

Отчет по лабораторной работе №6

Дисциплина: Основы информационной безопасности

Иванов Сергей Владимирович

Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
3	Вывод	15

Список иллюстраций

2.1	Проверка режима работы SELinux	5
2.2	Проверка службы apache	6
2.3	Проверка работы Apache	6
2.4	Состояние переключателей	6
2.5	Статистика по политике	7
2.6	Типы поддиректорий	7
2.7	Тип файлов	8
2.8	Создание файла	8
2.9	Проверка контекста	8
2.10	Отображение файла	8
2.11	Изучение справки	9
2.12	Изменение контекста файла	10
2.13	Отображение файла(ошибка)	10
2.14	Просмотр log-файлов	10
2.15	Изменение номера порта	11
2.16	Изменение порта	11
2.17	Попытка прослушивания 81 порта	12
2.18	log-файл	12
2.19	log-файлы	13
2.20	Добавление и проверка портов	13
2.21	Перезапуск сервера	13
2.22	Проверка сервера	13
2.23	Удаление порта	14
2.24	Удаление файла	14

1 Цель работы

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

2 Выполнение лабораторной работы

Убедимся, что SELinux работает в режиме enforcing политики targeted с помощью команд `getenforce` и `sestatus` (рис. 1)

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

Рис. 2.1: Проверка режима работы SELinux

Запускаю сервер `apache`, проверяю статус службы, убеждаюсь что она запущена. (рис. 2).

```
[svivanov@svivanov ~]$ sudo systemctl start httpd
[svivanov@svivanov ~]$ sudo systemctl enable httpd
[svivanov@svivanov ~]$ 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 Thu 2025-04-17 12:43:06 MSK; 3min 8s ago
     Docs: man:httpd.service(8)
  Main PID: 16856 (httpd)
   Status: "Total requests: 0; Idle/Busy workers 100/0; Requests/sec: 0; Bytes served/sec: 0"
    Tasks: 177 (limit: 23029)
   Memory: 32.4M
      CPU: 182ms
   CGroup: /system.slice/httpd.service
           └─16856 /usr/sbin/httpd -DFOREGROUND
             └─17114 /usr/sbin/httpd -DFOREGROUND
               └─17115 /usr/sbin/httpd -DFOREGROUND
                 └─17116 /usr/sbin/httpd -DFOREGROUND
                   └─17117 /usr/sbin/httpd -DFOREGROUND

Apr 17 12:43:06 svivanov.localdomain systemd[1]: Starting The Apache HTTP Server:
Apr 17 12:43:06 svivanov.localdomain httpd[16856]: Server configured, listening on:
Apr 17 12:43:06 svivanov.localdomain systemd[1]: Started The Apache HTTP Server:
lines 1-19/19 (END)
```

Рис. 2.2: Проверка службы apache

Найдем веб-сервер Apache в списке процессов, определим его контекст безопасности командой `ps -eZ | grep httpd`. Его контекст безопасности `httpd_t` (рис. 3).

```
[svivanov@svivanov ~]$ ps -eZ | grep httpd
system_u:system_r:httpd_t:s0      16856 ?        00:00:00 httpd
system_u:system_r:httpd_t:s0      17114 ?        00:00:00 httpd
system_u:system_r:httpd_t:s0      17115 ?        00:00:00 httpd
system_u:system_r:httpd_t:s0      17116 ?        00:00:00 httpd
system_u:system_r:httpd_t:s0      17117 ?        00:00:00 httpd
[svivanov@svivanov ~]$
```

Рис. 2.3: Проверка работы Apache

Посмотрим текущее состояние переключателей SELinux для Apache с помощью команды `sestatus -bigrep httpd` (рис. 4).

```
[svivanov@svivanov ~]$ 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
```

