

Лабораторная работа №6

Дисциплина: Основы информационной безопасности

Дарижапов Тимур Андреевич

Содержание

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

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

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:

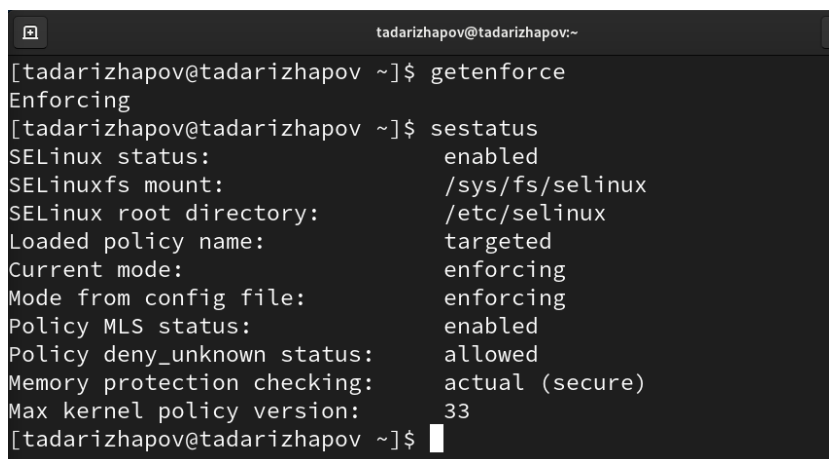
- чтобы открывать динамические РНР-страницы,
- для распределения поступающей на сервер нагрузки,
- для обеспечения отказоустойчивости сервера,
- чтобы потренироваться в настройке

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”(Рисунок 3.1).

A terminal window with a dark background and light text. The title bar shows 'tadarizhapov@tadarizhapov:~'. The user enters the command 'getenforce' and the output is 'Enforcing'. Then the user enters 'sestatus' and the output shows various SELinux settings: status is enabled, mount is /sys/fs/selinux, root directory is /etc/selinux, loaded policy name is targeted, current mode is enforcing, mode from config file is enforcing, policy MLS status is enabled, policy deny_unknown status is allowed, memory protection checking is actual (secure), and max kernel policy version is 33.

```
tadarizhapov@tadarizhapov ~]$ getenforce
Enforcing
tadarizhapov@tadarizhapov ~]$ 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
tadarizhapov@tadarizhapov ~]$
```

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” (Рисунок 3.2).


```
[tadarizhapov@tadarizhapov ~]$ service httpd start
Redirecting to /bin/systemctl start httpd.service
[tadarizhapov@tadarizhapov ~]$ 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 Fri 2023-10-13 22:27:53 MSK; 4s ago
     Docs: man:httpd.service(8)
   Main PID: 52999 (httpd)
   Status: "Started, listening on: port 80"
    Tasks: 213 (limit: 24670)
   Memory: 33.3M
      CPU: 50ms
   CGroup: /system.slice/httpd.service
           └─52999 /usr/sbin/httpd -DFOREGROUND
             └─53007 /usr/sbin/httpd -DFOREGROUND
               └─53008 /usr/sbin/httpd -DFOREGROUND
                 └─53009 /usr/sbin/httpd -DFOREGROUND
                   └─53010 /usr/sbin/httpd -DFOREGROUND

окт 13 22:27:53 tadarizhapov.localdomain systemd[1]: Starting The Apache HTTP Se>
окт 13 22:27:53 tadarizhapov.localdomain systemd[1]: Started The Apache HTTP Ser>
окт 13 22:27:53 tadarizhapov.localdomain httpd[52999]: Server configured, listen>
lines 1-19/19 (END)
```

Figure 3.2: Проверка работы веб-сервера

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

```
tadarizhapov@tadarizhapov:~$ ps auxZ | grep httpd
system_u:system_r:httpd_t:s0 root 53586 0.0 0.2 20116 11472
? Ss 14:11 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 53594 0.0 0.1 21600 7228
? S 14:11 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 53595 0.0 0.4 2455692 17104
? Sl 14:11 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 53596 0.0 0.4 2324556 17100
? Sl 14:11 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 53597 0.0 0.4 2259020 17100
? Sl 14:11 0:00 /usr/sbin/httpd -DFOREGROUND
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 tadariz+ 53881 0.0
0.0 221820 2340 pts/0 S+ 14:13 0:00 grep --color=auto httpd
[tadarizhapov@tadarizhapov ~]$
```

