

Лабораторная работа № 6

Мандатное разграничение прав в Linux

Алескеров Тимур Магомедович НБИБД-01-18

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	9
5	Выводы	16
	Список литературы	17

Список иллюстраций

4.1	getenforce и sestatus	9
4.2	service httpd status	10
4.3	ps -eZ grep httpd	10
4.4	sestatus -b grep httpd	11
4.5	seinfo	11
4.6	html-файл /var/www/html/test.html	12
4.7	Обратимся к файлу через веб-сервер	12
4.8	ls -Z /var/www/html/test.html	13
4.9	сообщение об ошибке	13
4.10	найдем строчку Listen 80 и заменим её на Listen 81.	14
4.11	Убедимся, что порт 81 появился в списке	15
4.12	Убедимся, что порт 81 появился в списке	15

Список таблиц

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

1. Запустить серверо
2. Настроить права дотупа к файлам
3. Поменять порт сервера

3 Теоретическое введение

Security Enhanced Linux может работать двумя различными способами:

Enforcing: SELinux запрещает доступ на основе правил политики SELinux, набора руководящих принципов, которые управляют механизмом безопасности.

Permissive: SELinux не запрещает доступ, но в журнале регистрируются отказы для действий, которые были бы запрещены при запуске в принудительном режиме. SELinux также можно отключить.

Хотя это не сам режим работы, это все же вариант.

Однако научиться использовать этот инструмент лучше, чем просто игнорировать его. Имейте это в виду!

Чтобы отобразить текущий режим SELinux, используйте `getenforce`.

Если вы хотите переключить режим работы, используйте `setenforce 0` (чтобы установить для него **Permissive**) или `setenforce 1` (**Enforcing**).

Поскольку это изменение не выдержит перезагрузки, вам потребуется отредактировать файл `/etc/selinux/config` и установить для переменной `SELINUX` значение `enforcing`, `permissive`, или `disabled`, чтобы обеспечить постоянство при перезагрузках:

Напомним, что если `Getenforce` возвращает `Disabled`, вам нужно отредактировать `/etc/selinux/config` с нужным режимом работы и перезагрузиться.

В противном случае вы не сможете установить (или переключить) режим работы с помощью `setenforce`.

Одно из типичных применений `setenforce` состоит в переключении между режимами SELinux (от принудительного к разрешающему или наоборот) для

устранения неполадок в работе приложения, которое работает неправильно или работает не так, как ожидалось.

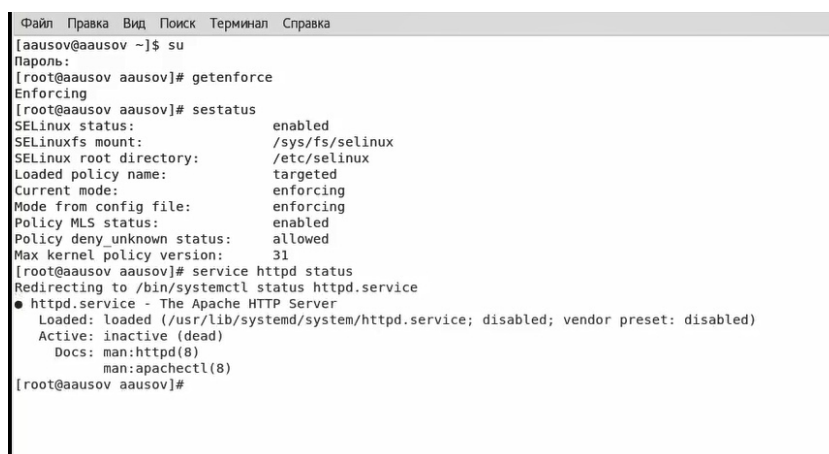
Если оно работает после того, как вы установили SELinux в режим Permissive, вы можете быть уверены, что надо рыться в проблеме с разрешениями SELinux.