Рис. 2.4: Состояние переключателей

Посмотрим статистику по политике с помощью команды `seinfo`, множество пользователей - 8, ролей - 15, типов - 5169 (рис. 5).

```
[svivanov@svivanov ~]$ 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:                    5169    Attributes:           259
Users:                    8       Roles:                15
Booleans:                 358     Cond. Expr.:         390
Allow:                    65633   Neverallow:           0
Auditallow:               176     Dontaudit:            8703
Type_trans:               271851  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:               1      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
[svivanov@svivanov ~]$
```

Рис. 2.5: Статистика по политике

Определим тип файлов и поддиректорий, находящихся в директории `/var/www`, с помощью команды `ls -lZ /var/www`. Файлов 0, 2 поддиректории, владелец - root (рис. 6).

```
[svivanov@svivanov ~]$ 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
[svivanov@svivanov ~]$
```

Рис. 2.6: Типы поддиректорий

Определим тип файлов, находящихся в директории `/var/www/html`: `ls -lZ /var/www/html`. Файлов нет (рис. 7).

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

Рис. 2.7: Тип файлов

Создадим от имени суперпользователя html-файл `/var/www/html/test.html` следующего содержания: (рис. 8).

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

```
[svivanov@svivanov html]$ sudo touch test.html
[svivanov@svivanov html]$ sudo nano /var/www/html/html.test
[svivanov@svivanov html]$ sudo cat /var/www/html/html.test
<html>
<body>test</body>
</html>
[svivanov@svivanov html]$
```

Рис. 2.8: Создание файла

Проверим контекст созданного файла. Контекст - `httpd_sys_content_d` (рис. 9).

```
[svivanov@svivanov html]$ ls -lZ /var/www/html
total 4
-rw-r--r--. 1 root root unconfined_u:object_r:httpd_sys_content_t:s0 33 Apr 17 1
2:59 html.test
-rw-r--r--. 1 root root unconfined_u:object_r:httpd_sys_content_t:s0  0 Apr 17 1
2:56 test.html
[svivanov@svivanov html]$
```

Рис. 2.9: Проверка контекста

Обратимся к файлу через веб-сервер, введя в браузере адрес `http://127.0.0.1/test.html`. Убедимся, что файл был успешно отображён. (рис. 10).

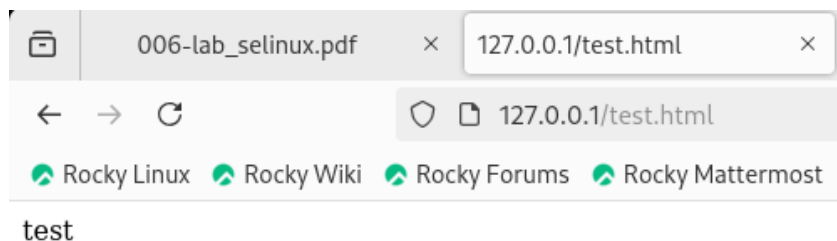
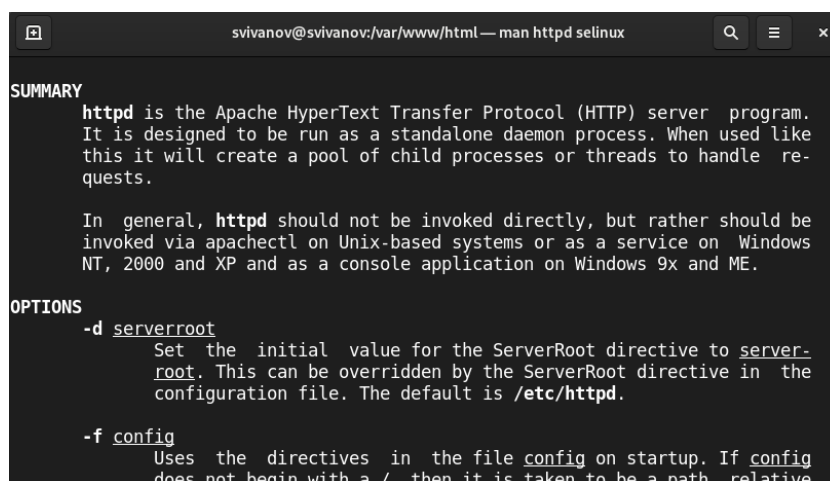


