

# **Лабораторная работа №6**

**Дисциплина: Основы информационной безопасности**

Аветисян Давид Артурович

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Теоретическое введение</b>	<b>6</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>8</b>
<b>4</b>	<b>Выводы</b>	<b>18</b>
<b>5</b>	<b>Список литературы</b>	<b>19</b>

# List of Figures

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 . . . . .	13
3.10	Обращение к файлу через веб-сервер . . . . .	13
3.11	Просмотр log-файла . . . . .	14
3.12	Установка веб-сервера Apache на прослушивание TCP-порта 81 . .	14
3.13	Перезапуск веб-сервера и анализ лог-файлов . . . . .	15
3.14	Содержание файла var/log/audit/audit.log . . . . .	15
3.15	Проверка установки порта 81 . . . . .	16
3.16	Возвращение исходного контекста файлу . . . . .	16
3.17	Обращение к файлу через веб-сервер . . . . .	16
3.18	Возвращение Listen 80 и попытка удалить порт 81 . . . . .	17
3.19	Удаление файла test.html . . . . .	17

## List of Tables

# 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 Теоретическое 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. Более подробно см. в [1].

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:

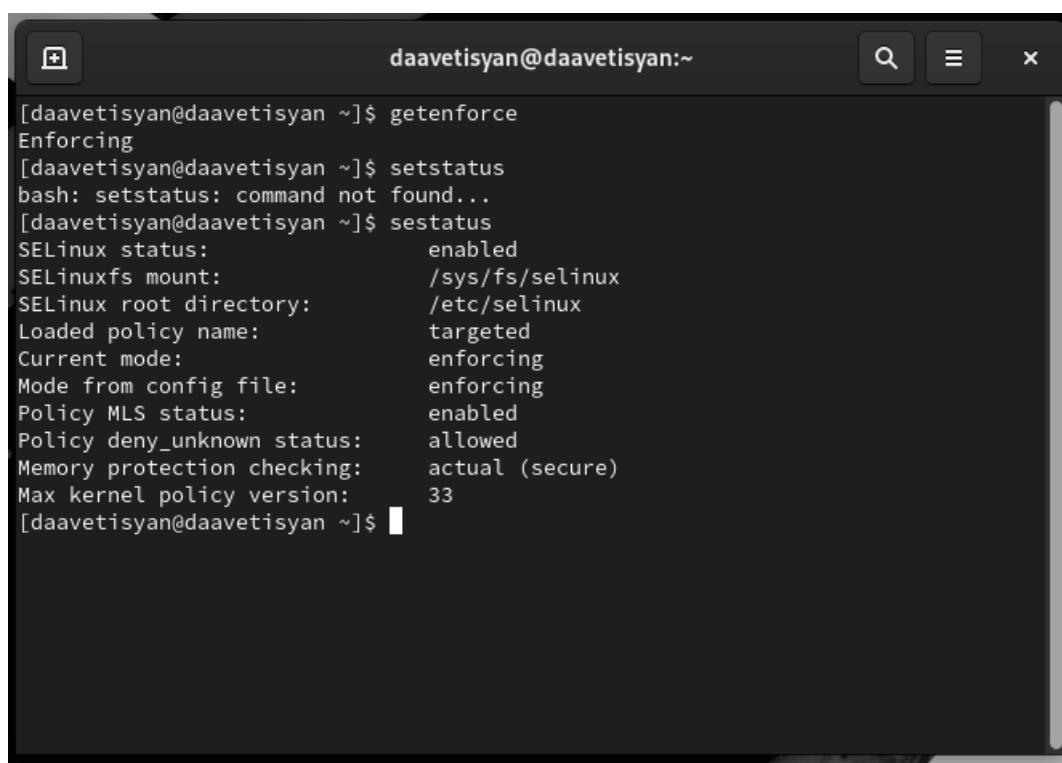
- чтобы открывать динамические РНР-страницы,
- для распределения поступающей на сервер нагрузки,
- для обеспечения отказоустойчивости сервера,
- чтобы потренироваться в настройке