Два классических случая, когда нам, скорее всего, придется иметь дело с SELinux:

Изменение порта по умолчанию, на котором слушает демон. Установка директивы DocumentRoot для виртуального хоста вне /var/www/html. [1]

4 Выполнение лабораторной работы

1. Предварительно настроил систему, установил необходимые утилиты.
2. Войдем в систему с полученными учётными данными и убедимся, что SELinux работает в режиме enforcing политики targeted с помощью команд `getenforce` и `sestatus` (рис. 4.1):



```
Файл  Правка  Вид  Поиск  Терминал  Справка
[aausov@aausov ~]$ su
Пароль:
[root@aausov aausov]# getenforce
Enforcing
[root@aausov aausov]# 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
Max kernel policy version:      31
[root@aausov aausov]# 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; vendor preset: disabled)
   Active: inactive (dead)
     Docs: man:httpd(8)
           man:apachectl(8)
[root@aausov aausov]#
```

Рис. 4.1: `getenforce` и `sestatus`

3. Обратимся с помощью браузера к веб-серверу, запущенному на компьютере, и убедимся, что последний работает: `service httpd status` (рис. 4.2).

```

Docs: man:htp(8)
man:apachectl(8)
[root@aasov aasov]# /etc/rc.d/init.d/httpd status
bash: /etc/rc.d/init.d/httpd: Нет такого файла или каталога
[root@aasov aasov]# service httpd start
Redirecting to /bin/systemctl start httpd.service
[root@aasov aasov]# 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; vendor preset: disabled)
   Active: active (running) since Бс 2021-11-21 15:26:51 MSK; 5s ago
     Docs: man:htp(8)
           man:apachectl(8)
   Main PID: 2810 (httpd)
   Status: "Processing requests..."
     Tasks: 6
    CGroup: /system.slice/httpd.service
            └─2810 /usr/sbin/httpd -DFOREGROUND
              └─2815 /usr/sbin/httpd -DFOREGROUND
                └─2816 /usr/sbin/httpd -DFOREGROUND
                  └─2817 /usr/sbin/httpd -DFOREGROUND
                    └─2818 /usr/sbin/httpd -DFOREGROUND
                      └─2819 /usr/sbin/httpd -DFOREGROUND
ноя 21 15:26:51 aasov.localdomain systemd[1]: Starting The Apache HTTP Server...
ноя 21 15:26:51 aasov.localdomain systemd[1]: Started The Apache HTTP Server.

```

Рис. 4.2: service httpd status

- Найдем веб-сервер Apache в списке процессов, определим его контекст безопасности. Например, можно использовать команду `ps auxZ | grep httpd` или `ps -eZ | grep httpd`

(рис. 4.3).

```

[root@aasov aasov]# ps -eZ | grep httpd
system_u:system_r:httpd_t:s0      2810 ?        00:00:00 httpd
system_u:system_r:httpd_t:s0      2815 ?        00:00:00 httpd
system_u:system_r:httpd_t:s0      2816 ?        00:00:00 httpd
system_u:system_r:httpd_t:s0      2817 ?        00:00:00 httpd
system_u:system_r:httpd_t:s0      2818 ?        00:00:00 httpd
system_u:system_r:httpd_t:s0      2819 ?        00:00:00 httpd

```

Рис. 4.3: ps -eZ | grep httpd

- Посмотрим текущее состояние переключателей SELinux для Apache с помощью команды `sestatus -b | grep httpd`, обращая внимание, что многие из них находятся в положении «off».

