

# **Отчёт по лабораторной работе №6**

**Основы информационной безопасности**

Мурашов Иван Вячеславович

# Содержание

|   |                                |    |
|---|--------------------------------|----|
| 1 | Цель работы                    | 5  |
| 2 | Теоретическое введение         | 6  |
| 3 | Выполнение лабораторной работы | 8  |
| 4 | Выводы                         | 16 |

## Список иллюстраций

|      |  |    |
|------|--|----|
| 3.1  | проверка режима работы SELinux . . . . .   | 8  |
| 3.2  | Проверка работы Apache . . . . .           | 9  |
| 3.3  | Контекст безопасности Apache . . . . .     | 9  |
| 3.4  | Состояние переключателей SELinux . . . . . | 10 |
| 3.5  | Статистика по политике . . . . .           | 11 |
| 3.6  | Типы поддиректорий . . . . .               | 11 |
| 3.7  | Типы файлов . . . . .                      | 11 |
| 3.8  | Создание файла . . . . .                   | 12 |
| 3.9  | Контекст файла . . . . .                   | 12 |
| 3.10 | Отображение файла . . . . .                | 13 |
| 3.11 | Изучение справки по команде . . . . .      | 14 |
| 3.12 | Изменение контекста . . . . .              | 14 |
| 3.13 | Отображение файла . . . . .                | 14 |
| 3.14 | Попытка прочесть лог-файл . . . . .        | 15 |
| 3.15 | Изменение файла . . . . .                  | 15 |

## Список таблиц

# 1 Цель работы

Развить навыки администрирования ОС Linux. Получить первое практическое знакомство с технологией SELinux<sup>1</sup>. Проверить работу SELinx на практике совместно с веб-сервером Apache. [course?]

## 2 Теоретическое введение

1. **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 [f?].

2. **Apache** — это свободное программное обеспечение, с помощью которого можно создать веб-сервер. Данный продукт возник как доработанная версия другого HTTP-клиента от национального центра суперкомпьютерных приложений (NCSA).

*Для чего нужен Apache сервер:*

- чтобы открывать динамические PHP-страницы,
- для распределения поступающей на сервер нагрузки,
- для обеспечения отказоустойчивости сервера,
- чтобы потренироваться в настройке сервера и запуске PHP-скриптов.

Apache является кроссплатформенным ПО и поддерживает такие операционные системы, как Linux, BSD, MacOS, Microsoft, BeOS и другие.

Более подробно см. в [s?].

### 3 Выполнение лабораторной работы

Вхожу в систему под своей учетной записью. Убеждаюсь, что SELinux работает в режиме enforcing политики targeted с помощью команд `getenforce` и `sestatus` (рис. 3.1).

```
[ivmurashov@ivmurashov ~]$ getenforce
Enforcing
[ivmurashov@ivmurashov ~]$ 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
[ivmurashov@ivmurashov ~]$
```

Рис. 3.1: проверка режима работы SELinux

Запускаю сервер `apache`, далее обращаюсь с помощью браузера к веб-серверу, запущенному на компьютере, он работает, что видно из вывода команды `service httpd status` (рис. 3.2).



```
[ivmurashov@ivmurashov ~]$ sudo systemctl start httpd
[sudo] password for ivmurashov:
[ivmurashov@ivmurashov ~]$ sudo systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[ivmurashov@ivmurashov ~]$ 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 Sat 2025-05-03 00:23:55 MSK; 40s ago
     Docs: man:httpd.service(8)
   Main PID: 7378 (httpd)
    Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec: 0 B/sec"
      Tasks: 177 (limit: 29427)
    Memory: 32.5M
       CPU: 453ms
    CGroup: /system.slice/httpd.service
            └─7378 /usr/sbin/httpd -DFOREGROUND
              7379 /usr/sbin/httpd -DFOREGROUND
              7380 /usr/sbin/httpd -DFOREGROUND
              7381 /usr/sbin/httpd -DFOREGROUND
              7382 /usr/sbin/httpd -DFOREGROUND

May 03 00:23:55 ivmurashov.localdomain systemd[1]: Starting The Apache HTTP Server...
May 03 00:23:55 ivmurashov.localdomain httpd[7378]: Server configured, listening on: port 80
May 03 00:23:55 ivmurashov.localdomain systemd[1]: Started The Apache HTTP Server.
lines 1-19/19 (END)
• httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
   Active: active (running) since Sat 2025-05-03 00:23:55 MSK; 40s ago
     Docs: man:httpd.service(8)
   Main PID: 7378 (httpd)
    Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec: 0 B/sec"
      Tasks: 177 (limit: 29427)
    Memory: 32.5M
       CPU: 453ms
    CGroup: /system.slice/httpd.service
            └─7378 /usr/sbin/httpd -DFOREGROUND
              7379 /usr/sbin/httpd -DFOREGROUND
              7380 /usr/sbin/httpd -DFOREGROUND
              7381 /usr/sbin/httpd -DFOREGROUND
              7382 /usr/sbin/httpd -DFOREGROUND

May 03 00:23:55 ivmurashov.localdomain systemd[1]: Starting The Apache HTTP Server...
May 03 00:23:55 ivmurashov.localdomain httpd[7378]: Server configured, listening on: port 80
May 03 00:23:55 ivmurashov.localdomain systemd[1]: Started The Apache HTTP Server.
```

