

Отчёта по лабораторной работе №6

Мандатное разграничение прав в Linux

Кармацкий Никита Сергеевич

Содержание

1	Цель работы	5
2	Теоретическое введение	6
3	Выполнение лабораторной работы	8
4	Выводы	19

Список иллюстраций

3.1	проверка режима работы SELinux	8
3.2	Проверка работы Apache	9
3.3	Контекст безопасности Apache	9
3.4	Состояние переключателей SELinux	10
3.5	Статистика по политике	10
3.6	Типы поддиректорий	11
3.7	Типы файлов	11
3.8	Создание файла	11
3.9	Контекст файла	12
3.10	Отображение файла	12
3.11	Изучение справки по команде	13
3.12	Изменение контекста	13
3.13	Отображение файла	14
3.14	Попытка прочесть лог-файл	14
3.15	Изменение файла	14
3.16	Изменение порта	15
3.17	Попытка прослушивания другого порта	15
3.18	Проверка лог-файлов	16
3.19	Проверка лог-файлов	16
3.20	Проверка портов	16
3.21	Перезапуск сервера	17
3.22	Проверка сервера	17
3.23	Проверка порта 81	18
3.24	Удаление файла	18

Список таблиц

1 Цель работы

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

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 [f?].

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 [s?].

3 Выполнение лабораторной работы

1. Вошли в систему под своей учетной записью. Убедились, что SELinux работает в режиме enforcing политики targeted с помощью команд `getenforce` и `sestatus` (рис. 3.1).

```
[nskarmatskiy@nskarmatskiy ~]$ getenforce
Enforcing
[nskarmatskiy@nskarmatskiy ~]$ 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
```

Рис. 3.1: проверка режима работы SELinux

2. Запускаем сервер `apache`, далее обращаемся с помощью браузера к веб-серверу, запущенному на компьютере, он работает, что видно из вывода команды `service httpd status` (рис. 3.2).


```
[nskarmatskiy@nskarmatskiy ~]$ service httpd status
Redirecting 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 Sun 2024-09-01 17:07:26 MSK; 18min ago
     Docs: man:httpd.service(8)
   Main PID: 54443 (httpd)
   Status: "Total requests: 0; Idle/Busy workers 100/0; Requests/sec: 0; Bytes served: 0"
     Tasks: 177 (limit: 11741)
   Memory: 21.9M
     CPU: 2.013s
   CGroup: /system.slice/httpd.service
           └─54443 /usr/sbin/httpd -DFOREGROUND
             └─54444 /usr/sbin/httpd -DFOREGROUND
               └─54445 /usr/sbin/httpd -DFOREGROUND
                 └─54446 /usr/sbin/httpd -DFOREGROUND
                   └─54447 /usr/sbin/httpd -DFOREGROUND
```

Рис. 3.2: Проверка работы Apache

3. С помощью команды `ps auxZ | grep httpd` найдем веб-сервер Apache в списке процессов. Его контекст безопасности - `httpd_t` (рис. 3.3).

```
[nskarmatskiy@nskarmatskiy ~]$ ps auxZ | grep httpd
system_u:system_r:httpd_t:s0 root 54443 0.0 0.5 29652 10044 ?
Ss 17:06 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 54444 0.0 0.4 31768 8500 ?
S 17:07 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 54445 0.0 0.6 1453120 11668 ?
Sl 17:07 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 54446 0.0 0.6 1453120 11764 ?
Sl 17:07 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 54447 0.0 0.7 1585216 14136 ?
Sl 17:07 0:00 /usr/sbin/httpd -DFOREGROUND
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 nskarma+ 55016 0.0 0.1 22
1456 2048 pts/11 S+ 17:26 0:00 grep --color=auto httpd
```

