменения внесены не были.

```
Listen 81

<IfModule ssl_module>
    Listen 443
</IfModule>

<IfModule mod_gnutls.c>
    Listen 443
</IfModule>
```

Следующим шагом выполним `semanage port -a -t http_port_t -p tcp 81`. Эта команда внесет порт 81 в список доступных для сервиса `httpd`, в чем далее мы можем убедиться командой `semanage port -l | grep 81`. При попытке теперь запустить веб-сервер ошибок не возникнет.

```
(tabuller@jordi):~$ sudo semanage port -a -t http_port_t -p tcp 81
libsemanage.get_home_dirs: Error while fetching users. Returning list so far.
libsemanage.add_user: user sddm not in password file

(tabuller@jordi):~$ semanage port -l | grep 81
ValueError: SELinux policy is not managed or store cannot be accessed.

(tabuller@jordi):~$ sudo semanage port -l | grep 81
http_port_t      tcp      1782, 2207, 2280, 8290, 8292, 9100, 9101, 9102, 9220, 9221, 9222,
9280, 9281, 9282, 9290, 9291, 50000, 50002
http_cache_port_t tcp      3120, 0000, 0110, 10001-10010
http_port_t      tcp      81, 80, 443, 450, 8000, 8009, 8443, 8448
kubernetes_port_t tcp      6443, 10259, 10250-10257, 2379-2381, 10248-10250
snit_port_t      tcp      2012
puppet_port_t    tcp      8140
radioclient_port_t udp      1646, 1013
radius_port_t    udp      1645, 1012
transproxy_port_t tcp      8081
varnishd_port_t  tcp      6061-6062
zookeeper_client_port_t tcp      2181
```

Рис. 12: Добавленный в конфигурацию порт

## Выполнение лабораторной работы

В завершение работы вернем систему к настройкам по умолчанию: удалим созданный файл и добавленные контексты, а также выключим сервис apache.

```
(tabuller@jordi)~$ sudo rm /var/www/html/test.html

(tabuller@jordi)~$ sudo semanage port -d -t http_port_t -p tcp 81
libsemanage.get_hose_dirs: Error while fetching users. Returning list so far.
libsemanage.add_user: user sddm not in password file

(tabuller@jordi)~$ sudo semanage port -l | grep 81
http_port_t          tcp          1782, 2287, 2288, 8298, 8292, 9188, 9181, 9182, 9220, 9221, 9222,
9228, 9221, 9282, 9298, 9291, 50000, 50002
http_cache_port_t    tcp          3128, 8088, 8118, 10001-10010
kubernetes_port_t    tcp          6443, 10259, 10256-10257, 2379-2381, 10248-10250
monit_port_t         tcp          2112
popper_port_t        tcp          8180
radacct_port_t       udp          1646, 1813
radius_port_t        udp          1645, 1812
transproxy_port_t    tcp          8881
varnishd_port_t      tcp          6081-6082
zookeeper_client_port_t tcp          2181

(tabuller@jordi)~$ service apache2 stop
```

Рис. 13: Завершение работы

## Выводы

---

Получено первое практическое знакомство с технологией SELinux. Проверена работу SELinux на практике совместно с веб-сервером Apache.