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

- 1) Входим в систему под своей учетной записью и убеждаемся, что SELinux работает в режиме enforcing политики targeted с помощью команд “getenforce” и “sestatus” (fig. 3.1).



```
[daavetisyan@daavetisyan ~]$ getenforce
Enforcing
[daavetisyan@daavetisyan ~]$ setstatus
bash: setstatus: command not found...
[daavetisyan@daavetisyan ~]$ 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
[daavetisyan@daavetisyan ~]$
```

Figure 3.1: Проверка режима enforcing политики targeted

- 2) Обращаемся с помощью браузера к веб-серверу, запущенному на моем компьютере, и убеждаемся, что последний работает с помощью команды “service httpd status” (fig. 3.2).

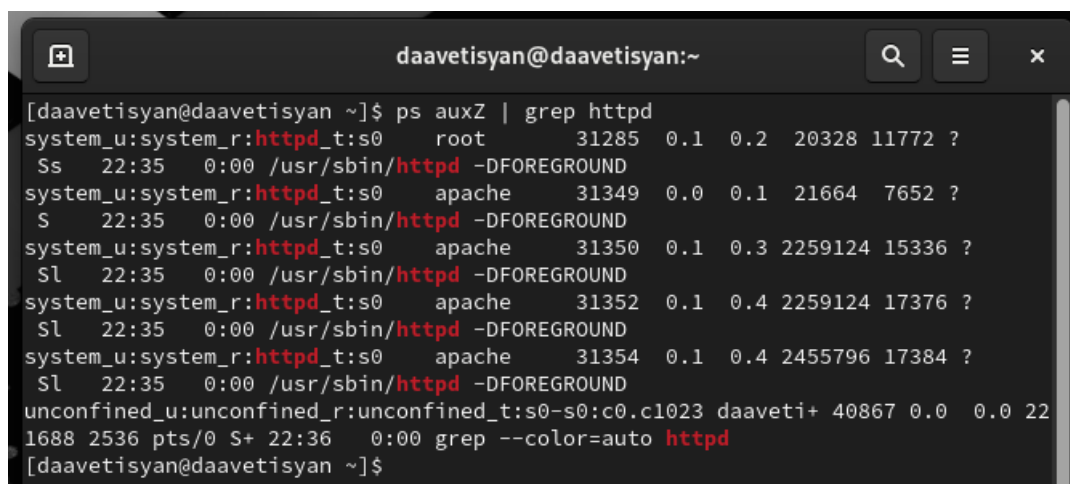


```
[daavetisyan@daavetisyan ~]$ service httpd start
Redirecting to /bin/systemctl start httpd.service
[daavetisyan@daavetisyan ~]$ service httpd status
Redirecting to /bin/systemctl status httpd.service
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: d>
   Active: active (running) since Sat 2023-10-14 22:35:08 MSK; 3s ago
     Docs: man:httpd.service(8)
  Main PID: 31285 (httpd)
    Status: "Started, listening on: port 80"
     Tasks: 213 (limit: 24610)
    Memory: 37.6M
       CPU: 128ms
    CGroup: /system.slice/httpd.service
            └─31285 /usr/sbin/httpd -DFOREGROUND
              └─31349 /usr/sbin/httpd -DFOREGROUND
                └─31350 /usr/sbin/httpd -DFOREGROUND
                  └─31352 /usr/sbin/httpd -DFOREGROUND
                    └─31354 /usr/sbin/httpd -DFOREGROUND

окт 14 22:35:08 daavetisyan systemd[1]: Starting The Apache HTTP Server...
окт 14 22:35:08 daavetisyan httpd[31285]: AH00558: httpd: Could not reliably de>
окт 14 22:35:08 daavetisyan httpd[31285]: Server configured, listening on: port>
окт 14 22:35:08 daavetisyan systemd[1]: Started The Apache HTTP Server.
lines 1-20/20 (END)...skipping...
```