(рис. 4.4).

```

user_exec_content on
varnishd_connect_any off
virt_read_qemu_ga_data off
virt_rw_qemu_ga_data off
virt_sandbox_use_all_caps on
virt_sandbox_use_audit on
virt_sandbox_use_fusefs off
virt_sandbox_use_mknod off
virt_sandbox_use_netlink off
virt_sandbox_use_sys_admin off
virt_transition_userdomain off
virt_use_comm off
virt_use_execmem off
virt_use_fusefs off
virt_use_glusterd off
virt_use_nfs off
virt_use_rawip off
virt_use_samba off
virt_use_sanlock off
virt_use_usb on
virt_use_xserver off
webadm_manage_user_files off
webadm_read_user_files off
wine_mmap_zero_ignore off
xdm_bind_vnc_tcp_port off
xdm_exec_bootloader off
xdm_sysadm_login off
xdm_write_home off
xen_use_nfs off
xend_run_blkmap on
xend_run_qemu on
xguest_connect_network on
xguest_exec_content on
xguest_mount_media on
xguest_use_bluetooth on
xserver_clients_write_xshm off
xserver_execmem off
xserver_object_manager off
zabbix_can_network off
zabbix_run_sudo off
zarafe_setrlimit off
zebra_write_config off
zoneminder_anon_write off
zoneminder_run_sudo off
[root@aasov aasov]#

```

Рис. 4.4: sestatus -b | grep httpd

6. Посмотрим статистику по политике с помощью команды seinfo, также определим множество пользователей, ролей, типов. (рис. 4.5).

```

Выполнено!
[root@aasov aasov]# seinfo

Statistics for policy file: /sys/fs/selinux/policy
Policy Version & Type: v.31 (binary, mls)

Classes:          130      Permissions:       272
Sensitivities:    1       Categories:       1024
Types:            4793    Attributes:       253
Users:            8       Roles:            14
Booleans:         316     Cond. Expr.:     362
Allow:            107834   Neverallow:      0
Auditallow:       158     Dontaudit:       10022
Type_trans:       18153   Type_change:     74
Type_member:      35      Role_allow:      37
Role_trans:       414     Range_trans:     5899
Constraints:      143     Validatetrans:   0
Initial SIDs:     27      Fs_use:          32
Genfscon:         103     Portcon:         614
Netifcon:         0       Nodecon:         0
Permissives:      0       Polcap:          5

[root@aasov aasov]# ls -lZ /var/www
drwxr-xr-x. root root system_u:object_r:httpd_sys_script_exec_t:s0 cgi-bin
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 html

```

Рис. 4.5: seinfo

7. Определим тип файлов и поддиректорий, находящихся в директории /var/www, с помощью команды `ls -lZ /var/www` Определим тип файлов, находящихся в директории /var/www/html: `ls -lZ /var/www/html` (рис. 4.5)
8. Определим круг пользователей, которым разрешено создание файлов в директории /var/www/html. Создание файлов в директории разрешено только суперпользователю. Создадим от имени суперпользователя (так как в дистрибутиве после установки только ему разрешена запись в директорию) html-файл /var/www/html/test.html следующего содержания:
- test

(рис. 4.6)

```
drwxr-xr-x. root root system_u:object_r:httpd_sys_script_ex
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t
[root@aausov aausov]# s -lZ /var/www/html
bash: s: команда не найдена...
[root@aausov aausov]# ls -lZ /var/www/html
[root@aausov aausov]# mcedit /var/www/html/test.html

[root@aausov aausov]# cat /var/www/html/test.html
<html>
<body>test</body>
</html>[root@aausov aausov]#
```

Рис. 4.6: html-файл /var/www/html/test.html

9. Обратимся к файлу через веб-сервер. Убедимся, что файл был успешно отображён.(рис. 4.7)

