

Отчет по лабораторной работе №6

Основы информационной безопасности

Efe kantoz, НКАбд-01-23

Содержание

1	Цель работы	1
2	Теоретическое введение	1
3	Выполнение лабораторной работы	2
4	Выводы	10
	Список литературы.....	Error! Bookmark not defined.

1 Цель работы

Развить навыки администрирования ОС Linux. Получить первое практическое знакомство с технологией SELinux¹. Проверить работу SELinux на практике совместно с веб-сервером Apache. [1]

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 [2].

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 [3].

3 Выполнение лабораторной работы

Вошла в систему под своей учетной записью. Убедилась, что SELinux работает в режиме enforcing политики targeted с помощью команд `getenforce` и `sestatus` (рис. 1).

```
$ getenforce
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
tevdvorkina@tevdvorkina ~]$
```

Рис. 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. 2).

```

directing 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 2024-04-20 04:52:10 MSK; 31s ago
Docs: man:httpd.service(8)
Main PID: 30093 (httpd)
Status: "Total requests: 0; Idle/Busy workers 100/0; Requests/sec: 0; Bytes served/sec: 0"
Tasks: 213 (limit: 10899)
Memory: 37.9M
CPU: 301ms
CGroup: /system.slice/httpd.service
├─30093 /usr/sbin/httpd -DFOREGROUND
├─30133 /usr/sbin/httpd -DFOREGROUND
├─30134 /usr/sbin/httpd -DFOREGROUND
├─30135 /usr/sbin/httpd -DFOREGROUND
└─30136 /usr/sbin/httpd -DFOREGROUND

Apr 20 04:52:10 evdvorkina systemd[1]: Starting The Apache HTTP Server...
Apr 20 04:52:10 evdvorkina httpd[30093]: AH00558: httpd: Could not reliably determine the

```

Рис. 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).

```

system_u:system_r:httpd_t:s0 root 30093 0.1 0.6 20340 11624 ?
ss 04:52 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 30133 0.0 0.4 21676 7436 ?
S 04:52 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 30134 0.0 1.0 2193664 19320 ?
sl 04:52 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 30135 0.0 0.8 2062528 15228 ?
sl 04:52 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 30136 0.0 0.8 2062528 15228 ?
sl 04:52 0:00 /usr/sbin/httpd -DFOREGROUND
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 evdvork+ 42224 0.0 0.1 22
1688 2388 pts/0 S+ 04:53 0:00 grep --color=auto httpd

```

Рис. 3: Контекст безопасности Apache

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

```

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

```

Рис. 4: Состояние переключателей SELinux

Просмотрела статистику по политике с помощью команды `seinfo`. Множество пользователей - 8, ролей - 39, типов - 5135. (рис. 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:        5135     Attributes:       259
Users:        8       Roles:           15
Booleans:     357     Cond. Expr.:     390
Allow:        65409   Neverallow:      0
Auditallow:   172     Dontaudit:       8647
Type_trans:   267813  Type_change:     94
Type_member:  37      Range_trans:     6164
Role allow:   39      Role_trans:      419
Constraints:  70      Validatetrans:   0
MLS Constrai: 72     MLS Val. Tran:   0
Permissives:  2      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

[evdvorkina@evdvorkina ~]$
```

Рис. 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. 6).

```
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
```

Рис. 6: Типы поддиректорий

В директории `/var/www/html` нет файлов. (рис. 7).

```
$ ls -lZ /var/www/html
```

Рис. 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. 8).

```
sudo touch /var/www/html/test.html
```

Рис. 8: Создание файла

Проверяю контекст созданного файла. По умолчанию это httpd_sys_content_t (рис. 9).