Рис. 3.2: Проверка работы Apache

С помощью команды `ps auxZ | grep httpd` нашла веб-сервер Apache в списке процессов. Его контекст безопасности - `httpd_t` (рис. 3.3).

```
[ivmurashov@ivmurashov ~]$ ps auxZ | grep httpd
system_u:system_r:httpd_t:s0 root      7378  0.0  0.2 21232 11472 ?        Ss   00:23   0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache  7379  0.0  0.1 22964  7556 ?        S   00:23   0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache  7380  0.1  0.3 2358708 17356 ?       Sl   00:23   0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache  7381  0.1  0.2 2162036 10976 ?       Sl   00:23   0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache  7382  0.1  0.2 2162036 13244 ?       Sl   00:23   0:00 /usr/sbin/httpd -DFOREGROUND
unconfined_u:unconfined_r:unconfined_t:s0-cb:c1023 ivmuras+ 7721  0.0  0.0 221060 2304 pts/0 R+  00:26   0:00 grep --color=auto httpd
```

Рис. 3.3: Контекст безопасности Apache

Просматриваю текущее состояние переключателей SELinux для Apache с помощью команды `sestatus -bigrep httpd` (рис. 3.4).

```

[ivmurashov@ivmurashov ~]$ sestatus -b httpd
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

Policy booleans:
abrt_anon_write                off
abrt_handle_event              off
abrt_upload_watch_anon_write   on
antivirus_can_scan_system      off
antivirus_use_jit              off
auditadm_exec_content          on
authlogin_nsswitch_use_ldap     off
authlogin_radius               off
authlogin_yubikey              off
awstats_purge_apache_log_files off
boinc_execmem                  on
cdrecord_read_content           off
cluster_can_network_connect     off
cluster_manage_all_files        off
cluster_use_execmem             off
cobbler_anon_write             off
cobbler_can_network_connect     off
cobbler_use_cifs                off
cobbler_use_nfs                off
collectd_tcp_network_connect   off
colord_use_nfs                 off
condor_tcp_network_connect      off
conman_can_network              off

```

Рис. 3.4: Состояние переключателей SELinux

Просмотрел статистику по политике с помощью команды `seinfo`. Множество пользователей - 8, ролей - 39, типов - 5135. (рис. 3.5).

|  |        |                  |      |
|--|--------|------------------|------|
| 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:   | 5187   | Attributes:      | 259  |
| Users:   | 8      | Roles:           | 15   |
| Booleans:  | 358    | Cond. Expr.:     | 390  |
| Allow:   | 66245  | Neverallow:      | 0    |
| Auditallow:  | 178    | Dontaudit:       | 8723 |
| Type_trans:  | 274461 | 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:                                       | 6      | Polcap:          | 6    |
| Defaults:  | 7      | Typebounds:      | 0    |
| Allowxperm:  | 0      | Neverallowxperm: | 0    |
| Auditallowxperm:                                   | 0      | Dontauditxperm:  | 0    |
| Ibendportcon:                                      | 0      | Ibpkeycon:       | 0    |
| Initial SIDs:                                      | 27     | Fs_use:          | 35   |
| Genfscon:  | 109    | Portcon:         | 665  |
| Netifcon:  | 0      | Nodecon:         | 0    |