Figure 3.2: Проверка работы веб-сервера

- 3) С помощью команды “ps auxZ | grep httpd” определяем контекст безопасности веб-сервера Apache - httpd\_t (fig. 3.3).



```
daavetisyan@daavetisyan:~$ ps auxZ | grep httpd
system_u:system_r:httpd_t:s0 root 31285 0.1 0.2 20328 11772 ?
Ss 22:35 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 31349 0.0 0.1 21664 7652 ?
S 22:35 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 31350 0.1 0.3 2259124 15336 ?
Sl 22:35 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 31352 0.1 0.4 2259124 17376 ?
Sl 22:35 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 31354 0.1 0.4 2455796 17384 ?
Sl 22:35 0:00 /usr/sbin/httpd -DFOREGROUND
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 daaveti+ 40867 0.0 0.0 22
1688 2536 pts/0 S+ 22:36 0:00 grep --color=auto httpd
[daavetisyan@daavetisyan ~]$
```

Figure 3.3: Контекст безопасности веб-сервера Apache

- 4) Посмотрим текущее состояние переключателей SELinux для Apache с помощью команды “sestatus -bigrep httpd”, многие из переключателей находятся

в положении “off” (fig. 3.4).

```
[daavetisyan@daavetisyan ~]$ 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.
[daavetisyan@daavetisyan ~]$ 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
boinc_execmem                  on
```

Figure 3.4: Текущее состояние переключателей SELinux

- 5) Посмотрим статистику по политике с помощью команды “seinfo”. Множество пользователей - 8, ролей - 14, типов 5100 (fig. 3.5).

```

[daavetisyan@daavetisyan ~]$ seinfo
Statistics for policy file: /sys/fs/selinux/policy
Policy Version:          33 (MLS enabled)
Target Policy:           selinux
Handle unknown classes:  allow
Classes:                 135
Sensitivities:           1
Types:                   5100
Users:                   8
Booleans:                353
Allow:                   65008
Auditallow:              170
Type_trans:              265344
Type_member:             35
Role allow:              38
Constraints:             70
MLS Constrains:          72
Permissives:             2
Defaults:                7
Allowxperm:              0
Auditallowxperm:         0
Ibendportcon:            0
Initial SIDs:            27
Genfscon:                109
Netifcon:                0
Permissions:             457
Categories:             1024
Attributes:              258
Roles:                   14
Cond. Expr.:            384
Neverallow:              0
Dontaudit:               8572
Type_change:             87
Range_trans:             6164
Role_trans:              420
Validate_trans:          0
MLS Val. Tran:           0
Polcap:                  6
Typebounds:              0
Neverallowxperm:         0
Dontauditxperm:          0
Ibpkeycon:               0
Fs_use:                  35
Portcon:                 660
Nodecon:                 0

```

Figure 3.5: Статистика по политике

- 6) С помощью команды “ls -lZ /var/www” посмотрим файлы и поддиректории, находящиеся в директории /var/www. Используя команду “ls -lZ /var/www/html”, определяем, что в данной директории файлов нет. Только владелец или суперпользователь может создавать файлы в директории /var/www/html (fig. 3.6).

```

[daavetisyan@daavetisyan ~]$ 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
[daavetisyan@daavetisyan ~]$ ls -lZ /var/www/html
итого 0
[daavetisyan@daavetisyan ~]$

```

Figure 3.6: Просмотр файлов и поддиректорий в директории /var/www

- 7) От имени суперпользователя создаём html-файл /var/www/html/test.html. Контекст созданного файла - httpd\_sys\_content\_t (fig. 3.7).

```

[root@daavetisyan ~]# su -
[root@daavetisyan ~]# touch /var/www/html/test.html
[root@daavetisyan ~]# nano /var/www/html/test.html
[root@daavetisyan ~]# cat /var/www/html/test.html
<html>
<body>test</body>
</html>
[root@daavetisyan ~]# su - daavetisyan
[daavetisyan@daavetisyan ~]$ ls -lZ /var/www/html/
итого 4
-rw-r--r--. 1 root root unconfined_u:object_r:httpd_sys_content_t:s0 33 окт 14 22:43 test.html
[daavetisyan@daavetisyan ~]$

