

Отчёт по лабораторной работе №6, Основы информационной безопасности

Мандатное разграничение прав в Linux

Ганина Таисия Сергеевна, НКАбд-01-22

Содержание

1	Цель работы	4
2	Теоретическое введение	5
3	Выполнение лабораторной работы	7
4	Вывод	14
5	Список литературы. Библиография	15

Список иллюстраций

3.1	(Проверка режима enforcing политики targeted)	7
3.2	(Проверка работы веб-сервера)	8
3.3	(Контекст безопасности веб-сервера Apache)	8
3.4	(Текущее состояние переключателей SELinux)	9
3.5	(Статистика по политике)	9
3.6	(Просмотр файлов и поддиректорий в директории /var/www)	10
3.7	(Создание файла /var/www/html/test.html)	10
3.8	(Обращение к файлу через веб-сервер)	10
3.9	(Изменение контекста)	11
3.10	(Обращение к файлу через веб-сервер)	11
3.11	(Просмотр log-файла)	12
3.12	(Обращение к файлу через веб-сервер)	13
3.13	(Возвращение Listen 80 и попытка удалить порт 81)	13

1 Цель работы

Развить навыки администрирования ОС Linux. Получить первое практическое знакомство с технологией SELinux¹. Проверить работу SELinx на практике совместно с веб-сервером Apache.

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 [1].

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:

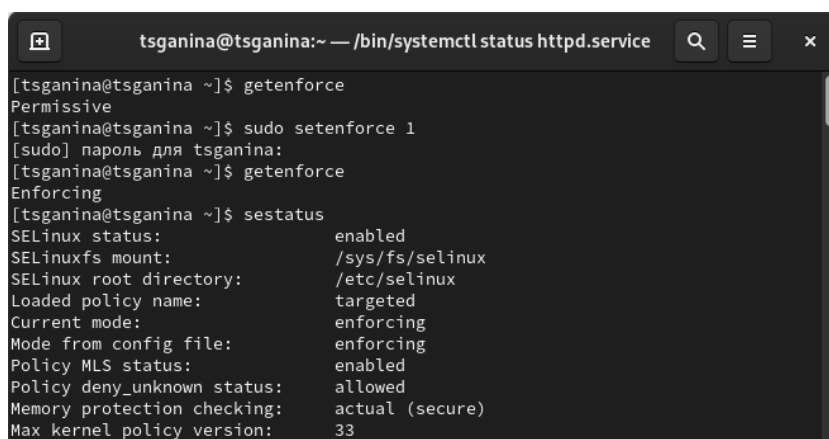
- чтобы открывать динамические PHP-страницы,
- для распределения поступающей на сервер нагрузки,
- для обеспечения отказоустойчивости сервера,
- чтобы потренироваться в настройке сервера и запуске PHP-скриптов.

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

Более подробно см. в [2].

3 Выполнение лабораторной работы

Вошли в систему под своей учетной записью и убедились, что SELinux работает в режиме enforcing политики targeted с помощью команд “getenforce” и “sestatus” (3.1)



```
tsganina@tsganina:~ — /bin/systemctl status httpd.service
[tsganina@tsganina ~]$ getenforce
Permissive
[tsganina@tsganina ~]$ sudo setenforce 1
[sudo] пароль для tsganina:
[tsganina@tsganina ~]$ getenforce
Enforcing
[tsganina@tsganina ~]$ 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. 3.1: (Проверка режима enforcing политики targeted)

Обратились с помощью браузера к веб-серверу, запущенному на компьютере, и убедились, что последний работает с помощью команды “service httpd status” (3.2)

```
tsganina@tsganina:~ — /bin/systemctl status httpd.service
[tsganina@tsganina ~]$ service httpd start
Redirecting to /bin/systemctl start httpd.service
[tsganina@tsganina ~]$ 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 Sat 2024-04-27 20:33:32 MSK; 42s ago
     Docs: man:httpd.service(8)
   Main PID: 100666 (httpd)
   Status: "Total requests: 0; Idle/Busy workers 100/0; Requests/sec: 0; Bytes served: 0"
    Tasks: 213 (limit: 12109)
   Memory: 50.0M
     CPU: 341ms
   CGroup: /system.slice/httpd.service
           └─100666 /usr/sbin/httpd -DFOREGROUND
             └─100677 /usr/sbin/httpd -DFOREGROUND
               └─100678 /usr/sbin/httpd -DFOREGROUND
                 └─100679 /usr/sbin/httpd -DFOREGROUND
                   └─100680 /usr/sbin/httpd -DFOREGROUND

anp 27 20:33:32 tsganina.localdomain systemd[1]: Starting The Apache HTTP Server:
anp 27 20:33:32 tsganina.localdomain httpd[100666]: Server configured, listening
anp 27 20:33:32 tsganina.localdomain systemd[1]: Started The Apache HTTP Server.
lines 1-19/19 (END)
```

Рис. 3.2: (Проверка работы веб-сервера)

С помощью команды “ps auxZ | grep httpd” определили контекст безопасности веб-сервера Apache - httpd_t (3.3)