Рис. 3.5: Статистика по политике

Типы поддиректорий, находящихся в директории /var/www, с помощью команды `ls -lZ /var/www` следующие: владелец - root, права на изменения только у владельца. Файлов в директории нет (рис. 3.6).

```
[ivmurashov@ivmurashov ~]$ ls -lZ /var/www
total 0
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_script_exec_t:s0 6 Jan 22 03:25 cgi-bin
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_content_t:s0 6 Jan 22 03:25 html
```

Рис. 3.6: Типы поддиректорий

В директории /var/www/html нет файлов. (рис. 3.7).

```
[ivmurashov@ivmurashov ~]$ ls -lZ /var/www/html
total 0
[ivmurashov@ivmurashov ~]$
```

Рис. 3.7: Типы файлов

Создать файл может только суперпользователь, поэтому от его имени создаем файл touch.html со следующим содержанием:

```
<html>
<body>test</body>
</html>
```

(рис. 3.8).

```
[ivmurashov@ivmurashov ~]$ sudo touch /var/www/html/test.html
[sudo] password for ivmurashov:
[ivmurashov@ivmurashov ~]$ sudo nano /var/www/html/test.html
[ivmurashov@ivmurashov ~]$ sudo cat /var/www/html/test.html
<html>
<body>test</body>
</html>
[ivmurashov@ivmurashov ~]$ sudo nano /var/www/html/test.html
[ivmurashov@ivmurashov ~]$ sudo cat /var/www/html/test.html
<html>
<body>test</body>
</html>
[ivmurashov@ivmurashov ~]$
```

Рис. 3.8: Создание файла

Проверяю контекст созданного файла. По умолчанию это httpd\_sys\_content\_t (рис. 3.9).

```
[ivmurashov@ivmurashov ~]$ ls -lZ /var/www/html
total 4
-rw-r--r--. 1 root root unconfined_u:object_r:httpd_sys_content_t:s0 33 May  3 00:31 test.html
[ivmurashov@ivmurashov ~]$
```

Рис. 3.9: Контекст файла

Обращаюсь к файлу через веб-сервер, введя в браузере адрес <http://127.0.0.1/test.html>. Файл был успешно отображён (рис. 3.10).

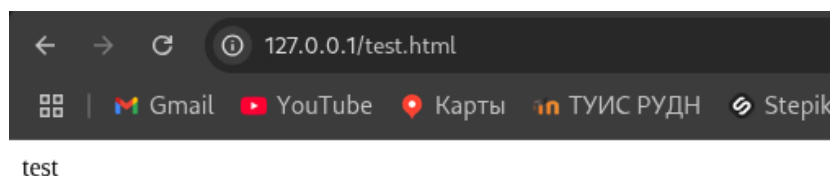


Рис. 3.10: Отображение файла

Изучаю справку `man httpd_selinux`. Рассмотрим полученный контекст детально. Так как по умолчанию пользователи CentOS являются свободными от типа (`unconfined` в переводе с англ. означает свободный), созданному нами файлу `test.html` был сопоставлен SELinux, пользователь `unconfined_u`. Это первая часть контекста. Далее политика ролевого разделения доступа RBAC используется процессами, но не файлами, поэтому роли не имеют никакого значения для файлов. Роль `object_r` используется по умолчанию для файлов на «постоянных» носителях и на сетевых файловых системах. (В директории `/ргос` файлы, относящиеся к процессам, могут иметь роль `system_r`. Если активна политика MLS, то могут использоваться и другие роли, например, `secadm_r`. Данный случай мы рассматривать не будем, как и предназначение `:s0`). Тип `httpd_sys_content_t` позволяет процессу `httpd` получить доступ к файлу. Благодаря наличию последнего типа мы получили доступ к файлу при обращении к нему через браузер. (рис. 3.11).

```
HTTPD(8)                                httpd                                HTTPD(8)
NAME
    httpd - Apache Hypertext Transfer Protocol Server
SYNOPSIS
    httpd [-d serverroot] [-f config] [-e directive] [-o parameter] [-s level] [-e file] [-k start|restart|graceful|stop|graceful-stop] [-h] [-l] [-L] [-t] [-v] [-V] [-X] [-M] [-T]
    On Windows systems, the following additional arguments are available:
    httpd [-k install|config|uninstall] [-n name] [-w]
SUMMARY
    httpd is the Apache HyperText Transfer Protocol (HTTP) server program. It is designed to be run as a standalone daemon process. When used like this it will create a pool of child processes or threads to handle requests.
    In general, httpd should not be invoked directly, but rather should be invoked via apachectl on Unix-based systems or as a service on Windows NT, 2000 and XP and as a console application on Windows 9x and ME.
OPTIONS
    -d serverroot
        Set the initial value for the ServerRoot directive to serverroot. This can be overridden by the ServerRoot directive in the configuration file. The default is /etc/httpd.
    -f config
        Uses the directives in the file config on startup. If config does not begin with a /, then it is taken to be a path relative to the ServerRoot. The default is conf/httpd.conf.
    -k start|restart|graceful|stop|graceful-stop
        Signals httpd to start, restart, or stop. See Stopping Apache httpd for more information.
```