```

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

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

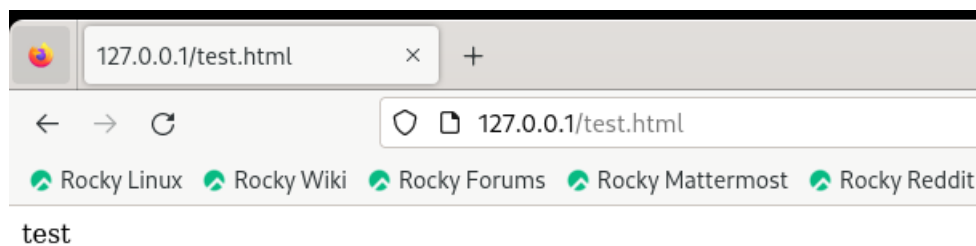


Figure 3.8: Обращение к файлу через веб-сервер

- 9) Изучив справку man httpd\_selinux, выясняем, 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). Изменяем контекст файла на samba\_share\_t командой “sudo chcon -t samba\_share\_t /var/www/html/test.html” и проверяем, что контекст поменялся (fig. 3.9).

```
[daavetisyan@daavetisyan ~]$ man httpd
[daavetisyan@daavetisyan ~]$ man selinux
[daavetisyan@daavetisyan ~]$ ls -Z /var/www/html/test.html
unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/test.html
[daavetisyan@daavetisyan ~]$ 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
[daavetisyan@daavetisyan ~]$ sudo chcon -t samba_share_t /var/www/html/test.html
[sudo] пароль для daavetisyan:
[daavetisyan@daavetisyan ~]$ ls -Z /var/www/html/test.html
unconfined_u:object_r:samba_share_t:s0 /var/www/html/test.html
[daavetisyan@daavetisyan ~]$
```