Рис. 3.3: Контекст безопасности Apache

4. Просмотрим текущее состояние переключателей SELinux для Apache с помощью команды `sestatus -bigrep httpd` (рис. 3.4).

```

samba_export_all_rw      off
samba_load_libgfapi      off
samba_portmapper        off
samba_run_unconfined     off
samba_share_fusefs       off
samba_share_nfs          off
sanlock_enable_home_dirs off
sanlock_use_fusefs       off
sanlock_use_nfs          off
sanlock_use_samba        off
saslauthd_read_shadow    off
screen_allow_session_sharing off
secadm_exec_content      on
secure_mode              off
secure_mode_insmode      off
secure_mode_policyload   off
selinuxuser_direct_dri_enabled on
selinuxuser_execheap     off
selinuxuser_execmod      off
selinuxuser_execstack    on
selinuxuser_mysql_connect_enabled off
selinuxuser_ping         on
selinuxuser_postgresql_connect_enabled off
selinuxuser_rw_noexecattrfile on

```

Рис. 3.4: Состояние переключателей SELinux

5. Просмотрим статистику по политике с помощью команды `seinfo`. Множество пользователей - 8, ролей - 39, типов - 5135. (рис. 3.5).

```

[nskarmatskiy@nskarmatskiy ~]$ 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:                   5145
Attributes:              259
Users:                   8
Roles:                   15
Booleans:                356
Cond. Expr.:             388
Allow:                   65500
Neverallow:              0
Auditallow:              176
Dontaudit:               8682
Type_trans:              271770
Type_change:             94
Type_member:             37
Range_trans:             5931

```

Рис. 3.5: Статистика по политике

6. 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. 3.6).

```
[nskarmatskiy@nskarmatskiy ~]$ ls -lZ /var/www/
итого 0
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_script_exec_t:s0 6 апр  8 19:30 cgi-bin
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_content_t:s0      6 апр  8 19:30 html
```

Рис. 3.6: Типы поддиректорий

7. В директории `/var/www/html` нет файлов. (рис. 3.7).

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

Рис. 3.7: Типы файлов

8. Создать файл может только суперпользователь, поэтому от его имени создаем файл `touch.html` со следующим содержанием:

```
<html>
<body>test</body>
</html>
```

(рис. 3.8).

```
[root@nskarmatskiy ~]# touch /var/www/html/
[root@nskarmatskiy ~]# ls /var/www/html/
[root@nskarmatskiy ~]# touch /var/www/html/test.html
[root@nskarmatskiy ~]# nano /var/www/html/test.html
[root@nskarmatskiy ~]# cat /var/www/html/test.html
<html>
<body>test</body>
</html>
[root@nskarmatskiy ~]# exit
```

Рис. 3.8: Создание файла

9. Проверяем контекст созданного файла. По умолчанию это `httpd_sys_content_t` (рис. 3.9).

```
[nskarmatskiy@nskarmatskiy ~]$ ls -lZ /var/www/html/
итого 4
-rw-r--r--. 1 root root unconfined_u:object_r:httpd_sys_content_t:s0 33 сен 1 17:31 test.html
```

Рис. 3.9: Контекст файла

10. Обращаемся к файлу через веб-сервер, введя в браузере адрес `http://127.0.0.1/test.html`. Файл был успешно отображён (рис. 3.10).

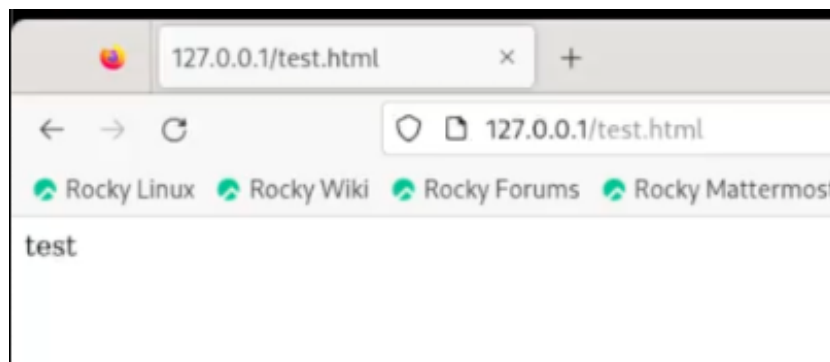


Рис. 3.10: Отображение файла

11. Изучим 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. 3.11).

```
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-sto
    [ -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 sho
    apachectl on Unix-based systems or as a service on Windows NT, 2000 an
    Manual page httpd(8)  Line 1 /usr/share/doc/httpd-2.4.18-14/
```

