

Отчет по лабораторной работе №6

Информационная безопасность

Чекалова Лилия Руслановна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
5	Выводы	16
	Список литературы	17

Список таблиц

Список иллюстраций

4.1	Проверка режима работы	8
4.2	Проверка работы веб-сервера	8
4.3	Запуск веб-сервера	9
4.4	Определение контекста безопасности	9
4.5	Просмотр переключателей	10
4.6	Статистика по политике	11
4.7	Определение типов поддиректорий	11
4.8	Создание файла test.html	12
4.9	Проверка контекста test.html	12
4.10	Просмотр файла в браузере	12
4.11	Смена контекста	12
4.12	Просмотр файла в браузере	13
4.13	Чтение лог-файлов	13
4.14	Смена порта	13
4.15	Перезапуск веб-сервера	14
4.16	Проверка лог-файлов	14
4.17	Просмотр списка портов	14
4.18	Смена контекста	15
4.19	Просмотр файла	15
4.20	Изменение порта	15
4.21	Попытка удаления порта	15

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 Задание

- Поиск информации про веб-сервер
- Работа с Html-файлами
- Просмотр лог-файлов

3 Теоретическое введение

SELinux представляет собой систему маркировки, каждый процесс файл, каталог, пользователь, устройство, порт и так далее имеет метку. SELinux определяет правила доступа процесса к объектам с определенными метками. Это называется политикой.

Владелец файла не имеет полной свободы действий над атрибутами безопасности. Стандартные атрибуты контроля доступа, такие как группа и владелец ничего не значат для SELinux. Полностью все управляется метками. Значения атрибутов могут быть установлены и без прав root, но на это нужно иметь специальные полномочия SELinux.

SELinux может работать в трех режимах — отключен, система полностью отключена и не работает, режим ограничений Enforcing — программа активирована и блокирует все не соответствующие политикам действия и третий режим Permissive — только фиксировать нарушения.

Политики SELinux бывают тоже нескольких типов. Политика targeted относится к типу Type Enforcement (TE) политик, в которых управление доступом к файлам осуществляется на основе ролей. Сюда же относится политика strict. Есть ещё политики Multi-Level Security (MLS), в которых добавлены дополнительные категории.

Более подробно о см. в [1,2].

4 Выполнение лабораторной работы

В качестве первого шага лабораторной работы мы проверили режим работы SELinux с помощью команд `getenforce` и `sestatus` (рис. 4.1).

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

Рис. 4.1: Проверка режима работы

Далее мы проверили, работает ли веб-сервер (рис. 4.2), и запустили его, так как он не работал (рис. 4.3).

```
[lrchekalova@lrchekalova ~]$ service httpd status
Redirecting to /bin/systemctl status httpd.service
o httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: d
   Active: inactive (dead)
   Docs: man:httpd.service(8)
lines 1-4/4 (END)
```

Рис. 4.2: Проверка работы веб-сервера


```

[lrchekalova@lrchekalova ~]$ service httpd start
Redirecting to /bin/systemctl start httpd.service
[lrchekalova@lrchekalova ~]$ service httpd status
Redirecting to /bin/systemctl status httpd.service
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
   Active: active (running) since Thu 2023-10-12 13:30:27 MSK; 15s ago
     Docs: man:httpd.service(8)
   Main PID: 3985 (httpd)
   Status: "Total requests: 0; Idle/Busy workers 100/0; Requests/sec: 0; Bytes served/sec: 0 B/sec"
    Tasks: 213 (limit: 12143)
   Memory: 29.4M
     CPU: 180ms
   CGroup: /system.slice/httpd.service
           └─3985 /usr/sbin/httpd -DFOREGROUND
             └─3993 /usr/sbin/httpd -DFOREGROUND
               └─3995 /usr/sbin/httpd -DFOREGROUND
                 └─3999 /usr/sbin/httpd -DFOREGROUND
                   └─4000 /usr/sbin/httpd -DFOREGROUND
[lrchekalova@lrchekalova ~]$