Figure 3.9: Изменение контекста

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

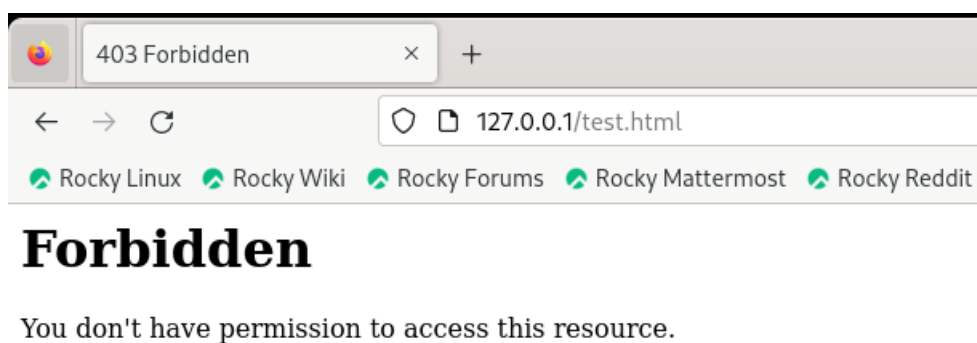


Figure 3.10: Обращение к файлу через веб-сервер

- 11) Командой “ls -l /var/www/html/test.html” убеждаемся, что читать данный файл может любой пользователь. Просматриваем системный лог-файл веб-сервера Apache командой “sudo tail /var/log/messages”, отображающий ошибки (fig. 3.11).

```
[daavetisyan@daavetisyan ~]$ sudo tail /var/log/messages
Oct 14 22:53:33 daavetisyan setroubleshoot[42130]: failed to retrieve rpm info for path '/var/www/html/test.html':
Oct 14 22:53:33 daavetisyan systemd[1]: Created slice Slice /system/dbus-1.1-org.fedoraproject.SetroubleshootPrivileged.
Oct 14 22:53:33 daavetisyan systemd[1]: Started dbus-1.1-org.fedoraproject.SetroubleshootPrivileged@0.service.
Oct 14 22:53:34 daavetisyan setroubleshoot[42130]: SELinux запрещает /usr/sbin/httpd доступ getattr к файл /var/www/html/test.html
Oct 14 22:53:34 daavetisyan setroubleshoot[42130]: SELinux запрещает /usr/sbin/httpd доступ getattr к файл /var/www/html/test.html
е исправить метку.$TARGET3знак _PATH по умолчанию должен быть httpd_sys_content_t#012То вы можете запустить restorecon. Возможно, п
м случае попытайтесь соответствующим образом изменить следующую команду.#012Сделать#012# /sbin/restorecon -v /var/www/html/test.ht
те лечить test.html как общедоступный контент#012То необходимо изменить метку test.html с public_content_t на public_content_rw_t.
/www/html/test.html'#012#012**** Модуль catchall предлагает (точность 1.41) *****#012#012Если вы считаете
отчет об ошибке.#012Чтобы разрешить доступ, можно создать локальный модуль политики.#012Сделать#012разрешить этот доступ сейчас, в
Oct 14 22:53:34 daavetisyan setroubleshoot[42130]: SELinux запрещает /usr/sbin/httpd доступ getattr к файл /var/www/html/test.html
е исправить метку.$TARGET3знак _PATH по умолчанию должен быть httpd_sys_content_t#012То вы можете запустить restorecon. Возможно, п
м случае попытайтесь соответствующим образом изменить следующую команду.#012Сделать#012# /sbin/restorecon -v /var/www/html/test.ht
те лечить test.html как общедоступный контент#012То необходимо изменить метку test.html с public_content_t на public_content_rw_t.
/www/html/test.html'#012#012**** Модуль catchall предлагает (точность 1.41) *****#012#012Если вы считаете
отчет об ошибке.#012Чтобы разрешить доступ, можно создать локальный модуль политики.#012Сделать#012разрешить этот доступ сейчас, в
Oct 14 22:53:44 daavetisyan systemd[1]: dbus-1.1-org.fedoraproject.SetroubleshootPrivileged@0.service: Deactivated successfully.
Oct 14 22:53:44 daavetisyan systemd[1]: dbus-1.1-org.fedoraproject.SetroubleshootPrivileged@0.service: Consumed 1.345s CPU time.
Oct 14 22:53:44 daavetisyan systemd[1]: setroubleshootd.service: Deactivated successfully.
[daavetisyan@daavetisyan ~]$
```

Figure 3.11: Просмотр log-файла

- 12) В файле /etc/httpd/conf/httpd.conf заменяем строчку “Listen 80” на “Listen 81”, чтобы установить веб-сервер Apache на прослушивание TCP-порта 81 (fig. 3.12).



```
daavetisyan@daavetisyan:~ — sudo nano /etc/httpd/conf/http...
GNU nano 5.6.1 /etc/httpd/conf/httpd.conf
#
# 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
^G Справка ^O Записать ^W Поиск ^K Вырезать ^T Выполнить ^C Позиция
^X Выход ^R ЧитФайл ^\ Замена ^U Вставить ^J Вывернуть ^_ К строке
```

Figure 3.12: Установка веб-сервера Apache на прослушивание TCP-порта 81

- 13) Перезапускаем веб-сервер Apache и анализируем лог-файлы командой “tail

-nl /var/log/messages” (fig. 3.13).

```
[daavetisyan@daavetisyan ~]$ su -
Пароль:
[root@daavetisyan ~]# service httpd restart
Redirecting to /bin/systemctl restart httpd.service
[root@daavetisyan ~]# tail -n1 /var/log/messages
Oct 14 22:58:30 daavetisyan systemd[1]: Started The Apache HTTP Server.
[root@daavetisyan ~]# tail -n3 /var/log/messages
Oct 14 22:58:30 daavetisyan systemd[1]: Started The Apache HTTP Server.
Oct 14 22:58:42 daavetisyan gnome-shell[1761]: Window manager warning: last_user_time (2134979)
_NET_ACTIVE_WINDOW. Trying to work around...
Oct 14 22:58:42 daavetisyan gnome-shell[1761]: Window manager warning: W13 appears to be one of
[root@daavetisyan ~]#
```