Рис. 2.10: Отображение файла

Изучим справку `man httpd_selinux`. Так как по умолчанию пользователи CentOS являются свободными от типа (`unconfined` в переводе с англ. 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 `/proc` 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. 11).



```
svivanov@svivanov:/var/www/html — man httpd_selinux
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 re-
  quests.

  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 server-
    root. 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
```

Рис. 2.11: Изучение справки

Изменим контекст файла `/var/www/html/test.html` с `httpd_sys_content_t`, на-
пример, на `samba_share_t`: `chcon -t samba_share_t /var/www/html/test.html ls -Z`
`/var/www/html/test.html` (рис. 12).

```
[svivanov@svivanov html]$ sudo chcon -t samba_share_t /var/www/html/test.html
[svivanov@svivanov html]$ ls -Z /var/www/html/test.html
unconfined_u:object_r:samba_share_t:s0 /var/www/html/test.html
[svivanov@svivanov html]$
```

Рис. 2.12: Изменение контекста файла

Попробуем ещё раз получить доступ к файлу через веб-сервер, введя в браузере адрес `http://127.0.0.1/test.html`. Мы получили сообщение об ошибке: (рис. 13).

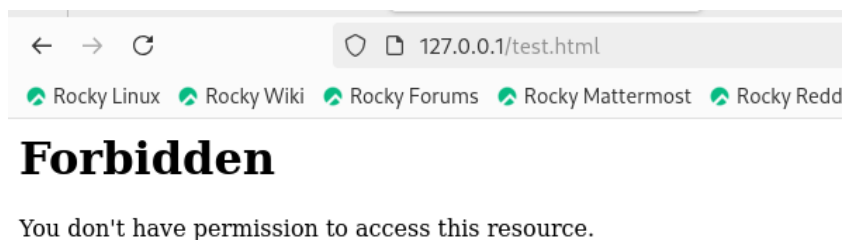


Рис. 2.13: Отображение файла(ошибка)

Файл не был отображен потому что мы установили контекст, к которому процесс `httpd` не имеет доступа. Просмотрим `log`-файлы веб-сервера Apache. Также посмотрим системный `log`-файл: `tail /var/log/messages`. (рис. 14)

```
[svivanov@svivanov html]$ sudo tail /var/log/messages
Apr 17 13:12:37 svivanov systemd[1]: Started SETroubleshoot daemon for processing new SELinux denial logs.
Apr 17 13:12:37 svivanov setroubleshoot[44533]: failed to retrieve rpm info for path '/var/www/html/test.html':
Apr 17 13:12:37 svivanov systemd[1]: Created slice Slice /system/dbus-:1.1-org.fedoraproject.SetroubleshootPrivileged.
Apr 17 13:12:37 svivanov systemd[1]: Started dbus-:1.1-org.fedoraproject.SetroubleshootPrivileged@0.service.
Apr 17 13:12:38 svivanov setroubleshoot[44533]: SELinux is preventing /usr/sbin/httpd from getattr access on the file /var/www/html/test.html. For complete SELinux messages run: sealert -l d74971b5-b6b4-4869-aeab-53e1cbd95b8b
Apr 17 13:12:38 svivanov setroubleshoot[44533]: SELinux is preventing /usr/sbin/httpd from getattr access on the file /var/www/html/test.html.#012#012***** Plugging restorecon (92.2 confidence) suggests *****#012#012If you want to fix the label. #012/var/www/html/test.html default label should be httpd_sys_content_t.#012Then you can run restorecon. The access attempt may have been stopped due to insufficient permissions to access a parent directory in which
```