```

Рис. 4.3: Запуск веб-сервера

Определили контекст безопасности процесса веб-сервера (рис. 4.4). Главной информацией для нас стал тип процесса — `httpd_t`.

```

[lrchekalova@lrchekalova ~]$ ps auxZ | grep httpd
system_u:system_r:httpd_t:s0 root      3985  0.0  0.5 20328 11720 ?        Ss   13:30
0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache  3993  0.0  0.3 21664  7600 ?        S    13:30
0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache  3995  0.0  0.7 1210612 15280 ?        Sl   13:30
0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache  3999  0.0  0.6 1079476 13228 ?        Sl   13:30
0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache  4000  0.0  0.6 1079476 13232 ?        Sl   13:30
0:00 /usr/sbin/httpd -DFOREGROUND
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 lrcheka+ 4258 0.0  0.1 221688 2420 pts/0
S+  13:32  0:00 grep --color=auto httpd
[lrchekalova@lrchekalova ~]$

```

Рис. 4.4: Определение контекста безопасности

Посмотрели текущее положение переключателей SELinux, большинство из них находятся в выключенном состоянии (рис. 4.5).

```
[lrchekalova@lrchekalova ~]$ 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. 4.5: Просмотр переключателей

Посмотрели статистику по политике с помощью `seinfo` (рис. 4.6). Определили, что множество пользователей имеет размер 8, множество ролей — 14, а множество типов — 5100.

```

[lrchekalova@lrchekalova ~]$ 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:        5100    Attributes:      258
Users:        8       Roles:           14
Booleans:     353     Cond. Expr.:    384
Allow:        65008   Neverallow:     0
Auditallow:   170     Dontaudit:      8572
Type_trans:   265344  Type_change:    87
Type_member:  35      Range_trans:    6164
Role allow:   38      Role_trans:     420
Constraints:  70      Validatetrans:  0
MLS Constrain: 72     MLS Val. Tran:  0
Permissives:  2      Polcap:         6
Defaults:     7      Typebounds:     0
Allowxperm:   0      Neverallowxperm: 0
Auditallowxperm: 0    Dontauditxperm: 0
Ibendportcon: 0      Ibkeycon:       0
Initial SIDs: 27     Fs_use:         35
Genfscon:     109    Portcon:        660
Netifcon:     0      Nodecon:        0

[lrchekalova@lrchekalova ~]$ seinfo -t

Types: 5100
NetworkManager_dispatcher_chronyc_script_t

```

Рис. 4.6: Статистика по политике

Определили тип файлов и поддиректорий директории /var/www (рис. 4.7). Поддиректория cgi-bin имеет тип httpd_sys_script_exec_t, а html — httpd_sys_content_t. Только пользователь-владелец имеет право создавать файлы в папке html.

```

[lrchekalova@lrchekalova ~]$ ls -lZ /var/www
итого 0
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_script_exec_t:s0 6 мая 16 23:21 cgi-bin
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_content_t:s0      6 мая 16 23:21 html
[lrchekalova@lrchekalova ~]$ ls -lZ /var/www/html
итого 0
[lrchekalova@lrchekalova ~]$

```

Рис. 4.7: Определение типов поддиректорий

Создали файл test.html в папке html от лица суперпользователя (рис. 4.8).

```
[lrchekalova@lrchekalova ~]$ su
Пароль:
[root@lrchekalova lrchekalova]# touch /var/www/html/test.html
[root@lrchekalova lrchekalova]# echo '<html>' > /var/www/html/test.html
[root@lrchekalova lrchekalova]# echo '<body>test</body>' >> /var/www/html/test.html
[root@lrchekalova lrchekalova]# echo '</html>' >> /var/www/html/test.html
[root@lrchekalova lrchekalova]#
```

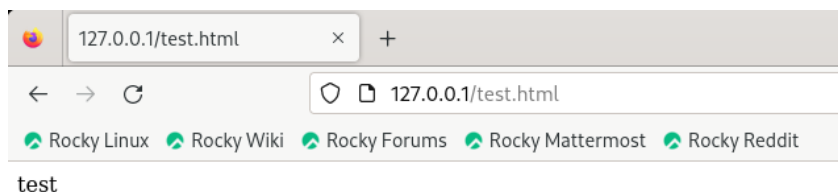
Рис. 4.8: Создание файла test.html

Проверили его контекст (рис. 4.9). Вновь созданным файлам в папке html по умолчанию присваивается тип `httpd_sys_content_t`.

