

# **Отчёт по лабораторной работе 6**

Соболев Максим Сергеевич

# Содержание

<b>1</b>	<b>Мандатное разграничение прав в Linux</b>	<b>4</b>
<b>2</b>	<b>Цель работы</b>	<b>5</b>
<b>3</b>	<b>Задание</b>	<b>6</b>
<b>4</b>	<b>Теоретическое введение</b>	<b>7</b>
<b>5</b>	<b>Выполнение лабораторной работы</b>	<b>8</b>
5.1	Шаг 1 . . . . .	8
5.2	Шаг 2 . . . . .	8
5.3	Шаг 3 . . . . .	9
5.4	Шаг 4 . . . . .	9
5.5	Шаг 5 . . . . .	10
5.6	Шаг 6 . . . . .	10
5.7	Шаг 7 . . . . .	11
5.8	Шаг 8 . . . . .	11
5.9	Шаг 9 . . . . .	11
5.10	Шаг 10 . . . . .	12
5.11	Шаг 11 . . . . .	12
5.12	Шаг 12 . . . . .	12
5.13	Шаг 13 . . . . .	13
5.14	Шаг 14 . . . . .	13
5.15	Шаг 15 . . . . .	14
5.16	Шаг 16 . . . . .	14
5.17	Шаг 17 . . . . .	14
5.18	Шаг 18 . . . . .	15
5.19	Шаг 19 . . . . .	15
5.20	Шаг 20 . . . . .	15
5.21	Шаг 21 . . . . .	16
5.22	Шаг 22 . . . . .	16
5.23	Шаг 23 . . . . .	17
5.24	Шаг 24 . . . . .	17
<b>6</b>	<b>Выводы</b>	<b>18</b>
	<b>Список литературы</b>	<b>19</b>

## Список иллюстраций

[illegible]

# **1 Мандатное разграничение прав в Linux**

## 2 Цель работы

Развить навыки администрирования ОС Linux. Получить первое практическое знакомство с технологией SELinux

Проверить работу SELinux на практике совместно с веб-сервером Apache

## 3 Задание

Исследовать технологию SELinux Исследовать работу SELinx на практике совместно с веб-сервером Apache

## 4 Теоретическое введение

SELinux — реализация системы принудительного контроля доступа, которая может работать параллельно с классической избирательной системой контроля доступа.

Apache HTTP-сервер — свободный веб-сервер. Apache является кроссплатформенным ПО, поддерживает операционные системы Linux, BSD, Mac OS, Microsoft Windows, Novell NetWare, BeOS. Основными достоинствами Apache считаются надёжность и гибкость конфигурации.

---

## 5 Выполнение лабораторной работы

### 5.1 Шаг 1

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
```

Рис. 5.1: 1

### 5.2 Шаг 2

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
```

Рис. 5.2: 2



## 5.3 Шаг 3

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
```

Рис. 5.3: 3

## 5.4 Шаг 4

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
httpd_enable_cgi                 on
httpd_enable_ftp_server          off
```

Рис. 5.4: 4

## 5.5 Шаг 5

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 Constrain:           72        MLS Val. Tran:         0
Permissives:             0         Polcap:                5
Defaults:                7         Typebounds:            0
Allowxperm:              0         Neverallowxperm:       0
Auditallowxperm:         0         Dontauditxperm:        0
Ibendportcon:            0         Ibpkeycon:             0
Initial SIDs:            27         Fs_use:                33
Genfscon:                106       Portcon:               651
Netifcon:                0         Nodecon:               0
```

Рис. 5.5: 5

## 5.6 Шаг 6

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
```

Рис. 5.6: 6

## 5.7 Шаг 7

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
```

Рис. 5.7: 7

Файлов нет

## 5.8 Шаг 8

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
```

Рис. 5.8: 8

Создание файлов разрешено только пользователю root

## 5.9 Шаг 9

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>
```

Рис. 5.9: 9

## 5.10 Шаг 10

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
```

Рис. 5.10: 10

## 5.11 Шаг 11

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

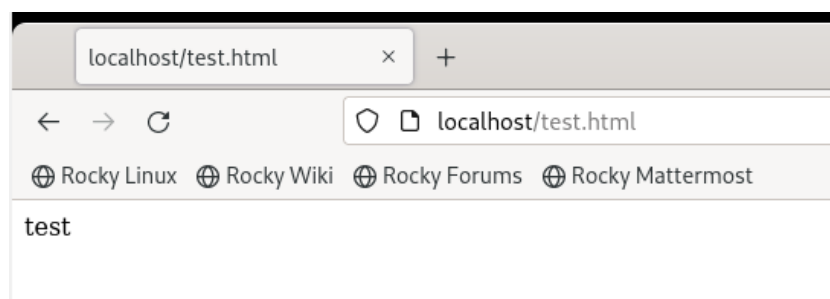


Рис. 5.11: 11

## 5.12 Шаг 12

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

Изучили.