Рис. 2.14: Просмотр `log`-файлов

Попробуем запустить веб-сервер Apache на прослушивание TCP-порта 81. Для этого в файле `/etc/httpd/httpd.conf` найдем строчку `Listen 80` и заменим её на `Listen 81`. (рис. 15)

```
[svivanov@svivanov httpd]$ cd /etc/httpd/conf
[svivanov@svivanov conf]$ ls
httpd.conf  magic
[svivanov@svivanov conf]$ nano /etc/httpd/conf/httpd.conf
[svivanov@svivanov conf]$ sudo nano /etc/httpd/conf/httpd.conf
[svivanov@svivanov conf]$
```

Рис. 2.15: Изменение номера порта

Изменение порта (рис. 16)

```
# page for more information.
#
#Listen 12.34.56.78:80
Listen 81

#
# Dynamic Shared Object (DSO) Sup
```

Рис. 2.16: Изменение порта

Выполним перезапуск веб-сервера Apache. Произошёл сбой, потому что порт 81 не добавлен в список прослушиваемых портов (рис. 17)

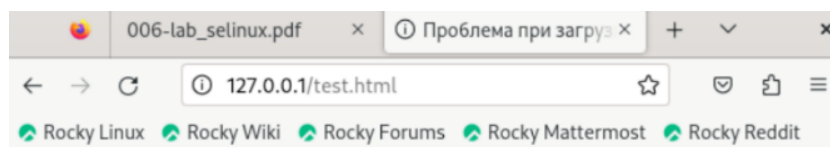


Рис. 2.17: Попытка прослушивания 81 порта

Проанализируем лог-файлы: `tail -nl /var/log/messages` (рис. 18)

```
[svivanov@svivanov ~]$ sudo tail /var/log/messages
Apr 17 14:11:32 svivanov systemd[1]: Starting The Apache HTTP Server...
Apr 17 14:11:32 svivanov httpd[47839]: Server configured, listening on: port 80
Apr 17 14:11:32 svivanov systemd[1]: Started The Apache HTTP Server.
Apr 17 14:11:57 svivanov systemd[1]: Stopping The Apache HTTP Server...
Apr 17 14:11:58 svivanov systemd[1]: httpd.service: Deactivated successfully.
Apr 17 14:11:58 svivanov systemd[1]: Stopped The Apache HTTP Server.
Apr 17 14:11:58 svivanov systemd[1]: Starting The Apache HTTP Server...
Apr 17 14:11:58 svivanov httpd[48041]: Server configured, listening on: port 80
Apr 17 14:11:58 svivanov systemd[1]: Started The Apache HTTP Server.
Apr 17 14:12:14 svivanov systemd[1]: packagekit.service: Deactivated successfully.
[svivanov@svivanov ~]$
```

Рис. 2.18: log-файл

Просмотрим файлы `/var/log/http/error_log`, `/var/log/http/access_log` и `/var/log/audit/audit.log`.
Записи появились в файле `error_log` (рис. 19)

```
[svivanov@svivanov ~]$ sudo cat /var/log/httpd/error_log
[Thu Apr 17 12:43:06.899398 2025] [core:notice] [pid 16856:tid 16856] SELinux po
lity enabled; httpd running as context system_u:system_r:httpd_t:s0
[Thu Apr 17 12:43:06.900752 2025] [suexec:notice] [pid 16856:tid 16856] AH01232:
suEXEC mechanism enabled (wrapper: /usr/sbin/suexec)
[Thu Apr 17 12:43:06.922589 2025] [lbmethod_heartbeat:notice] [pid 16856:tid 168
56] AH02282: No slotmem from mod_heartbeat
[Thu Apr 17 12:43:06.924638 2025] [mpm_event:notice] [pid 16856:tid 16856] AH004
89: Apache/2.4.62 (Rocky Linux) configured -- resuming normal operations
[Thu Apr 17 12:43:06.924658 2025] [core:notice] [pid 16856:tid 16856] AH00094: C
ommand line: '/usr/sbin/httpd -D FOREGROUND'
[Thu Apr 17 13:12:36.927244 2025] [core:error] [pid 17116:tid 17254] (13)Permiss
ion denied: [client 127.0.0.1:60068] AH00035: access to /test.html denied (files
ystem path '/var/www/html/test.html') because search permissions are missing on
```