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”(Рисунок 3.4).

```

[tadarizhapov@tadarizhapov ~]$ 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.
[tadarizhapov@tadarizhapov ~]$ 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

```

Figure 3.4: Текущее состояние переключателей SELinux

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

```
[tadarizhapov@tadarizhapov ~]$ 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        Categories:       1024
Types:        5100     Attributes:       258
Users:        8        Roles:           14
Booleans:     353      Cond. Expr.:     384
Allow:        65009     Neverallow:      0
Auditallow:   170      Dontaudit:       8572
Type_trans:   265337    Type_change:     87
Type_member:  35        Range_trans:     6164
Role allow:   38        Role_trans:      420
Constraints:  70        Validatetrans:   0
MLS Constrain: 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:         660
Netifcon:     0        Nodecon:         0
[tadarizhapov@tadarizhapov ~]$
```

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 супер-пользователь может создавать файлы в директории /var/www/html(Рисунок 3.6).

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

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 (Рисунок 3.7).

```

[tadarizhapov@tadarizhapov ~]$ su -
Пароль:
[root@tadarizhapov ~]# touch /var/www/html/test.html
[root@tadarizhapov ~]# nano /var/www/html/test.html
[root@tadarizhapov ~]# cat /var/www/html/test.html
<html>
<body>test</body>
</html>

[root@tadarizhapov ~]# su - tadarizhapov
[tadarizhapov@tadarizhapov ~]$ ls -lZ /var/www/html/
итого 4
-rw-r--r--. 1 root root unconfined_u:object_r:httpd_sys_content_t:s0 34 ок
т 14 14:33 test.html
[tadarizhapov@tadarizhapov ~]$

```

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 (Рисунок 3.8).

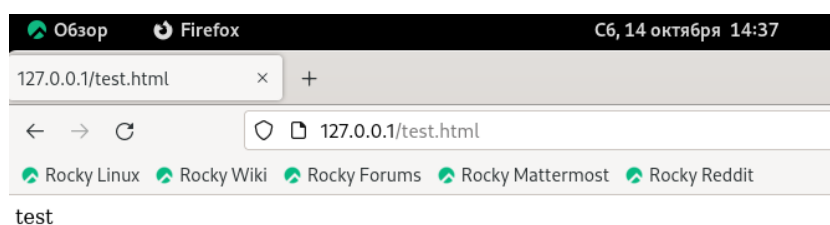


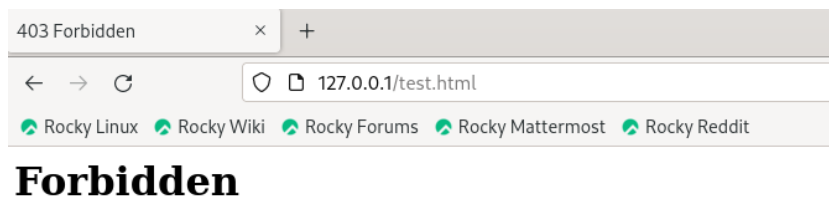
Figure 3.8: Обращение к файлу через веб-сервер

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(Рисунок 3.9).

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

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) (Рисунок 3.10).



You don't have permission to access this resource.