Рис. 3.11: Изучение справки по команде

12. Изменяем 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. 3.12).

```
[nskarmatskiy@nskarmatskiy ~]$ sudo chcon -t samba_share_t /var/www/html/test.html
[nskarmatskiy@nskarmatskiy ~]$ ls -Z /var/
account/ crash/ ftp/ lib/ log/ opt/ spool/ www/
adm/ db/ games/ local/ mail/ preserve/ tmp/ yp/
cache/ empty/ kerberos/ lock/ nis/ run/ .updated
[nskarmatskiy@nskarmatskiy ~]$ ls -Z /var/www/html/test.html
unconfined_u:object_r:samba_share_t:s0 /var/www/html/test.html
```

Рис. 3.12: Изменение контекста

13. При попытке отображения файла в браузере получаем сообщение об ошибке (рис. 3.13).

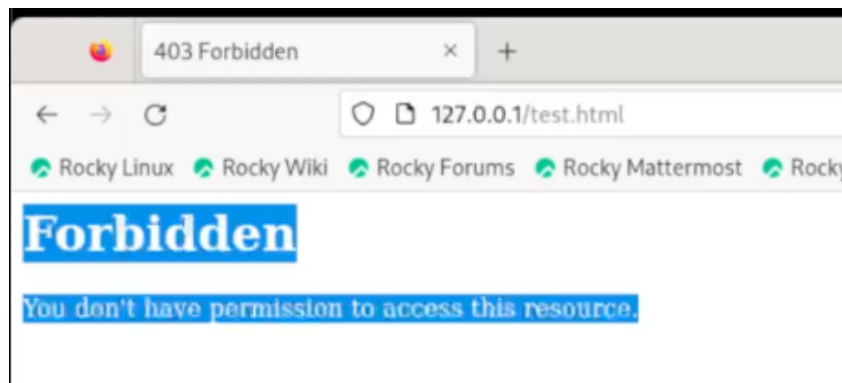


Рис. 3.13: Отображение файла

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.

14. Просматриваем log-файлы веб-сервера Apache и системный лог-файл: `tail /var/log/messages`. Если в системе окажутся запущенными процессы `setroubleshootd` и `audtd`, то вы также сможете увидеть ошибки, аналогичные указанным выше, в файле `/var/log/audit/audit.log`. (рис. 3.14).

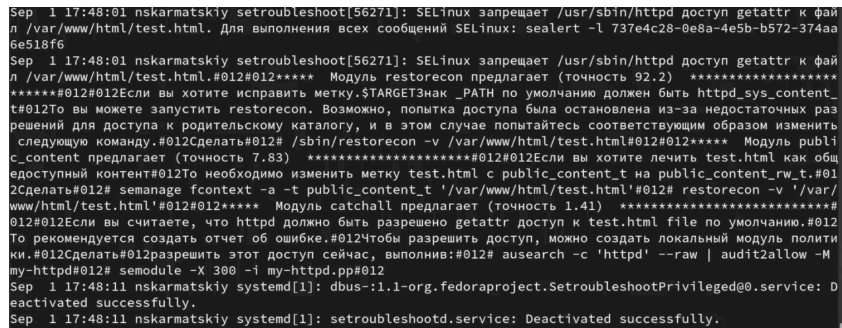
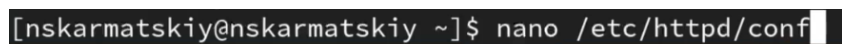
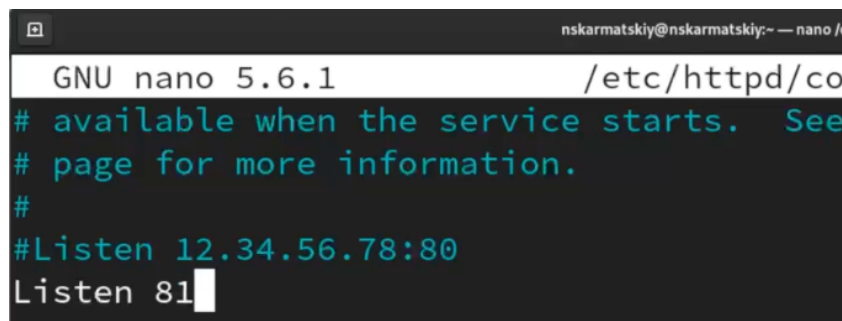


