

Отчёта по лабораторной работе № 6

Информационная безопасность

Адебайо Ридвануллахи Айофе

Содержание

1	Цель работы	5
2	Теорическое введение	6
3	Выполнение лабораторной работы	8
4	Выводы	17
5	Список литературы	18

Список иллюстраций

3.1	Проверка режима enforcing политики targeted	8
3.2	Проверка работы веб-сервера	9
3.3	Контекст безопасности веб-сервера Apache	9
3.4	Текущее состояние переключателей SELinux	10
3.5	Статистика по политике	11
3.6	Просмотр файлов и поддиректорий в директории /var/www . . .	11
3.7	Создание файла /var/www/html/test.html	12
3.8	Обращение к файлу через веб-сервер	12
3.9	Изменение контекста	12
3.10	Изменение контекста	13
3.11	Обращение к файлу через веб-сервер	13
3.12	Просмотр log-файла	14
3.13	Установка веб-сервера Apache на прослушивание TCP-порта 81 . .	14
3.14	Содержание файла var/log/audit/audit.log	15
3.15	Проверка установки порта 81	15
3.16	Возвращение исходного контекста файлу	16
3.17	Возвращение Listen 80 и попытка удалить порт 81	16

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 Теорическое введение

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.

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.

3 Выполнение лабораторной работы

Вошел в систему под своей учетной записью и убедился, что SELinux работает в режиме enforcing политики targeted с помощью команд “getenforce” и “sestatus”

```
[raadebayjo@raadebayjo ~]$ getenforce

Enforcing
[raadebayjo@raadebayjo ~]$
[raadebayjo@raadebayjo ~]$
[raadebayjo@raadebayjo ~]$
[raadebayjo@raadebayjo ~]$
[raadebayjo@raadebayjo ~]$ 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
[raadebayjo@raadebayjo ~]$ ss
```

Рис. 3.1: Проверка режима enforcing политики targeted

Обратился с помощью браузера к веб-серверу, запущенному на моем компьютере, и убедился, что последний работает с помощью команды “service httpd status”


```
[raadebayjo@raadebayjo ~]$ sudo systemctl start httpd
[sudo] password for raadebayjo:
[raadebayjo@raadebayjo ~]$ 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: d>
   Active: active (running) since Thu 2023-10-12 00:12:50 MSK; 12s ago
     Docs: man:httpd.service(8)
   Main PID: 33791 (httpd)
   Status: "Total requests: 0; Idle/Busy workers 100/0; Requests/sec: 0; Bytes>
     Tasks: 213 (limit: 18892)
    Memory: 41.5M
       CPU: 1.581s
    CGroup: /system.slice/httpd.service
            └─33791 /usr/sbin/httpd -DFOREGROUND
              └─33792 /usr/sbin/httpd -DFOREGROUND
                └─33793 /usr/sbin/httpd -DFOREGROUND
                  └─33794 /usr/sbin/httpd -DFOREGROUND
                    └─33795 /usr/sbin/httpd -DFOREGROUND

Oct 12 00:12:45 raadebayjo systemd[1]: Starting The Apache HTTP Server...
Oct 12 00:12:45 raadebayjo httpd[33791]: AH00558: httpd: Could not reliably det>
Oct 12 00:12:50 raadebayjo systemd[1]: Started The Apache HTTP Server.
Oct 12 00:12:50 raadebayjo httpd[33791]: Server configured, listening on: port >
lines 1-20/20 (END)
```

Рис. 3.2: Проверка работы веб-сервера

С помощью команды “ps auxZ | grep httpd” определил контекст безопасности веб-сервера Apache - httpd_t

```
[raadebayjo@raadebayjo ~]$ ps auxZ | grep httpd
system_u:system_r:httpd_t:s0 root 33791 0.3 0.3 20340 11668 ? Ss 00:12
0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 33792 0.0 0.2 21676 7612 ? S 00:12
0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 33793 0.6 0.6 2193664 19408 ? Sl 00:12
0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 33794 0.5 0.5 2062528 17360 ? Sl 00:12
0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 33795 0.5 0.5 2062528 17360 ? Sl 00:12
0:00 /usr/sbin/httpd -DFOREGROUND
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 raadeba+ 34047 0.0 0.0 221664 2372 pt
s/0 S+ 00:15 0:00 grep --color=auto httpd
[raadebayjo@raadebayjo ~]$
```