Рис. 2.19: log-файлы

Выполним команду `semanage port -a -t http_port_t -p tcp 81` После этого прове-
рим список портов командой `semanage port -l | grep http_port_t` (рис. 20)

```
[svivanov@svivanov ~]$ sudo semanage port -a -p tcp -t http_port_t 81
Port tcp/81 already defined, modifying instead
[svivanov@svivanov ~]$ semanage port -l | grep http_port_t
ValueError: SELinux policy is not managed or store cannot be accessed.
[svivanov@svivanov ~]$ sudo semanage port -l | grep http_port_t
http_port_t                tcp      81, 80, 81, 443, 488, 8008, 8009, 8443,
9000
pegasus_http_port_t        tcp      5988
[svivanov@svivanov ~]$
```

Рис. 2.20: Добавление и проверка портов

Попробуем запустить веб-сервер Apache ещё раз. Вернем контекст `httpd_sys_content_t`
к файлу `/var/www/html/test.html`: `chcon -t httpd_sys_content_t /var/www/html/test.html`.
(рис. 21)

```
[svivanov@svivanov ~]$ sudo chcon -t httpd_sys_content_t /var/www/html/test.html
[svivanov@svivanov ~]$ sudo systemctl restart httpd
[svivanov@svivanov ~]$
```

Рис. 2.21: Перезапуск сервера

Сервер запустился, т.к порт 81 теперь прослушивается. (рис. 22)

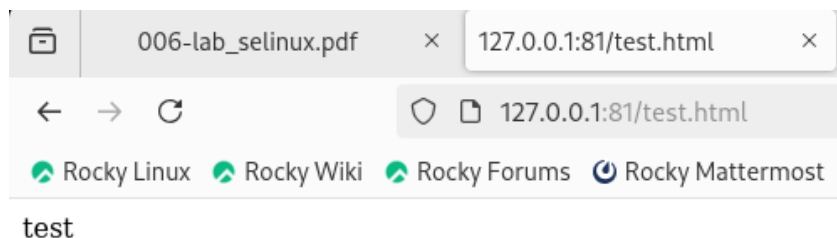


Рис. 2.22: Проверка сервера

Исправим обратно конфигурационный файл apache, вернув Listen 80. Удалим привязку http_port_t к 81 порту: semanage port -d -t http_port_t -p tcp 81 и проверим, что порт 81 удалён. (рис. 23)

```
[svivanov@svivanov ~]$ sudo nano /etc/httpd/conf/httpd.conf
[svivanov@svivanov ~]$ emanage port -d -t http_port_t -p tcp 81
bash: emanage: command not found...
[svivanov@svivanov ~]$ semanage port -d -t http_port_t -p tcp 81
ValueError: SELinux policy is not managed or store cannot be accessed.
[svivanov@svivanov ~]$ sudo semanage port -d -t http_port_t -p tcp 81
[svivanov@svivanov ~]$ semanage port -l | grep http_port_t
ValueError: SELinux policy is not managed or store cannot be accessed.
[svivanov@svivanov ~]$ 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
```

Рис. 2.23: Удаление порта

Удалим файл /var/www/html/test.html: rm /var/www/html/test.html. (рис. 24)

```
[svivanov@svivanov ~]$ sudo rm /var/www/html/test.html
```

Рис. 2.24: Удаление файла

3 Вывод

В ходе работы были развиты навыки администрирования ОС Linux. Получено первое практическое знакомство с технологией SELinux. Проверена работу SELinux на практике совместно с веб-сервером Apache.