Рис. 3.14: Попытка прочесть лог-файл

15. Чтобы запустить веб-сервер Apache на прослушивание TCP-порта 81 (а не 80, как рекомендует IANA и прописано в `/etc/services`) открываем файл `/etc/httpd/httpd.conf` для изменения. (рис. 3.15).



16. Находим строчку Listen 80 и заменяем её на Listen 81. (рис. 3.16).



```
nskarmatskiy@nskarmatskiy:~ — nano /etc/httpd/conf/httpd.conf
GNU nano 5.6.1
# available when the service starts. See
# page for more information.
#
#Listen 12.34.56.78:80
Listen 81
```

Рис. 3.16: Изменение порта

17. Выполняем перезапуск веб-сервера Apache. Произошёл сбой, потому что порт 80 для локальной сети, а 81 нет (рис. 3.17).

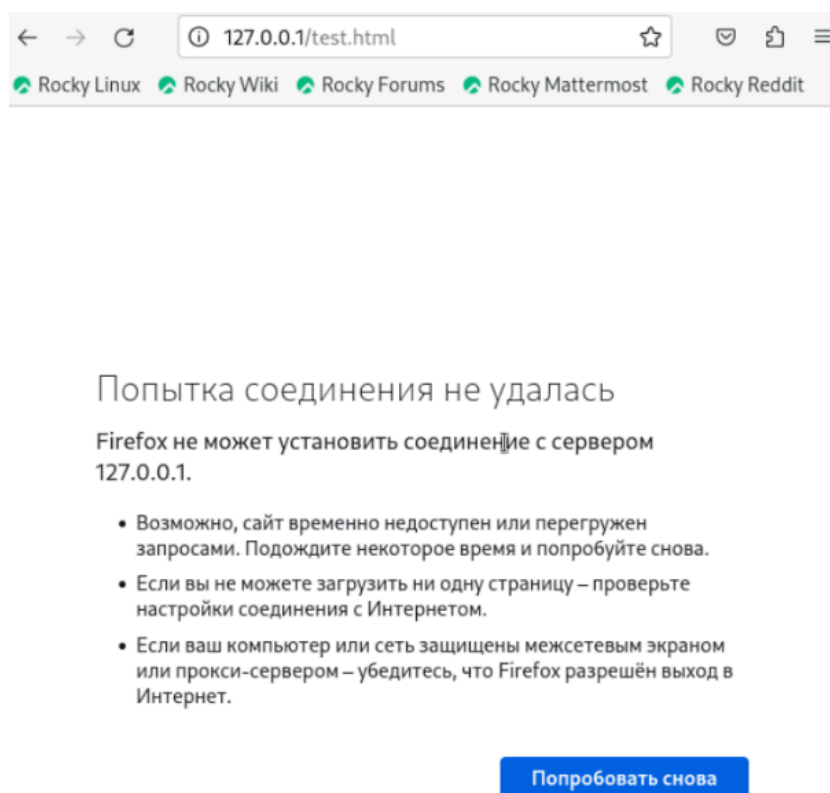


Рис. 3.17: Попытка прослушивания другого порта

18. Проанализируем лог-файлы: `tail -nl /var/log/messages` (рис. 3.18).

```

Sep  1 17:52:14 nskarmatskiy systemd[1]: httpd.service: Consumed 4.275s CPU
time.
Sep  1 17:52:14 nskarmatskiy systemd[1]: Starting The Apache HTTP Server...
Sep  1 17:52:14 nskarmatskiy systemd[1]: Started The Apache HTTP Server.
Sep  1 17:52:14 nskarmatskiy httpd[56496]: Server configured, listening on:
port 81
[nskarmatskiy@nskarmatskiy ~]$