```
$ ls -lZ /var/www/html/
unconfined_u:object_r:httpd_sys_content_t:s0 33 anp 20 05:01 test.html
$
```

Рис. 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. 10).

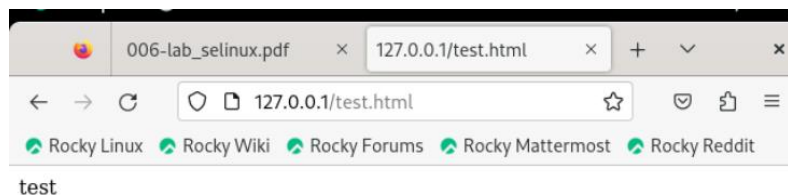


Рис. 10: Отображение файла

Изучила 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 `/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).

```
HTTPD(8)                                httpd

NAME
    httpd - Apache Hypertext Transfer Protocol Server

SYNOPSIS
    httpd [ -d serverroot ] [ -f config ] [ -C directive ] [ -c directive
    [ -e level ] [ -E file ] [ -k start|restart|graceful|stop|graceful-stop
    [ -L ] [ -S ] [ -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.
    be run as a standalone daemon process. When used like this it will c
    child processes or threads to handle requests.

    In general, httpd should not be invoked directly, but rather shou
    apachectl on Unix-based systems or as a service on Windows NT, 2000 an
    Manual page httpd(8) line 1 (press h for help or q to quit)
```

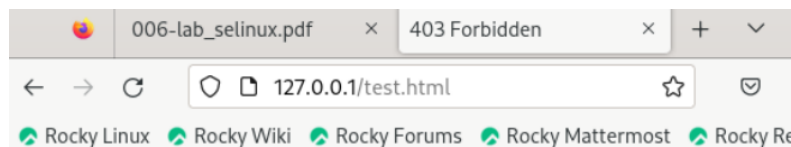
Рис. 11: Изучение справки по команде

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. 12).

```
sudo chcon -t samba_share_t /var/www/html/test.html
ls -lZ /var/www/html
confined_u:object_r:samba_share_t:s0 33 anp 20 05:01 test.html
```

Рис. 12: Изменение контекста

При попытке отображения файла в браузере получаем сообщение об ошибке (рис. 13).



Forbidden

You don't have permission to access this resource.

Рис. 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 `auditd`, то вы также сможете увидеть ошибки, аналогичные указанным выше, в файле `/var/log/audit/audit.log`. (рис. 14).

Рис. 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. 15).

```
sudo nano /etc/httpd/conf/httpd.conf
```

Рис. 15: Изменение файла

Нахожу строчку Listen 80 и заменяю её на Listen 81. (рис. 16).

```
# Do not add a slash at the end of the directory path. If you point
# ServerRoot at 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
# directives contained in it are actually available _before_ they are used.
# Statically compiled modules (those listed by 'httpd -l') do not need
# to be loaded here.
```

^G Справка	^O Записать	^W Поиск	^K Вырезать	^T Выполнить
^X Выход	^R ЧитФайл	^_ Замена	^U Вставить	^J Выводить

Рис. 16: Изменение порта

Выполняю перезапуск веб-сервера Apache. Произошёл сбой, потому что порт 80 для локальной сети, а 81 нет (рис. 17).

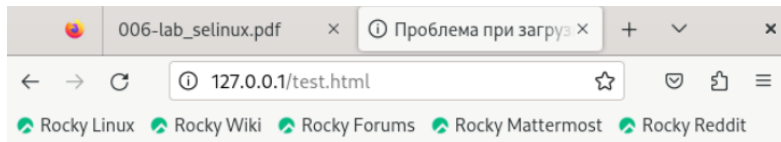


Рис. 17: Попытка прослушивания другого порта

Проанализируйте лог-файлы: `tail -n1 /var/log/messages` (рис. 18).

```
sudo tail -n1 /var/log/messages
systemd[1]: Started The Apache HTTP Server.
```

Рис. 18: Проверка лог-файлов

Просмотрите файлы `/var/log/http/error_log`, `/var/log/http/access_log` и `/var/log/audit/audit.log` и выясните, в каких файлах появились записи. Запись появилась в файлу `error_log` (рис. 19).