```
[tsganina@tsganina ~]$ ps auxZ | grep httpd
system_u:system_r:httpd_t:s0 root 100666 0.0 0.5 20340 11760 ?
Ss 20:33 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 100677 0.0 0.3 21676 7632 ?
S 20:33 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 100678 0.0 0.9 2521344 19312 ?
Sl 20:33 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 100679 0.0 1.0 2259136 21400 ?
Sl 20:33 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 100680 0.0 1.0 2324672 21376 ?
Sl 20:33 0:00 /usr/sbin/httpd -DFOREGROUND
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 tsganina 100957 0.0 0.1 21688 2460 pts/1 S+ 20:37 0:00 grep --color=auto httpd
[tsganina@tsganina ~]$
```

Рис. 3.3: (Контекст безопасности веб-сервера Apache)

Посмотрели текущее состояние переключателей SELinux для Apache с помощью команды “sestatus -b httpd”, многие из переключателей находятся в положении “off” (3.4)


```
tsganina@tsganina:~$ sestatus -b httpd
Without options, show SELinux status.
[tsganina@tsganina ~]$ 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
```

Рис. 3.4: (Текущее состояние переключателей SELinux)

Посмотрели статистику по политике с помощью команды “seinfo”. Множество пользователей - 8, ролей - 15, типов 5135 (3.5)

```
tsganina@tsganina:~$ seinfo
Policy Version:                33 (MLS enabled)
Target Policy:                 selinux
Handle unknown classes:        allow
Classes:                       135
Sensitivities:                 1
Types:                         5135
Users:                         8
Booleans:                      357
Allow:                         65380
Auditallow:                    172
Type_trans:                    267809
Type_member:                   37
Role_allow:                    39
Constraints:                   70
MLS Constrain:                 72
Permissives:                   2
Defaults:                      7
Allowxperm:                    0
Auditallowxperm:               0
Ibendportcon:                  0
Initial SIDs:                  27
Genfscon:                      109
Netifcon:                      0
Permissions:                   457
Categories:                   1024
Attributes:                    259
Roles:                         15
Cond. Expr.:                   390
Neverallow:                    0
Dontaudit:                     8647
Type_change:                   94
Range_trans:                   6164
Role_trans:                    419
Validatetrans:                 0
MLS Val. Tran:                 0
Polcap:                        6
Typebounds:                    0
Neverallowxperm:               0
Dontauditxperm:               0
Ibpkeycon:                     0
Fs_use:                        35
Portcon:                       665
Nodecon:                       0
```

Рис. 3.5: (Статистика по политике)

С помощью команды “ls -lZ /var/www” посмотрели файлы и поддиректории, находящиеся в директории /var/www. Используя команду “ls -lZ /var/www/html”,

определили, что в данной директории файлов нет. Только владелец/суперпользователь может создавать файлы в директории /var/www/html (3.6)

```
[tsganina@tsganina ~]$ ls -lZ /var/www
итого 0
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_script_exec_t:s0 6 окт 28 12
:35 cgi-bin
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_content_t:s0 6 окт 28 12
:35 html
[tsganina@tsganina ~]$ ls -lZ /var/www/html
итого 0
[tsganina@tsganina ~]$
```

Рис. 3.6: (Просмотр файлов и поддиректорий в директории /var/www)

От имени суперпользователя создали html-файл /var/www/html/test.html. Текст созданного файла - httpd_sys_content_t (3.7)

```
[root@tsganina tsganina]# cat /var/www/html/test.html
<html>
<body>test</body>
</html>
[root@tsganina tsganina]# exit
exit
[tsganina@tsganina ~]$ ls -lZ /var/www/html
итого 4
-rw-r--r--. 1 root root unconfined_u:object_r:httpd_sys_content_t:s0 33 апр 27 2
0:44 test.html
[tsganina@tsganina ~]$
```

Рис. 3.7: (Создание файла /var/www/html/test.html)

Обратились к файлу через веб-сервер, введя в браузере адрес “http://127.0.0.1/test.html”. Файл был успешно отображен (3.8)

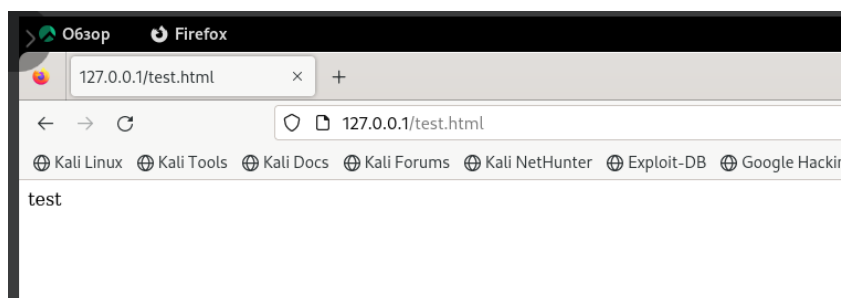


Рис. 3.8: (Обращение к файлу через веб-сервер)

Изучив справку man httpd_selinux, выяснили, 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). Изменили контекст файла на samba_share_t командой “sudo chcon -t samba_share_t/var/www/html/test.html” и проверили, что контекст поменялся (3.9)

