
Front matter

lang: ru-RU title: Отчёт по лабораторной работе №6 author: Георгес
Геден institute: РУДН, Москва, Россия

date: 12 Октября 2024

Formatting

toc: false slide_level: 2 theme: metropolis header-includes:

- `\metroset{progressbar=frametitle,sectionpage=progressbar,numberi
ng=fraction}`
- `'\makeatletter'`
- `'\beamer@ignorenonframefalse'`
- `'\makeatother' aspectratio: 43 section-titles: true`

Отчет по лабораторной работе №6

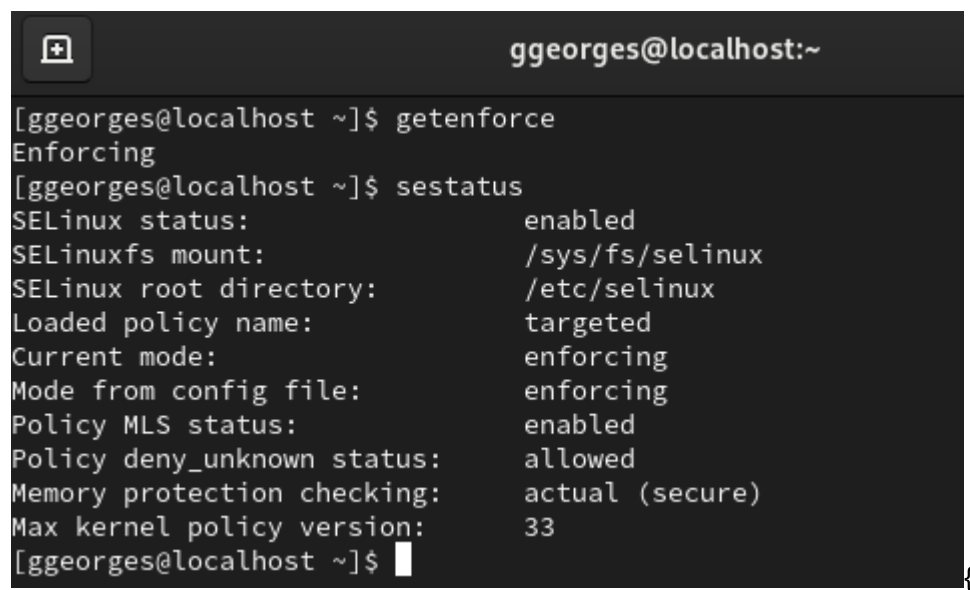
Цель работы: Развить навыки администрирования ОС Linux. Получить первое практическое знакомство с технологией SELinux. Проверить работу SELinux на практике совместно с веб-сервером Apache.

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 — роли, типы и домены. Более подробно см. в [1].

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



```
ggeorges@localhost:~  
[ggeorges@localhost ~]$ getenforce  
Enforcing  
[ggeorges@localhost ~]$ 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  
[ggeorges@localhost ~]$
```

width=70% }

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

```
ggeorges@localhost:~ — /bin/systemctl status httpd.service
[ggeorges@localhost ~]$ service httpd start
Redirecting to /bin/systemctl start httpd.service
[ggeorges@localhost ~]$ 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)
   Drop-In: /usr/lib/systemd/system/httpd.service.d
            └─php-fpm.conf
   Active: active (running) since Tue 2024-10-08 20:47:26 MSK; 1min 3s ago
     Docs: man:httpd.service(8)
  Main PID: 1102 (httpd)
    Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec: 0.00;Current workers 100/0/0/0"
      Tasks: 177 (limit: 23032)
    Memory: 38.9M
       CPU: 392ms
    CGroup: /system.slice/httpd.service
            └─1102 /usr/sbin/httpd -DFOREGROUND
               1207 /usr/sbin/httpd -DFOREGROUND
               1208 /usr/sbin/httpd -DFOREGROUND
               1209 /usr/sbin/httpd -DFOREGROUND
               1210 /usr/sbin/httpd -DFOREGROUND

oct. 08 20:47:26 localhost.localdomain systemd[1]: Starting The Apache HTTP Server: httpd.
oct. 08 20:47:26 localhost.localdomain httpd[1102]: AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 127.0.0.1 instead. Please see the README file in /usr/share/httpd for more details.
oct. 08 20:47:26 localhost.localdomain httpd[1102]: Server configured, listening on *
oct. 08 20:47:26 localhost.localdomain systemd[1]: Started The Apache HTTP Server: httpd.
lines 1-22/22 (END)...skipping...
```