Рис. 3.3: Контекст безопасности веб-сервера Apache

Посмотрел текущее состояние переключателей SELinux для Apache с помощью команды “sestatus -bigrep httpd”, многие из переключателей находятся в положении “off”

```

[raadebayjo@raadebayjo ~]$ sestatus -bigrep httpd
sestatus: invalid option -- 'i'

Usage: sestatus [OPTION]

  -v  Verbose check of process and file contexts.
  -b  Display current state of booleans.

Without options, show SELinux status.
[raadebayjo@raadebayjo ~]$ 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

```

Рис. 3.4: Текущее состояние переключателей SELinux

Посмотрел статистику по политике с помощью команды “seinfo”. Множество пользователей - 8, ролей - 14, типов 4995

```
[raadebayjo@raadebayjo ~]$ 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
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
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
```

Рис. 3.5: Статистика по политике

С помощью команды “ls -lZ /var/www” посмотрел файлы и поддиректории, находящиеся в директории /var/www. Используя команду “ls -lZ /var/www/html”, определил, что в данной директории файлов нет. Только владелец/суперпользователь может создавать файлы в директории /var/www/html

```
[raadebayjo@raadebayjo ~]$ ls -lZ /var/www
total 0
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_script_exec_t:s0 6 Jul 20 11:44 cgi-bin
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_content_t:s0      6 Jul 20 11:44 html
[raadebayjo@raadebayjo ~]$ ls -lZ /var/www/html
total 0
[raadebayjo@raadebayjo ~]$
```

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

От имени суперпользователя создал html-файл /var/www/html/test.html. Кон-текст созданного файла - httpd_sys_content_t

```
[raadebayjo@raadebayjo ~]$ su
Password:
[root@raadebayjo raadebayjo]# touch /var/www/html/test.html
[root@raadebayjo raadebayjo]# vim /var/www/html/test/html
[root@raadebayjo raadebayjo]# vim /var/www/html/test.html
[root@raadebayjo raadebayjo]# cat /var/www/html/test.html
<html>
    <body>test</body>
</html>
[root@raadebayjo raadebayjo]#
```

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

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

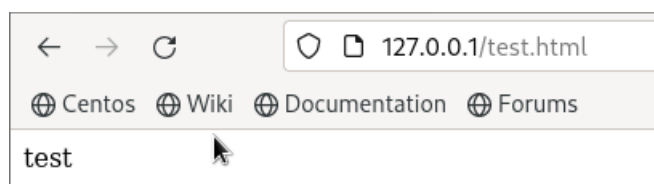


Рис. 3.8: Обращение к файлу через веб-сервер

Изучив справку `man httpd_selinux`, выяснила, 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). Изменил контекст файла на `samba_share_t` командой “`sudo chcon -t samba_share_t /var/www/html/test.html`” и проверил, что контекст поменялся

```
[root@raadebayjo raadebayjo]# man httpd_selinux
No manual entry for httpd_selinux
[root@raadebayjo raadebayjo]# exit
exit
[raadebayjo@raadebayjo ~]$ ls -Z /var/www/html/test.html
unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/test.html
[raadebayjo@raadebayjo ~]$
```

Рис. 3.9: Изменение контекста

```
[raadebayjo@raadebayjo ~]$ chcon -t samba_share_t /var/www/html/test.html
chcon: failed to change context of '/var/www/html/test.html' to 'unconfined_u:object_r:samba_share_t:s0': Operation not permitted
[raadebayjo@raadebayjo ~]$ su
Password:
[root@raadebayjo raadebayjo]# chcon -t samba_share_t /var/www/html/test.html
[root@raadebayjo raadebayjo]# ls -Z /var/www/html/test.html
unconfined_u:object_r:samba_share_t:s0 /var/www/html/test.html
[root@raadebayjo raadebayjo]#
```

Рис. 3.10: Изменение контекста

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

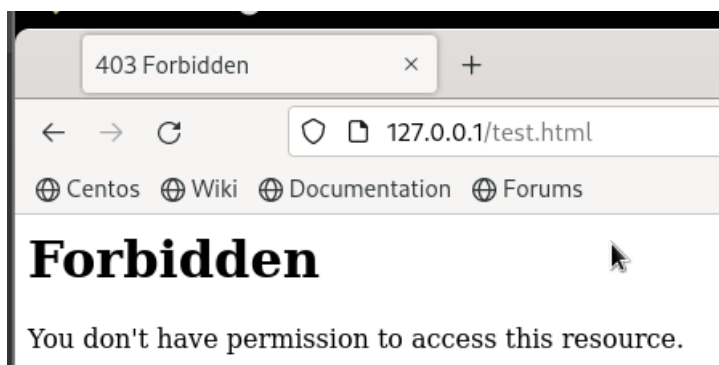


Рис. 3.11: Обращение к файлу через веб-сервер

Командой “ls -l /var/www/html/test.html” убедился, что читать данный файл может любой пользователь. Просмотрел системный лог-файл веб-сервера Apache командой “sudo tail /var/log/messages”, отображающий ошибки