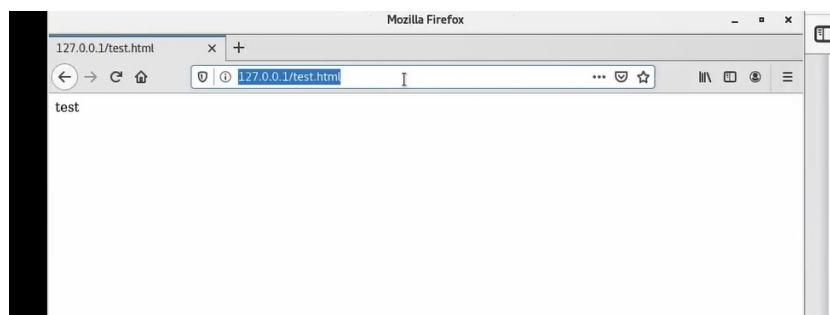


Рис. 4.7: Обратимся к файлу через веб-сервер

10. Изучим справку `man httpd_selinux` и выясним, какие контексты файлов определены для `httpd`. Сопоставим их с типом файла `test.html`. Проверить контекст файла можно командой `ls -Z`. `ls -Z /var/www/html/test.html` SELinux требует наличия у файлов расширенных атрибутов, определяющих тип файла. Политика управляет видом доступа демона к этим файлам. Политика SELinux для демона `httpd` позволяет пользователям настроить web-службы максимально безопасным методом с высокой степенью гибкости. рис. 4.8)

```
<body>test</body>
</html>[root@aasov aasov]# man httpd_selinux
Нет справочной страницы для httpd_selinux
[root@aasov aasov]# ls -Z ls -Z /var/www/html/test.html
ls: неверный ключ - «.»
По команде «ls --help» можно получить дополнительную информацию.
[root@aasov aasov]# ls -Z /var/www/html/test.html
-rw-r--r-- root root unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/test.html
[root@aasov aasov]#
```

Рис. 4.8: `ls -Z /var/www/html/test.html`

11. Попробуем ещё раз получить доступ к файлу через веб-сервер, введя в браузере адрес `http://127.0.0.1/test.html`. Однако, мы получаем сообщение об ошибке: (рис. 4.9)

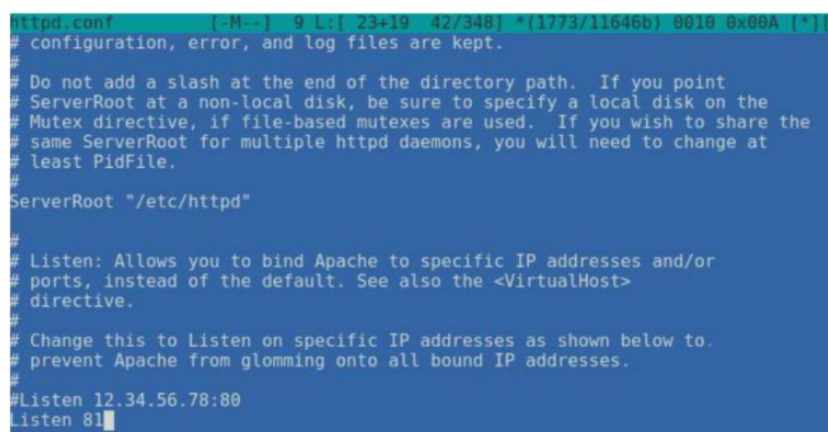


Рис. 4.9: сообщение об ошибке

12. Проанализируем ситуацию. Почему файл не был отображён, если права доступа позволяют читать этот файл любому пользователю? Из-за

смены контекста доступ к файлу из браузера запрещен. SELinux требует наличия у файлов расширенных атрибутов, определяющих тип файла. `ls -l /var/www/html/test.html` Просмотрим log-файлы веб-сервера Apache. Также посмотрим системный лог-файл: `tail /var/log/messages` Если в системе окажутся запущенными процессы `setroubleshootd` и `audtd`, то мы также сможем увидеть ошибки, аналогичные указанным выше, в файле `/var/log/audit/audit.log`.