{ width=70% }

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

```
ggeorges@localhost:~
[ggeorges@localhost ~]$ ps auxZ | grep httpd
system_u:system_r:httpd_t:s0 root 1102 0.0 0.2 20152 10684 ?
Ss 20:47 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 root 1105 0.0 0.7 182340 26744 ?
Ss 20:47 0:00 php-fpm: master process (/etc/php-fpm.conf)
system_u:system_r:httpd_t:s0 apache 1201 0.0 0.3 184564 12912 ?
S 20:47 0:00 php-fpm: pool www
system_u:system_r:httpd_t:s0 apache 1203 0.0 0.3 184564 12912 ?
S 20:47 0:00 php-fpm: pool www
system_u:system_r:httpd_t:s0 apache 1204 0.0 0.3 184564 13040 ?
S 20:47 0:00 php-fpm: pool www
system_u:system_r:httpd_t:s0 apache 1205 0.0 0.3 184564 12912 ?
S 20:47 0:00 php-fpm: pool www
system_u:system_r:httpd_t:s0 apache 1206 0.0 0.3 184564 13040 ?
S 20:47 0:00 php-fpm: pool www
system_u:system_r:httpd_t:s0 apache 1207 0.0 0.1 22036 7116 ?
S 20:47 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 1208 0.1 0.5 2423316 19232 ?
Sl 20:47 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 1209 0.1 0.2 2161108 10832 ?
Sl 20:47 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 1210 0.1 0.4 2161108 15084 ?
Sl 20:47 0:00 /usr/sbin/httpd -DFOREGROUND
```

{ width=70% }

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.

```
ggeorges@localhost:~  
[ggeorges@localhost ~]$ ls -lZ /var/www/  
total 4  
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_script_exec_t:s0 6  
8 août 19:30 cgi-bin  
drwxr-xr-x. 12 apache apache system_u:object_r:httpd_sys_content_t:s0 4096 1  
1 sept. 17:40 html  
[ggeorges@localhost ~]$
```

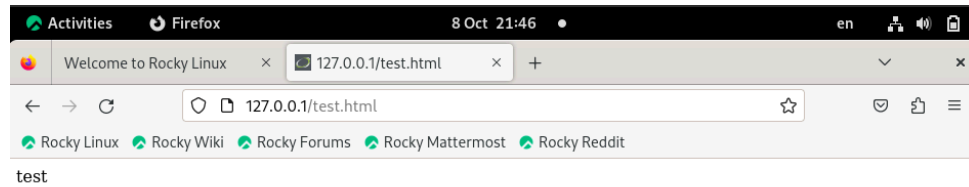
{ width=70% }

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

```
ggeorges@localhost:~  
[ggeorges@localhost ~]$ su -  
Password:  
[root@localhost ~]# touch /var/www/html/test.html  
[root@localhost ~]# emacs test.html&  
[1] 49510  
[root@localhost ~]# bash: emacs: commande inconnue...  
Les paquets fournissant ces fichiers sont :  
'emacs-nox'  
'emacs-lucid'  
'emacs'  
  
[1]+  Termine 127          emacs test.html  
[root@localhost ~]# touch /var/www/html/test.html  
[root@localhost ~]# emacs test.html&  
[1] 49522  
[root@localhost ~]# bash: emacs: commande inconnue...  
Les paquets fournissant ces fichiers sont :  
'emacs-nox'  
'emacs-lucid'  
'emacs'  
  
[1]+  Termine 127          emacs test.html  
[root@localhost ~]# cat /var/www/html/test.html  
[root@localhost ~]# nano /var/www/html/test.html  
[root@localhost ~]# cat /var/www/html/test.html  
<html>  
<body>test</body>  
</html>  
[root@localhost ~]# exit  
déconnexion  
[ggeorges@localhost ~]$
```