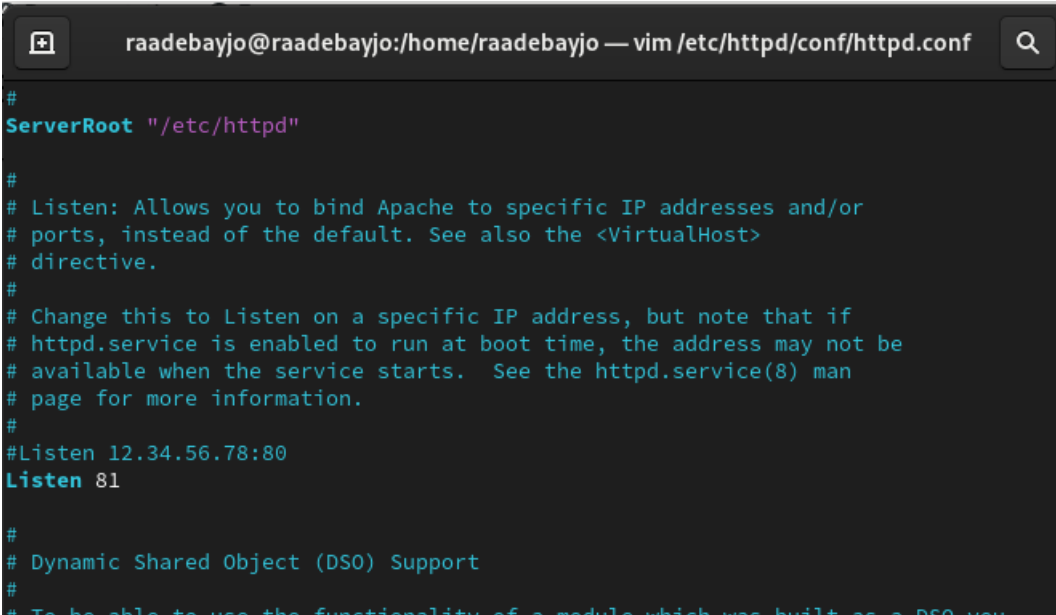
```

[root@raadebayjo raadebayjo]# ls -l /var/www/html/test.html
-rw-r--r--. 1 root root 34 Oct 12 00:26 /var/www/html/test.html
[root@raadebayjo raadebayjo]# tail /var/log/messages
Oct 12 00:40:27 raadebayjo firefox.desktop[35248]: Missing chrome or resource URL: resource:
//gre/modules/UpdateListener.jsm
Oct 12 00:40:27 raadebayjo firefox.desktop[35248]: Missing chrome or resource URL: resource:
//gre/modules/UpdateListener.sys.mjs
Oct 12 00:40:30 raadebayjo systemd[1]: Started SETroubleshoot daemon for processing new SELi
nux denial logs.
Oct 12 00:40:34 raadebayjo setroubleshoot[35528]: failed to retrieve rpm info for path '/var
/www/html/test.html':
Oct 12 00:40:34 raadebayjo systemd[1]: Created slice Slice /system/dbus-:1.1-org.fedoraproje
ct.SetroubleshootPrivileged.
Oct 12 00:40:34 raadebayjo systemd[1]: Started dbus-:1.1-org.fedoraproject.SetroubleshootPri
vileged@0.service.
Oct 12 00:40:43 raadebayjo systemd[1]: setroubleshootd.service: Deactivated successfully.
Oct 12 00:40:43 raadebayjo systemd[1]: setroubleshootd.service: Consumed 9.334s CPU time.
Oct 12 00:40:52 raadebayjo systemd[1]: dbus-:1.1-org.fedoraproject.SetroubleshootPrivileged@
0.service: Deactivated successfully.
Oct 12 00:40:52 raadebayjo systemd[1]: dbus-:1.1-org.fedoraproject.SetroubleshootPrivileged@
0.service: Consumed 12.865s CPU time.
[root@raadebayjo raadebayjo]#

```

Рис. 3.12: Просмотр log-файла

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



```

#
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

```

Рис. 3.13: Установка веб-сервера Apache на прослушивание TCP-порта 81

Перезапускаем веб-сервер Apache и анализирует лог-файлы командой “tail -nl

/var/log/messages”