```
[tsganina@tsganina ~]$ man httpd_selinux
Нет справочной страницы для httpd_selinux
[tsganina@tsganina ~]$ man httpd
[tsganina@tsganina ~]$ chcon -t samba_share_t /var/www/html/test.html
chcon: не удалось изменить контекст безопасности '/var/www/html/test.html' на «unconfined_u:object_r:samba_share_t:s0»: Операция не позволена
[tsganina@tsganina ~]$ su
Пароль:
[root@tsganina tsganina]# chcon -t samba_share_t /var/www/html/test.html
[root@tsganina tsganina]# ls -lZ /var/www/html/test.html
-rw-r--r--. 1 root root unconfined_u:object_r:samba_share_t:s0 33 anp 27 20:44 /var/www/html/test.html
[root@tsganina tsganina]#
```

Рис. 3.9: (Изменение контекста)

Попробовали еще раз получить доступ к файлу через веб-сервер, введя в браузере адрес “http://127.0.0.1/test.html” и получили сообщение об ошибке (т.к. к установленному ранее контексту процесс httpd не имеет доступа) (3.10)

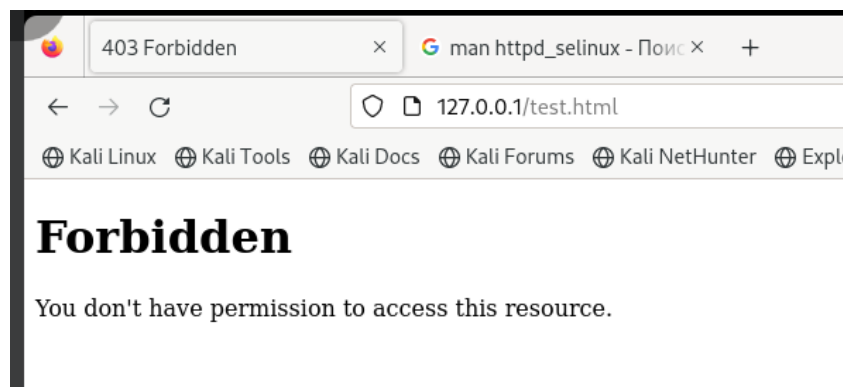
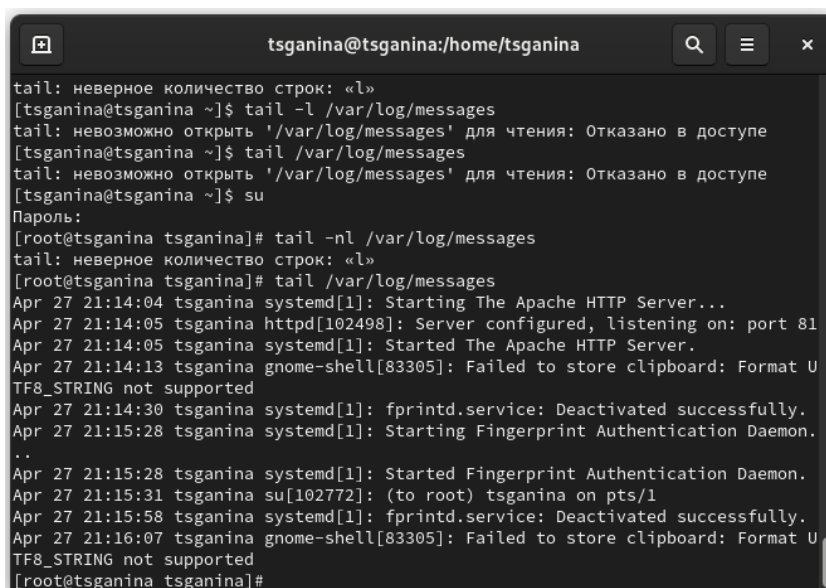


Рис. 3.10: (Обращение к файлу через веб-сервер)

Просмотрели системный лог-файл веб-сервера Apache командой “sudo tail

/var/log/messages”, отображающий ошибки (3.11)