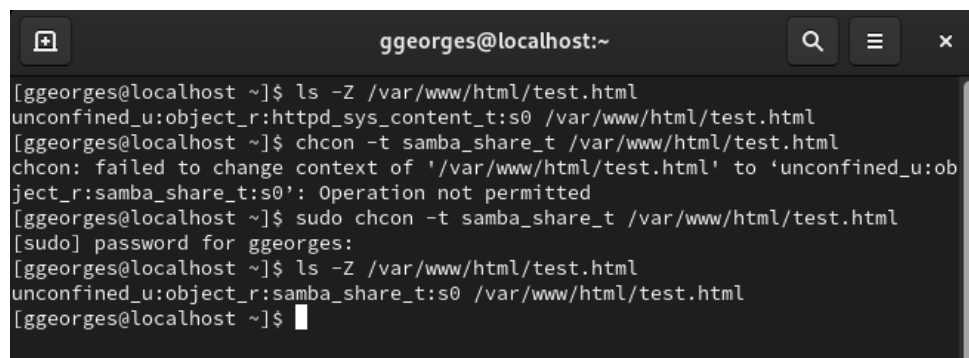
{ width=70% }

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



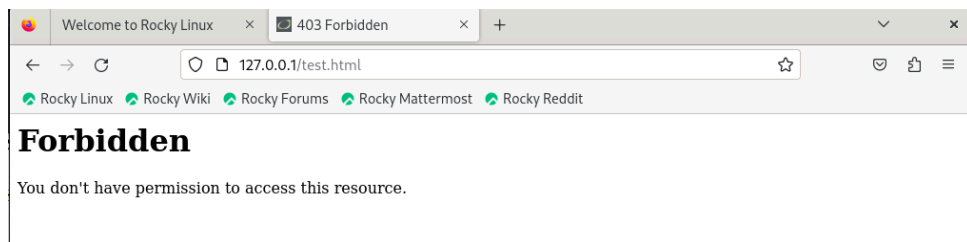
{ width=70% }

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.



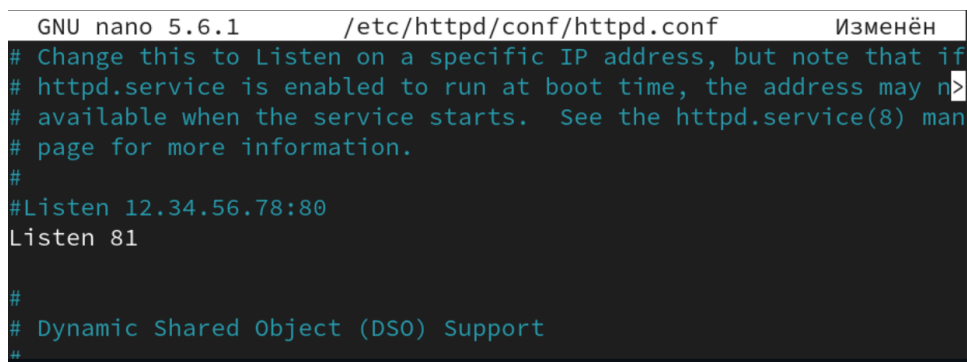
{ width=70% }

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).



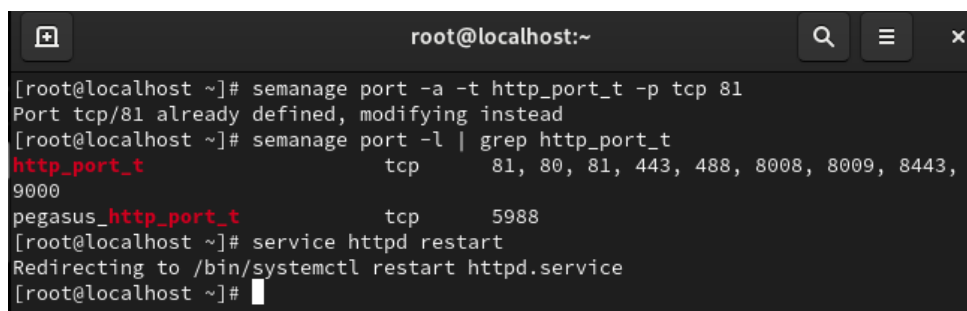
```
{ width=70% }
```

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



```
{ width=70% }
```

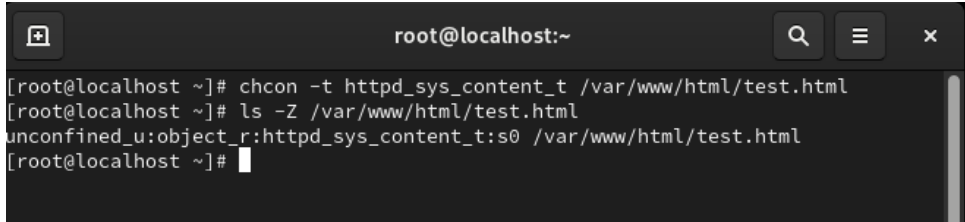
Выполняем команду `"semanage port -a -t http_port_t -p tcp 81"` и убеждаемся, что порт TCP-81 установлен. Проверяем список портов командой `"semanage port -l | grep http_port_t"`, убеждаемся, что порт 81 есть в списке и запускаем веб-сервер Apache снова.



```
{ width=70% }
```

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 titled 'root@localhost:~' with search, menu, and close buttons. It shows the following commands and output:

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

{ width=70% }

Выводы

- 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.