```
[Sat Apr 20 04:52:10.304359 2024] [core:notice] [pid 30093:tid 30093] SELinux
policy enabled; httpd running as context system_u:system_r:httpd_t:s0
[Sat Apr 20 04:52:10.307330 2024] [suexec:notice] [pid 30093:tid 30093] AH0123
2: suEXEC mechanism enabled (wrapper: /usr/sbin/suexec)
AH00558: httpd: Could not reliably determine the server's fully qualified domain
name, using fe80::a00:27ff:fe98:bdea%enp0s3. Set the 'ServerName' directive
globally to suppress this message
[Sat Apr 20 04:52:10.371973 2024] [lbmethod_heartbeat:notice] [pid 30093:tid 3
0093] AH02282: No slotmem from mod_heartbeat
[Sat Apr 20 04:52:10.389422 2024] [mpm_event:notice] [pid 30093:tid 30093] AH0
0489: Apache/2.4.57 (Rocky Linux) configured -- resuming normal operations
[Sat Apr 20 04:52:10.389524 2024] [core:notice] [pid 30093:tid 30093] AH00094:
Command line: '/usr/sbin/httpd -D FOREGROUND'
[Sat Apr 20 05:09:47.974451 2024] [core:error] [pid 30136:tid 30312] (13)Permi
ssion denied: [client 127.0.0.1:44098] AH00035: access to /test.html denied (f
ilesystem path '/var/www/html/test.html') because search permissions are missi
ng on a component of the path
[Sat Apr 20 05:15:41.743945 2024] [core:error] [pid 30134:tid 30322] (13)Permi
ssion denied: [client 127.0.0.1:58006] AH00035: access to /test.html denied (f
ilesystem path '/var/www/html/test.html') because search permissions are missi
ng on a component of the path
[Sat Apr 20 05:16:30.614988 2024] [mpm_event:notice] [pid 30093:tid 30093] AH0
0492: caught SIGWINCH, shutting down gracefully
[Sat Apr 20 05:16:31.052047 2024] [core:notice] [pid 42305:tid 42305] SELinux
```

Рис. 19: Проверка лог-файлов

Выполняю команду `semanage port -a -t http_port_t -p tcp 81` После этого проверяю список портов командой `semanage port -l | grep http_port_t` Порт 81 появился в списке (рис. 20).

```
$ sudo semanage port -a -t http_port_t -p tcp 81
```

Рис. 20: Проверка портов

Перезапускаю сервер Apache (рис. 21).

```
$ sudo systemctl restart httpd
```

Рис. 21: Перезапуск сервера

Теперь он работает, ведь мы внесли порт 81 в список портов `httpd_port_t` (рис. 22).

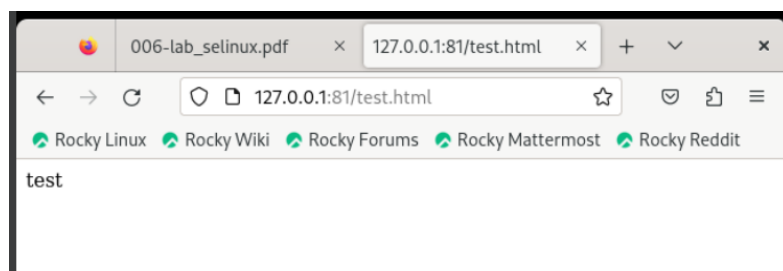


Рис. 22: Проверка сервера

Возвращаю в файле `/etc/httpd/httpd.conf` порт 80, вместо 81. Проверяю, что порт 81 удален, это правда. (рис. 23).

```
$ sudo nano /etc/httpd/conf/httpd.conf
$ semanage port -d -t http_port_t -p tcp 81
nix не задана, или нет доступа к хранилищу.
$ sudo semanage port -d -t http_port_t -p tcp 81
s defined in policy, cannot be deleted
```

Рис. 23: Проверка порта 81

Далее удаляю файл test.html, проверяю, что он удален(рис. 24).



```
~]$ ls -lZ /var/www/html
```

Рис. 24: Удаление файла

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.