```
tsganina@tsganina:/home/tsganina
tail: неверное количество строк: «1»
[tsganina@tsganina ~]$ tail -l /var/log/messages
tail: невозможно открыть '/var/log/messages' для чтения: Отказано в доступе
[tsganina@tsganina ~]$ tail /var/log/messages
tail: невозможно открыть '/var/log/messages' для чтения: Отказано в доступе
[tsganina@tsganina ~]$ su
Пароль:
[root@tsganina tsganina]# tail -nl /var/log/messages
tail: неверное количество строк: «1»
[root@tsganina tsganina]# tail /var/log/messages
Apr 27 21:14:04 tsganina systemd[1]: Starting The Apache HTTP Server...
Apr 27 21:14:05 tsganina httpd[102498]: Server configured, listening on: port 81
Apr 27 21:14:05 tsganina systemd[1]: Started The Apache HTTP Server.
Apr 27 21:14:13 tsganina gnome-shell[83305]: Failed to store clipboard: Format U
TF8_STRING not supported
Apr 27 21:14:30 tsganina systemd[1]: fprintd.service: Deactivated successfully.
Apr 27 21:15:28 tsganina systemd[1]: Starting Fingerprint Authentication Daemon.
..
Apr 27 21:15:28 tsganina systemd[1]: Started Fingerprint Authentication Daemon.
Apr 27 21:15:31 tsganina su[102772]: (to root) tsganina on pts/1
Apr 27 21:15:58 tsganina systemd[1]: fprintd.service: Deactivated successfully.
Apr 27 21:16:07 tsganina gnome-shell[83305]: Failed to store clipboard: Format U
TF8_STRING not supported
[root@tsganina tsganina]#
```

Рис. 3.11: (Просмотр log-файла)

В файле /etc/httpd/conf/httpd.conf заменили строчку “Listen 80” на “Listen 81”, чтобы установить веб-сервер Apache на прослушивание TCP-порта 81

Перезапускаем веб-сервер Apache и анализируем лог-файлы командой “tail -nl /var/log/messages”

Просмотрели файлы “var/log/http/error_log”, “/var/log/http/access_log” и “/var/log/audit/audit.log” и выяснили, что запись появилась в последнем файле

Выполнили команду “semanage port -a -t http_port_t -p tcp 81” и убедились, что порт TCP-81 установлен. Проверили список портов командой “semanage port -l | grep http_port_t”, убедились, что порт 81 есть в списке и запускаем веб-сервер Apache снова.

Вернули 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 попробовали получить доступ к файлу через веб-сервер, введя адрес “http://127.0.0.1:81/test.html”, в результате чего увидели содержимое файла - слово “test” (3.12)

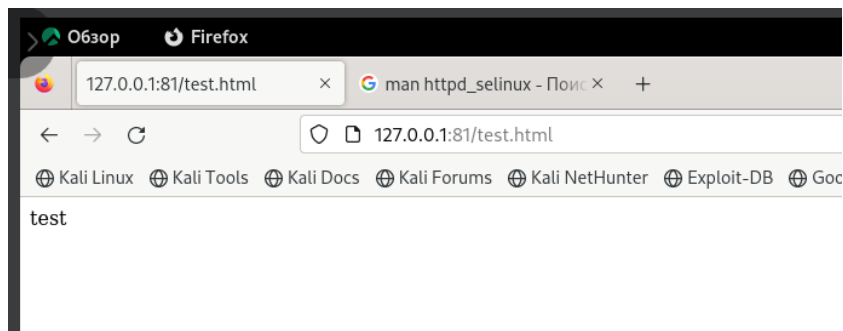


Рис. 3.12: (Обращение к файлу через веб-сервер)

Исправили обратно конфигурационный файл `apache`, вернув `Listen 80`. Попытались удалить привязку `http_port` к 81 порту командой `semanage port -d -t http_port_t -p tcp 81`, но этот порт определен на уровне политики, поэтому его нельзя удалить

Удалили файл `/var/www/html/test.html` командой `rm /var/www/html/test.html`
(3.13)

```
[tsganina@tsganina ~]$ semanage port -d -t http_port_t -p tcp 81
ValueError: Политика SELinux не задана, или нет доступа к хранилищу.
[tsganina@tsganina ~]$ sudo semanage port -d -t http_port_t -p tcp 81
ValueError: Port tcp/81 is defined in policy, cannot be deleted
[tsganina@tsganina ~]$ sudo rm /var/www/html/test.html
[tsganina@tsganina ~]$
```

Рис. 3.13: (Возвращение `Listen 80` и попытка удалить порт 81)

4 Вывод

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.

5 Список литературы. Библиография

[0] Методические материалы курса

[1] SELinux: <https://habr.com/ru/companies/kingservers/articles/209644/>

[2] Apache: <https://2domains.ru/support/vps-i-servery/shto-takoye-apache>