## 5.13 Шаг 13

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]#
```

Рис. 5.12: 13

## 5.14 Шаг 14

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.`

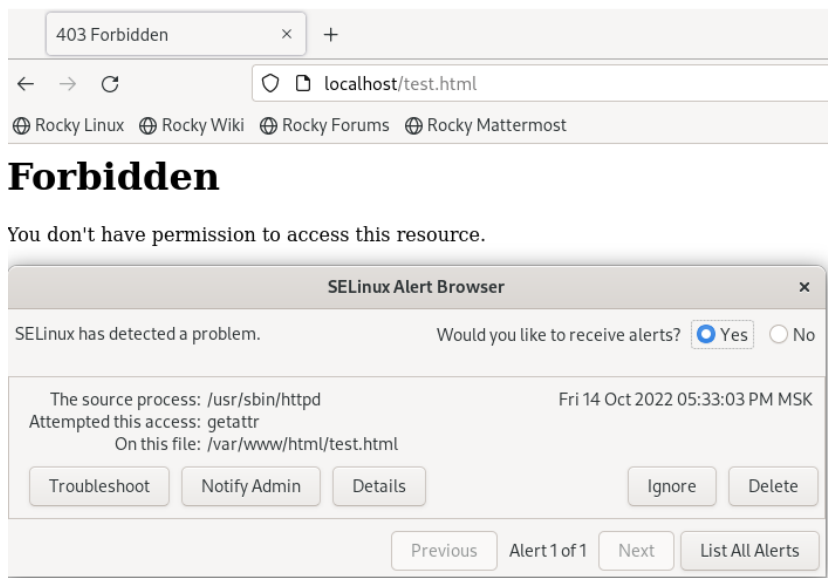


Рис. 5.13: 14

## 5.15 Шаг 15

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 permission=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" EGD="apache" SGID="apache" FSGID="apache"
type=PROCTITLE msg=audit(1665757983.560:204): proctitle=2F7573722F7362696E2F6974747064002D44464F524547524F554E44
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 permission=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" EGD="apache" SGID="apache" FSGID="apache"
type=PROCTITLE msg=audit(1665757983.560:205): proctitle=2F7573722F7362696E2F6974747064002D44464F524547524F554E44
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"
```

Рис. 5.14: 15

## 5.16 Шаг 16

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

```
#Listen 12.34.56.78
Listen 81
```

Рис. 5.15: 16

## 5.17 Шаг 17

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.

## 5.18 Шаг 18

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

## 5.19 Шаг 19

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

```

Рис. 5.16: 19

## 5.20 Шаг 20

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.

```

Рис. 5.17: 20

## 5.21 Шаг 21

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».

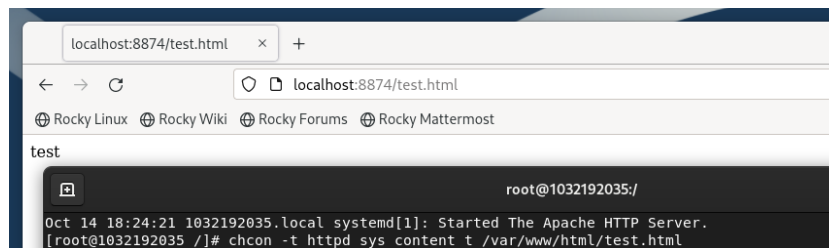


Рис. 5.18: 21

## 5.22 Шаг 22

Исправим обратно конфигурационный файл `apache`, вернув `Listen 80`



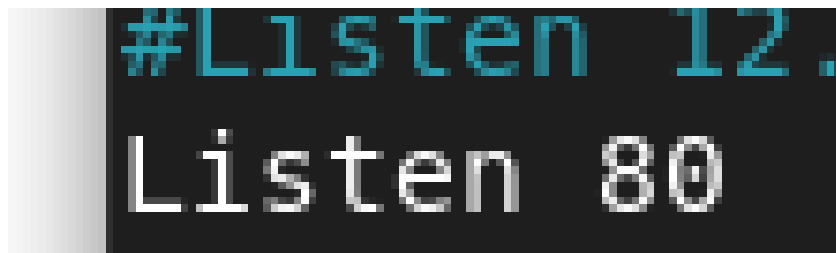
A terminal window with a black background. The first line shows a blue prompt character followed by the text "#Listen 12.". The second line shows the text "Listen 80" in white.

Рис. 5.19: 22

## 5.23 Шаг 23

Удалим привязку http\_port\_t к 8874 порту: semanage port -d -t http\_port\_t -p tcp 8874 и проверим, что порт 8874 удалён

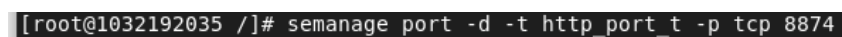
A terminal window with a black background. The prompt is "[root@1032192035 /]#". The command being executed is "semanage port -d -t http\_port\_t -p tcp 8874".

Рис. 5.20: 23

## 5.24 Шаг 24

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

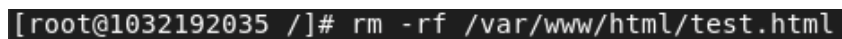
A terminal window with a black background. The prompt is "[root@1032192035 /]#". The command being executed is "rm -rf /var/www/html/test.html".

Рис. 5.21: 24

## 6 Выводы

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

Проверили работу SELinux на практике совместно с веб-сервером Apache

# Список литературы

1. xattr(7) — Linux manual page // Linux man-pages project URL: <https://man7.org/linux/man-pages/man7/xattr.7.html> (дата обращения: 30.09.2022).