Рис. 3.11: Изучение справки по команде

Изменяю контекст файла /var/www/html/test.html с httpd\_sys\_content\_t на любой другой, к которому процесс httpd не должен иметь доступа, например, на samba\_share\_t: `chcon -t samba_share_t /var/www/html/test.html ls -Z /var/www/html/test.html` Контекст действительно поменялся (рис. 3.12).

```
[ivmurashov@ivmurashov ~]$ sudo chcon -t samba_share_t /var/www/html/test.html
[ivmurashov@ivmurashov ~]$ ls -lZ /var/www/html
total 4
-rw-r--r--. 1 root root unconfined_u:object_r:samba_share_t:s0 33 May  3 00:31 test.html
[ivmurashov@ivmurashov ~]$
```

Рис. 3.12: Изменение контекста

При попытке отображения файла в браузере получаем сообщение об ошибке (рис. 3.13).

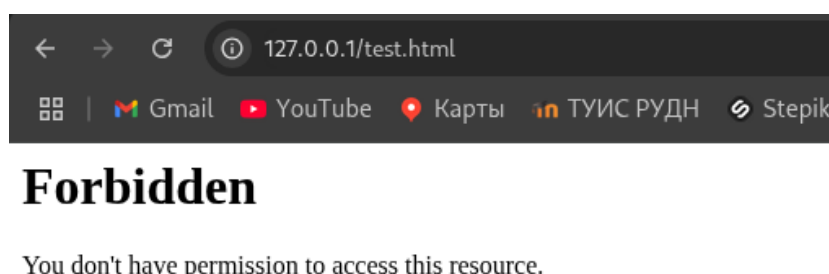


Рис. 3.13: Отображение файла

файл не был отображён, хотя права доступа позволяют читать этот файл любому пользователю, потому что установлен контекст, к которому процесс httpd не должен иметь доступа.

Просматриваю log-файлы веб-сервера Apache и системный лог-файл `tail /var/log/messages`. Если в системе окажутся запущенными процессы `setroubleshootd` и `audtd`, то вы также сможете увидеть ошибки, аналогичные указанным выше, в файле `/var/log/audit/audit.log`. (рис. 3.14).

[illegible]

Рис. 3.14: Попытка прочесть лог-файл

Чтобы запустить веб-сервер Apache на прослушивание TCP-порта 81 (а не 80, как рекомендует IANA и прописано в /etc/services) открываю файл /etc/httpd/httpd.conf для изменения. (рис. 3.15).

```

imvurashov@imvurashov:~ — sudo nano /etc/httpd/conf/httpd.conf
GNU nano 5.6.1 /etc/httpd/conf/httpd.conf
#
# Do not add a slash at the end of the directory path. If you point
# ServerRoot to a non-local disk, be sure to specify a local disk on the
# Mutex directive, if file-based mutexes are used. If you wish to share the
# same ServerRoot for multiple httpd daemons, you will need to change at
# least PidFile.
#
ServerRoot "/etc/httpd"
#
#
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, instead of the default. See also the <VirtualHost>
# directive.
#
# Change this to Listen on a specific IP address, but note that if
# httpd.service is enabled to run at boot time, the address may not be
# available when the service starts. See the httpd.service(8) man
# page for more information.
#
#Listen 12.34.56.78:80
Listen 81
#
# Dynamic Shared Object (DSO) Support
#
# To be able to use the functionality of a module which was built as a DSO you
# have to place corresponding 'LoadModule' lines at this location so the

```

Рис. 3.15: Изменение файла

Нахожу строчку Listen 80 и заменяю её на Listen 81

## 4 Выводы

В ходе выполнения данной лабораторной работы были развиты навыки администрирования ОС Linux, получено первое практическое знакомство с технологией SELinux и проверена работа SELinux на практике совместно с веб-сервером Apache.