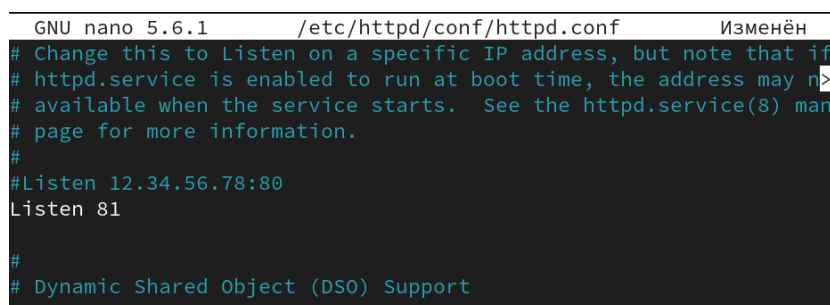
Figure 3.10: Обращение к файлу через веб-сервер

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 (Рисунок 3.11).

```
[tadarizhapov@tadarizhapov ~]$ ls -l /var/www/html/test.html
-rw-r--r--. 1 root root 33 окт 14 14:35 /var/www/html/test.html
[tadarizhapov@tadarizhapov ~]$ sudo tail /var/log/messages
Oct 14 14:51:14 tadarizhapov systemd[1]: Started dbus-1.1-1-org.fedoraproject.SetroubleshootPrivileged@0.service.
Oct 14 14:51:15 tadarizhapov setroubleshoot[55685]: SELinux запрещает /usr/sbin/httpd доступ getattr к файлу /var/www/html/test.html. Для выполнения всех сообщений SELinux: sealert -l fc59836c-9485-4746-9c8f-200eb8ed45cd
Oct 14 14:51:15 tadarizhapov setroubleshoot[55685]: SELinux запрещает /usr/sbin/httpd доступ getattr к файлу /var/www/html/test.html.#012#012***** Модуль restorecon предлагает (точность 92.2) *****#012#012Если вы хотите исправить метку.$TARGETЗнак
.PATH по умолчанию должен быть httpd_sys_content_t.#012То вы можете запустить restorecon. Возможно, попытка доступа была остановлена и
в-за недостаточных разрешений для доступа к родительскому каталогу, и в этом случае попытайтесь соответствующим образом изменить следу
ющую команду.#012Сделать#012# /sbin/restorecon -v /var/www/html/test.html#012#012***** Модуль public_content предлагает (точность 7.8
3) *****#012#012Если вы хотите лечить test.html как общедоступный контент#012То необходимо изменить метку test.html с
public_content_t на public_content_rw_t.#012Сделать#012# semanage fcontext -a -t public_content_t '/var/www/html/test.html'#012# rest
orecon -v '/var/www/html/test.html'#012#012***** Модуль catchall предлагает (точность 1.41) *****#012#012Если
вы считаете, что httpd должно быть разрешено getattr доступ к test.html file по умолчанию.#012То рекомендуется создать отчет об ошибке
.#012Тобы разрешить доступ, можно создать локальный модуль политики.#012Сделать#012#разрешить этот доступ сейчас, выполнив:#012# ausea
rch -c 'httpd' --raw | audit2allow -M my-httpd#012# semodule -X 300 -i my-httpd.pp#012
Oct 14 14:51:15 tadarizhapov setroubleshoot[55685]: SELinux запрещает /usr/sbin/httpd доступ getattr к файлу /var/www/html/test.html. Д
ля выполнения всех сообщений SELinux: sealert -l fc59836c-9485-4746-9c8f-200eb8ed45cd
Oct 14 14:51:15 tadarizhapov setroubleshoot[55685]: SELinux запрещает /usr/sbin/httpd доступ getattr к файлу /var/www/html/test.html.#0
12#012***** Модуль restorecon предлагает (точность 92.2) *****#012#012Если вы хотите исправить метку.$TARGETЗнак
.PATH по умолчанию должен быть httpd_sys_content_t.#012То вы можете запустить restorecon. Возможно, попытка доступа была остановлена и
в-за недостаточных разрешений для доступа к родительскому каталогу, и в этом случае попытайтесь соответствующим образом изменить следу
ющую команду.#012Сделать#012# /sbin/restorecon -v /var/www/html/test.html#012#012***** Модуль public_content предлагает (точность 7.8
3) *****#012#012Если вы хотите лечить test.html как общедоступный контент#012То необходимо изменить метку test.html с
public_content_t на public_content_rw_t.#012Сделать#012# semanage fcontext -a -t public_content_t '/var/www/html/test.html'#012# rest
```