Figure 3.13: Перезапуск веб-сервера и анализ лог-файлов

- 14) Просматриваем файлы “var/log/http/error\_log”, “/var/log/http/access\_log” и “/var/log/audit/audit.log” и выясняем, что запись появилась в последнем файле (fig. 3.14).

```
[root@daavetisyan ~]# cat /var/log/audit/audit.log
type=DAEMON_START msg=audit(1695499329.607:7029): op=start ver=3.0.7 format=enriched kernel=5.14.0-284.11.1.el9_2.x86_64 auid=4294967295 ses=4294967295 subj=system_u:system_r:auditd:s0
type=CONFIG_CHANGE msg=audit(1695499329.675:5): op=set audit_backlog_limit=8192 old=64 auid=4294967295 ses=4294967295 subj=system_u:system_r:auditd:s0
type=SYSCALL msg=audit(1695499329.675:5): arch=c000003e syscall=44 success=yes exit=60 a0=3 a1=7fff6e3904a0 a2=3c a3=0 items=1
295 comm="auditctl" exe="/usr/sbin/auditctl" subj=system_u:system_r:unconfined_service_t:s0 key=(null)ARCH=x86_64 SYSCALL=ser
type=PROCTITLE msg=audit(1695499329.675:5): proctitle=2F7362696E2F617564697463746C002D52002F6574632F61756469742F61756469742E7
type=CONFIG_CHANGE msg=audit(1695499329.677:6): op=set audit_failure=1 old=1 auid=4294967295 ses=4294967295 subj=system_u:sys
type=SYSCALL msg=audit(1695499329.677:6): arch=c000003e syscall=44 success=yes exit=60 a0=3 a1=7fff6e3904a0 a2=3c a3=0 items=
295 comm="auditctl" exe="/usr/sbin/auditctl" subj=system_u:system_r:unconfined_service_t:s0 key=(null)ARCH=x86_64 SYSCALL=ser
type=PROCTITLE msg=audit(1695499329.677:6): proctitle=2F7362696E2F617564697463746C002D52002F6574632F61756469742F61756469742E7
type=CONFIG_CHANGE msg=audit(1695499329.679:7): op=set audit_backlog_wait_time=60000 old=60000 auid=4294967295 ses=4294967295
type=SYSCALL msg=audit(1695499329.679:7): arch=c000003e syscall=44 success=yes exit=60 a0=3 a1=7fff6e3904a0 a2=3c a3=0 items=
295 comm="auditctl" exe="/usr/sbin/auditctl" subj=system_u:system_r:unconfined_service_t:s0 key=(null)ARCH=x86_64 SYSCALL=ser
type=PROCTITLE msg=audit(1695499329.679:7): proctitle=2F7362696E2F617564697463746C002D52002F6574632F61756469742F61756469742E7
type=SERVICE_START msg=audit(1695499329.682:8): pid=1 uid=0 auid=4294967295 ses=4294967295 subj=system_u:system_r:init_t:s0 n
AUID="unset"
type=SYSTEM_BOOT msg=audit(1695499329.692:9): pid=762 uid=0 auid=4294967295 ses=4294967295 subj=system_u:system_r:init_t:s0 n
ss'UID="root" AUID="unset"
type=SERVICE_START msg=audit(1695499329.698:10): pid=1 uid=0 auid=4294967295 ses=4294967295 subj=system_u:system_r:init_t:s0
ess'UID="root" AUID="unset"
type=SERVICE_START msg=audit(1695499330.459:11): pid=1 uid=0 auid=4294967295 ses=4294967295 subj=system_u:system_r:init_t:s0
ot" AUID="unset"
type=SERVICE_START msg=audit(1695499330.487:12): pid=1 uid=0 auid=4294967295 ses=4294967295 subj=system_u:system_r:init_t:s0
ess'UID="root" AUID="unset"
type=BPF msg=audit(1695499330.493:13): prog-id=18 op=LOAD
type=SERVICE_START msg=audit(1695499330.580:14): pid=1 uid=0 auid=4294967295 ses=4294967295 subj=system_u:system_r:init_t:s0
"root" AUID="unset"
type=BPF msg=audit(1695499330.584:15): prog-id=19 op=LOAD
type=SERVICE_START msg=audit(1695499330.597:16): pid=1 uid=0 auid=4294967295 ses=4294967295 subj=system_u:system_r:init_t:s0
root" AUID="unset"
type=SERVICE_START msg=audit(1695499330.600:17): pid=1 uid=0 auid=4294967295 ses=4294967295 subj=system_u:system_r:init_t:s0
ID="root" AUID="unset"
type=SERVICE_START msg=audit(1695499330.603:18): pid=1 uid=0 auid=4294967295 ses=4294967295 subj=system_u:system_r:init_t:s0
" AUID="unset"
```