```
[root@lrchekalova lrchekalova]# ls -lZ /var/www/html/test.html
-rw-r--r--. 1 root root unconfined_u:object_r:httpd_sys_content_t:s0 33 окт 12 13:49 /var/www/html/test.html
[root@lrchekalova lrchekalova]#
```

Рис. 4.9: Проверка контекста test.html

Обратились к файлу через веб-сервер и увидели его содержимое (рис. 4.10).



127.0.0.1/test.html

test

Рис. 4.10: Просмотр файла в браузере

Снова проверили контекст файла и поменяли его на другой (рис. 4.11). Новый контекст файла не позволяет процессу `httpd` получить доступ к файлу при обращении к нему через браузер.

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

Рис. 4.11: Смена контекста

Попробовали открыть файл в браузере (рис. 4.12). Возникла ошибка из-за нового контекста.

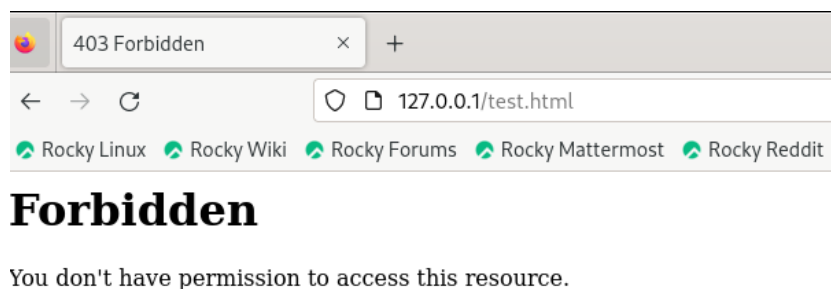


Рис. 4.12: Просмотр файла в браузере

Посмотрели лог-файлы (рис. 4.13). Увидели запись о неудачной попытке браузера получить доступ к файлу (ошибка 403).

```
[root@lrchekalova lrchekalova]# tail /etc/httpd/logs/access_log
127.0.0.1 - - [12/Oct/2023:13:54:50 +0300] "GET /test.html HTTP/1.1" 200 33 "-" "Mozilla/5.0
(X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0"
127.0.0.1 - - [12/Oct/2023:13:54:51 +0300] "GET /favicon.ico HTTP/1.1" 404 196 "http://127.0.
0.1/test.html" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0"
127.0.0.1 - - [12/Oct/2023:14:03:05 +0300] "GET /test.html HTTP/1.1" 403 199 "-" "Mozilla/5.0
(X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0"
```

Рис. 4.13: Чтение лог-файлов

Изменили порт в конфигурационном файле httpd.conf с 80 на 81 (рис. 4.14).

```
34 ServerRoot "/etc/httpd"
35
36 #
37 # Listen: Allows you to bind Apache to specific IP addresses and/or
38 # ports, instead of the default. See also the <VirtualHost>
39 # directive.
40 #
41 # Change this to Listen on a specific IP address, but note that if
42 # httpd.service is enabled to run at boot time, the address may not be
43 # available when the service starts. See the httpd.service(8) man
44 # page for more information.
45 #
46 #Listen 12.34.56.78:80
47 Listen 81
48
```

Рис. 4.14: Смена порта

Перезапустили веб-сервер, получили сообщение о том, что он запущен на прослушивание 81 порта (рис. 4.15).

```

[root@lrchekalova conf]# service httpd restart
Redirecting to /bin/systemctl restart httpd.service
[root@lrchekalova conf]# cd
[root@lrchekalova ~]# service httpd status
Redirecting to /bin/systemctl status httpd.service
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
   Active: active (running) since Thu 2023-10-12 14:27:19 MSK; 2min 9s ago
     Docs: man:httpd.service(8)
   Main PID: 6491 (httpd)
    Status: "Total requests: 0; Idle/Busy workers 100/0; Requests/sec: 0; Bytes served/sec: 0"
   Tasks: 213 (limit: 12143)
  Memory: 34.7M
    CPU: 260ms
   CGroup: /system.slice/httpd.service
           └─6491 /usr/sbin/httpd -DFOREGROUND
             └─6492 /usr/sbin/httpd -DFOREGROUND
               └─6496 /usr/sbin/httpd -DFOREGROUND
                 └─6497 /usr/sbin/httpd -DFOREGROUND
                   └─6499 /usr/sbin/httpd -DFOREGROUND

окт 12 14:27:19 lrchekalova.localdomain systemd[1]: Starting The Apache HTTP Server...
окт 12 14:27:19 lrchekalova.localdomain systemd[1]: Started The Apache HTTP Server.
окт 12 14:27:19 lrchekalova.localdomain httpd[6491]: Server configured, listening on: port 81

```

Рис. 4.15: Перезапуск веб-сервера

Проверили лог-файлы и нашли информацию о переключении веб-сервера на прослушивание нового порта (рис. 4.16).