Figure 3.11: Просмотр log-файла

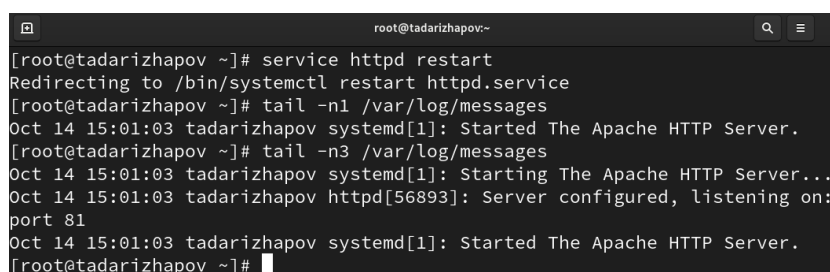
12) В файле /etc/httpd/conf/httpd.conf заменяем строчку “Listen 80” на “Listen 81”, чтобы установить веб-сервер Apache на прослушивание TCP-порта 81 (Рисунок 3.12).



```
GNU nano 5.6.1 /etc/httpd/conf/httpd.conf Изменён
# 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 n
# 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
#
```

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

13) Перезапускаем веб-сервер Apache и анализируем лог-файлы командой “tail -n1 /var/log/messages” (Рисунок 3.13).



```
root@tadarizhapov:~
[root@tadarizhapov ~]# service httpd restart
Redirecting to /bin/systemctl restart httpd.service
[root@tadarizhapov ~]# tail -n1 /var/log/messages
Oct 14 15:01:03 tadarizhapov systemd[1]: Started The Apache HTTP Server.
[root@tadarizhapov ~]# tail -n3 /var/log/messages
Oct 14 15:01:03 tadarizhapov systemd[1]: Starting The Apache HTTP Server...
Oct 14 15:01:03 tadarizhapov httpd[56893]: Server configured, listening on:
port 81
Oct 14 15:01:03 tadarizhapov systemd[1]: Started The Apache HTTP Server.
[root@tadarizhapov ~]#
```

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 (Рисунок 3.14).

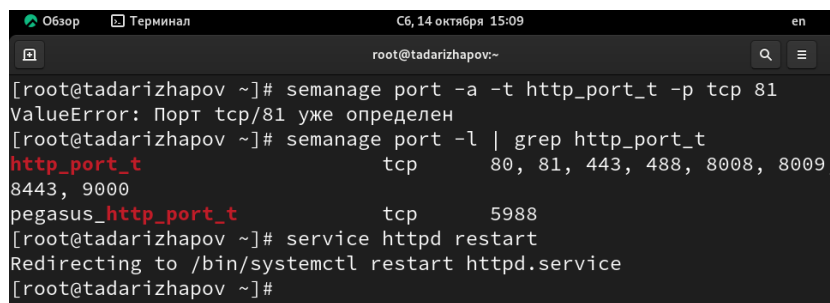
```

[root@tadarizhapov ~]# cat /var/log/audit/audit.log
type=DAEMON_START msg=audit(1697224687.684:967): op=star ver=3.0.7 format=enriched kernel=5.14.0-284.11.1.el9_2.x86_64
type=SERVICE_START msg=audit(1697224687.691:5): pid=1 uid=0 auid=4294967295 ses=4294967295 subj=system_u:system_r:auditd_t:s0 res=successAUID="unset" UID="root"
type=CONFIG_CHANGE msg=audit(1697224687.726:6): op=set audit_backlog_limit=8192 old=64 auid=4294967295 ses=4294967295 subj=system_u:system_r:unconfined_service_t:s0 res=1AUID="unset"
type=SYSCALL msg=audit(1697224687.726:6): arch=c000003e syscall=44 success=yes exit=60 a0=3 a1=7ffcb5d0c90 a2=3c a3=0 items=0 ppid=754 pid=764 auid=4294967295 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=(none) ses=4294967295 comm="auditctl" exe="/usr/sbin/auditctl" subj=system_u:system_r:unconfined_service_t:s0 key=(null)ARCH=x86_64 SYSCALL=sendto AUID="unset" UID="root" GID="root" EUID="root" SUID="root" FSUID="root" Egid="root" SGID="root" FSGID="root"
type=PROCTITLE msg=audit(1697224687.726:6): proctitle=2F7362696E2F617564697463746C002D52002F6574632F61756469742F61756469742E72756C6573
type=CONFIG_CHANGE msg=audit(1697224687.726:7): op=set audit_failure=1 old=1 auid=4294967295 ses=4294967295 subj=system_u:system_r:unconfined_service_t:s0 res=1AUID="unset"
type=SYSCALL msg=audit(1697224687.726:7): arch=c000003e syscall=44 success=yes exit=60 a0=3 a1=7ffcb5d0c90 a2=3c a3=0 items=0 ppid=754 pid=764 auid=4294967295 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=(none) ses=4294967295 comm="auditctl" exe="/usr/sbin/auditctl" subj=system_u:system_r:unconfined_service_t:s0 key=(null)ARCH=x86_64 SYSCALL=sendto AUID="unset" UID="root" GID="root" EUID="root" SUID="root" FSUID="root" Egid="root" SGID="root" FSGID="root"
type=PROCTITLE msg=audit(1697224687.726:7): proctitle=2F7362696E2F617564697463746C002D52002F6574632F61756469742F61756469742E72756C6573