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

- 15) Выполняем команду “semanage port -a -t http\_port\_t -p tcp 81” и убеждаемся, что порт TCP-81 установлен. Проверяем список портов командой “semanage

port -l | grep http\_port\_t”, убеждаемся, что порт 81 есть в списке и запускаем веб-сервер Apache снова (fig. 3.15).

```
[root@daavetisyan ~]# semanage port -a -t http_port_t -p tcp 81
ValueError: Порт tcp/81 уже определен
[root@daavetisyan ~]# 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@daavetisyan ~]# service httpd restart
Redirecting to /bin/systemctl restart httpd.service
[root@daavetisyan ~]#
```

Figure 3.15: Проверка установки порта 81

- 16) Вернём контекст “httpd\_sys\_content\_t” файлу “/var/www/html/test.html” командой “chcon -t httpd\_sys\_content\_t /var/www/html/test.html” (fig. 3.16) и после этого пробуем получить доступ к файлу через веб-сервер, введя адрес “http://127.0.0.1:81/test.html”, в результате чего увидим содержимое файла - слово “test” (fig. 3.17).

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

Figure 3.16: Возвращение исходного контекста файлу

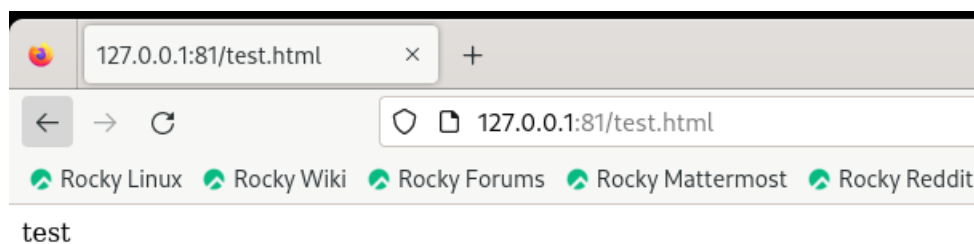


Figure 3.17: Обращение к файлу через веб-сервер

- 17) Исправим обратно конфигурационный файл apache, вернув “Listen 80”. Попробуем 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 (fig. 3.18).



```

[root@daavetisyan ~]# nano /etc/httpd/conf/httpd.conf
[root@daavetisyan ~]# semanage port -d -t http_port_t -p tcp 81
ValueError: Порт tcp/81 определен на уровне политики и не может быть удален
[root@daavetisyan ~]# 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@daavetisyan ~]# cat /etc/httpd/conf/httpd.conf | grep "Listen"
# Listen: Allows you to bind Apache to specific IP addresses and/or
# Change this to Listen on a specific IP address, but note that if
#Listen 12.34.56.78:80
Listen 80
[root@daavetisyan ~]#

```

Figure 3.18: Возвращение Listen 80 и попытка удалить порт 81

- 18) Удаляем файл “/var/www/html/test.html” командой “rm /var/www/html/test.html” (fig. 3.19).

```

[root@daavetisyan ~]# rm -R /var/www/html/test.html
rm: удалить обычный файл '/var/www/html/test.html'? y
[root@daavetisyan ~]# ls /var/www/html/
[root@daavetisyan ~]#

```

Figure 3.19: Удаление файла test.html

## 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

- SELinux – описание и особенности работы с системой [Электронный ресурс]. URL: <https://habr.com/ru/company/kingservers/blog/209644/>.
- Что такое Apache и зачем он нужен? [Электронный ресурс]. URL: <https://2domains.ru/support/vps-i-servery/shto-takoye-apache>.