13. Попробуем запустить веб-сервер Apache на прослушивание TCP-порта 81 (а не 80, как рекомендует IANA и прописано в `/etc/services`). Для этого в файле `/etc/httpd/httpd.conf` найдем строчку `Listen 80` и заменим её на `Listen 81`. (рис. 4.10)



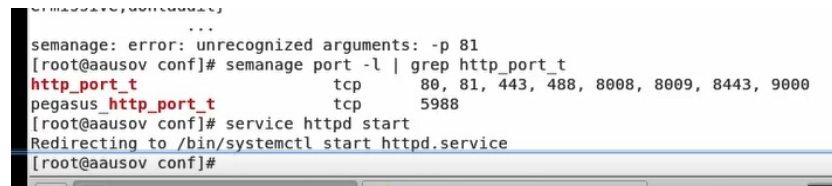
```
httpd.conf [-M--] 9 L: [ 23+19 42/348] *(1773/11646b) 0010 0x00A [*]]>
# configuration, error, and log files are kept.
#
# Do not add a slash at the end of the directory path. If you point
# ServerRoot at a non-local disk, be sure to specify a local disk on the
# Mutex directive, if file-based mutexes 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 specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses.
#
#Listen 12.34.56.78:80
Listen 81
```

Рис. 4.10: найдем строчку `Listen 80` и заменим её на `Listen 81`.

14. Выполним перезапуск веб-сервера Apache и проанализируем лог-файлы: `tail -nl /var/log/messages`. Просмотрим файлы и выясним, в каких файлах появились записи: `/var/log/http/error_log` `/var/log/http/access_log` `/var/log/audit/audit.log`

Перезапуск прошел успешно, т.к. 81 порт в данной системе прописан в политике. Ошибок не выскочило, только записи о перезапуске.

15. Выполним команду `semanage port -a -t http_port_t -p tcp 81` После этого проверим список портов командой `semanage port -l | grep http_port_t` Убедимся, что порт 81 появился в списке. Команда ни на что не повлияла, порт там уже был. В списке есть. (рис. 4.11)



```
semanage: error: unrecognized arguments: -p 81
[root@aasov conf]# 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@aasov conf]# service httpd start
Redirecting to /bin/systemctl start httpd.service
[root@aasov conf]#
```

Рис. 4.11: Убедимся, что порт 81 появился в списке

16. Попробуем запустить веб-сервер Apache ещё раз. Поняли ли вы, почему он сейчас запустился, а в предыдущем случае не смог? Успешно перезапустили. И тогда, и сейчас это было успешно, т.к. порт 81 уже был прописан.
17. Вернем контекст `httpd_sys_content_t` к файлу `/var/www/html/ test.html`: `chcon -t httpd_sys_content_t /var/www/html/test.html` После этого попробуем получить доступ к файлу через веб-сервер, введя в браузере адрес `http://127.0.0.1:81/test.html`. Мы видим содержимое файла – слово «test». (рис. 4.12)

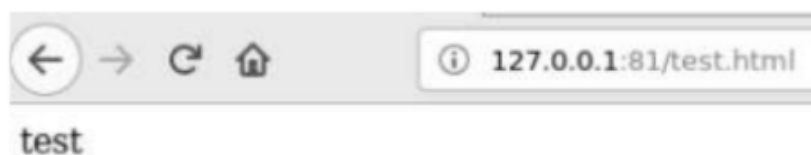


Рис. 4.12: Убедимся, что порт 81 появился в списке

18. Исправим обратно конфигурационный файл `apache`, вернув `Listen 80`.
19. Удалим привязку `http_port_t` к 81 порту: `semanage port -d -t http_port_t -p tcp 81`
20. Удалим файл `/var/www/html/test.html`: `rm /var/www/html/test.html`

5 Выводы

В ходе данной лабораторной работы мы развили навыки администрирования ОС Linux, впервые практически познакомились с технологией SELinux¹ и проверили работу SELinux на практике совместно с веб-сервером Apache.

Список литературы

1. suid [Электронный ресурс]. Сайт, 2021. URL: <https://itsecforu.ru/2019/07/25/%F0%9F%9B%A1%EF%B8%8F-%D1%80%D0%B5%D0%B0%D0%BB%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F-%D0%BC%D0%B0%D0%BD%D0%B4%D0%B0%D1%82%D0%BD%D0%BE%D0%B3%D0%BE-%D0%BA%D0%BE%D0%BD%D1%82%D1%80%D0%BE%D0%BB%D1%8F-%D0%B4/>.