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

```
[root@raadebayjo raadebayjo]# tail -n1 /var/log/httpd/error_log
[Thu Oct 12 00:53:36.384764 2023] [core:notice] [pid 35842:tid 35842] AH00094: Command line:
'/usr/sbin/httpd -D FOREGROUND'
[root@raadebayjo raadebayjo]# tail -n1 /var/log/httpd/access_log
127.0.0.1 - - [12/Oct/2023:00:40:26 +0300] "GET /favicon.ico HTTP/1.1" 404 196 "http://127.0
.0.1/test.html" "Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0"
[root@raadebayjo raadebayjo]# tail -n1 /var/log/audit/audit.log
type=SERVICE_START msg=audit(1697061216.374:243): 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/system
d" hostname=? addr=? terminal=? res=success'UID="root" AUID="unset"
[root@raadebayjo raadebayjo]#
```

Рис. 3.14: Содержание файла var/log/audit/audit.log

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

```
[root@raadebayjo raadebayjo]# semanage port -a -t http_port_t -p tcp 81
ValueError: Port tcp/81 already defined
[root@raadebayjo raadebayjo]# semanage port -l | grep http_port_t
semanage port: error: one of the arguments -a/--add -d/--delete -m/--modify -l/--list -E/--e
xtract -D/--deleteall is required
[root@raadebayjo raadebayjo]# 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@raadebayjo raadebayjo]#
```

Рис. 3.15: Проверка установки порта 81

Вернул контекст “httpd_sys_content_t” файлу “/var/www/html/test.html” коман-дой “chcon -t httpd_sys_content_t /var/www/html/test.html” (рис. 3.16) и после этого попробовал получить доступ к файлу через веб-сервер, введя адрес “http://127.0.0.1:81/test.html”, в результате чего увидел содежимое файла - слово “test”

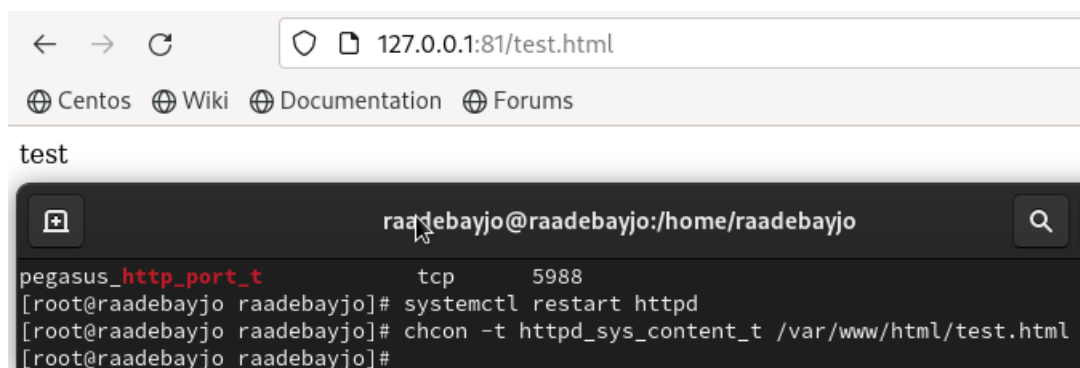


Рис. 3.16: Возвращение исходного контекста файлу

Исправил обратно конфигурационный файл apache, вернув “Listen 80”. Попытался 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. Удалил файл “/var/www/html/test.html” командой “rm /var/www/html/test.html”

```
[root@raadebayjo raadebayjo]# vim /etc/httpd/conf/httpd.conf
[root@raadebayjo raadebayjo]# semanage port -d -t http_port_t -p tcp 81
ValueError: Port tcp/81 is defined in policy, cannot be deleted
[root@raadebayjo raadebayjo]# 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@raadebayjo raadebayjo]# rm /var/www/html/test.html
rm: remove regular file '/var/www/html/test.html'? y
[root@raadebayjo raadebayjo]#
```

Рис. 3.17: Возвращение Listen 80 и попытка удалить порт 81

4 Выводы

В ходе выполнения данной лабораторной работы я развил навыки администрирования ОС Linux, получил первое практическое знакомство с технологией SELinux и проверил работу SELinux на практике совместно с веб-сервером Apache.

5 Список литературы

1. Кулябов Д. С. *Лабораторная работа №5**: 006-lab_selinux.pdf*
2. Использование SETUID, SETGID и Sticky bit для расширенной настройки прав доступа в операционных системах Linux [Электронный ресурс]. 2023.URL: <https://ruvds.com/ru/helpcenter/suid-sgid-sticky-bit-linux/> (дата обращения: 05.10.2023)