```

[root@lrchekalova ~]# tail -n1 /var/log/messages
Oct 12 14:29:46 lrchekalova httpd[6750]: Server configured, listening on: port 81
[root@lrchekalova ~]# tail -n1 /var/log/httpd/error_log
tail: невозможно открыть '/var/log/httpd/error_log' для чтения: Нет такого файла или каталога
[root@lrchekalova ~]# tail -n1 /etc/httpd/logs/error_log
[Thu Oct 12 14:29:46.934962 2023] [core:notice] [pid 6750:tid 6750] AH00094: Command line: '/usr/sbin/httpd -D FOREGROUND'
[root@lrchekalova ~]# tail -n1 /etc/httpd/logs/access_log
127.0.0.1 - - [12/Oct/2023:14:03:05 +0300] "GET /test.html HTTP/1.1" 403 199 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0"
[root@lrchekalova ~]# tail -n1 /var/log/audit/audit.log
type=SERVICE_START msg=audit(1697110186.904:322): pid=1 uid=0 auid=4294967295 ses=4294967295 subj=system_u:system_r:init_t:s0 msg='unit=httpd comm="systemd" exe="/usr/lib/systemd/systemd" hostname=? addr=? terminal=? res=success'UID="root" AUID="unset"
[root@lrchekalova ~]#

```

Рис. 4.16: Проверка лог-файлов

Посмотрели список портов веб-сервера, нашли там указанный нами порт (рис. 4.17).

```

[root@lrchekalova ~]# semanage port -a -t http_port_t -p tcp 81
ValueError: Порт tcp/81 уже определен
[root@lrchekalova ~]# 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
[root@lrchekalova ~]#

```

Рис. 4.17: Просмотр списка портов

Вернули файлу test.html старый контекст (рис. 4.18).

```
[root@lrchekalova ~]# chcon -t httpd_sys_content_t /var/www/html/test.html
[root@lrchekalova ~]#
```

Рис. 4.18: Смена контекста

Получили доступ к файлу через веб-сервер в браузере (рис. 4.19).

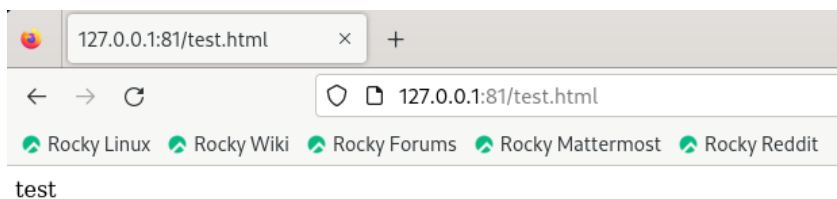


Рис. 4.19: Просмотр файла

Вернули порт 80 в конфигурационном файле (рис. 4.20).

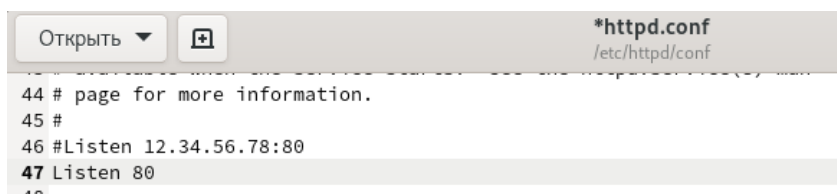


Рис. 4.20: Изменение порта

Попытались удалить 81 порт, но столкнулись с ошибкой, что он определен на уровне политики и не может быть удален (рис. 4.21). После этого удалили файл test.html.

```
[root@lrchekalova ~]# semanage port -d -t http_port_t -p tcp 81
ValueError: Порт tcp/81 определен на уровне политики и не может быть удален
[root@lrchekalova ~]# 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
[root@lrchekalova ~]# rm /var/www/html/test.html
rm: удалить обычный файл '/var/www/html/test.html'? y
[root@lrchekalova ~]#
```

Рис. 4.21: Попытка удаления порта

5 Выводы

В результате лабораторной работы я получила базовые навыки администрирования ОС Linux, познакомилась с технологией SELinux и проверила ее работу на практике совместно с веб-сервером Apache.

Список литературы

1. Мандатное разграничение прав в Linux [Электронный ресурс]. URL: https://esystem.rudn.ru/pluginfile.php/2090282/mod_resource/content/2/006-lab_selinux.pdf.
2. Настройка SELinux [Электронный ресурс]. URL: <https://losst.pro/nastrojka-selinux>.