```

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 (Рисунок 3.15).



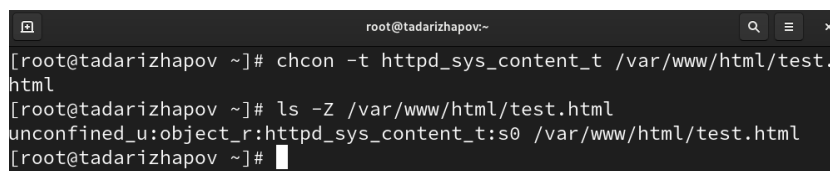
```

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

```

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” (Рисунок 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” (Рисунок 3.17).



```

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

```

Figure 3.16: Возвращение исходного контекста файлу

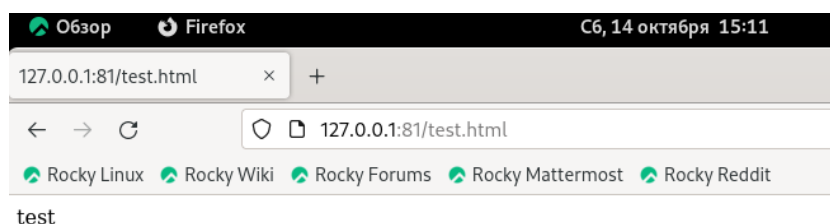


Figure 3.17: Обращение к файлу через веб-сервер

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(Рисунок 3.18).

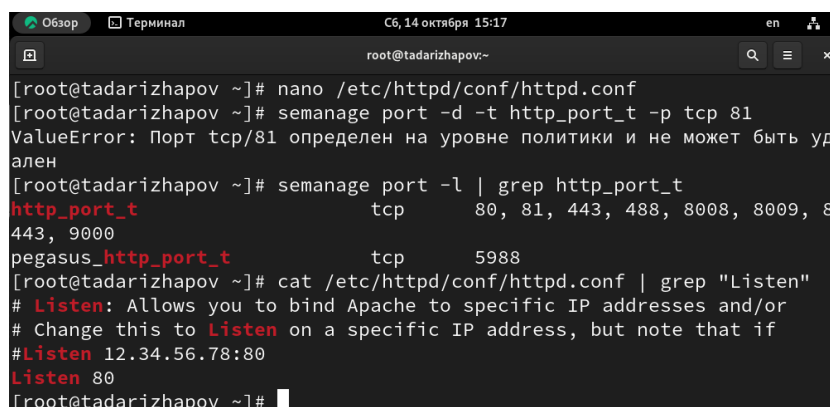


Figure 3.18: Возвращение Listen 80 и попытка удалить порт 81

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



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>.