```

Рис. 3.18: Проверка лог-файлов

19. Просмотрим файлы `/var/log/http/error_log`, `/var/log/http/access_log` и `/var/log/audit/audit.log` и выясним, в каких файлах появились записи. Запись появилась в файле `error_log` (рис. 3.19).

```

addr=? terminal=/dev/pts/11 res=success'UID="nskarmatskiy" AUID="guest"
type=USER_ACCT msg=audit(1725202436.794:965): pid=56766 uid=1000 auid=1001 s
es=6 subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 msg='op=PAM:
accounting grantors=pam_unix acct="nskarmatskiy" exe="/usr/bin/sudo" hostnam
e=? addr=? terminal=/dev/pts/11 res=success'UID="nskarmatskiy" AUID="guest"
type=USER_CMD msg=audit(1725202436.804:966): pid=56766 uid=1000 auid=1001 se
s=6 subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 msg='cwd="/ho
me/nskarmatskiy" cmd=636174202F661722F6C6F672F61756469742F61756469742E6C6F6
7 exe="/usr/bin/sudo" terminal=pts/11 res=success'UID="nskarmatskiy" AUID="g
uest"
type=CRED_REFR msg=audit(1725202436.804:967): pid=56766 uid=1000 auid=1001 s
es=6 subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 msg='op=PAM:
setcred grantors=pam_env,pam_unix acct="root" exe="/usr/bin/sudo" hostname=?
addr=? terminal=/dev/pts/11 res=success'UID="nskarmatskiy" AUID="guest"
type=USER_START msg=audit(1725202436.814:968): pid=56766 uid=1000 auid=1001
ses=6 subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 msg='op=PAM
:session_open grantors=pam_keyinit,pam_limits,pam_systemd,pam_unix acct="roo
t" exe="/usr/bin/sudo" hostname=? addr=? terminal=/dev/pts/11 res=success'UI
D="nskarmatskiy" AUID="guest"
[nskarmatskiy@nskarmatskiy ~]$

```

Рис. 3.19: Проверка лог-файлов

20. Выполняем команду `semanage port -a -t http_port_t -p tcp 81` После этого проверяем список портов командой `semanage port -l | grep http_port_t` Порт 81 появился в списке (рис. 3.20).

```

[nskarmatskiy@nskarmatskiy ~]$ sudo semanage port -a -t http_port_t -p tcp 81
Port tcp/81 already defined, modifying instead
[nskarmatskiy@nskarmatskiy ~]$ semanage port -l | grep http_port_t
semanage port: error: one of the arguments -a/--add -d/--delete -m/--modify -l/--list -E/--extract -t
/--deleteall is required
[nskarmatskiy@nskarmatskiy ~]$ semanage port -l | grep http_port_t
ValueError: Политика SELinux не задана, или нет доступа к хранилищу.
[nskarmatskiy@nskarmatskiy ~]$ sudo semanage port -l | grep http_port_t
[nskarmatskiy@nskarmatskiy ~]$ sudo semanage port -l | grep http_port_t
http_port_t      tcp      81, 80, 81, 443, 488, 8008, 8009, 8443, 9000
pegasus_http_port_t  tcp      5988
[nskarmatskiy@nskarmatskiy ~]$

```

Рис. 3.20: Проверка портов

21. Перезапускаем сервер Apache (рис. 3.21).


```
[nskarmatskiy@nskarmatskiy ~]$ systemctl restart httpd
[nskarmatskiy@nskarmatskiy ~]$ systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
   Active: active (running) since Sun 2024-09-01 18:03:10 MSK; 1min 52s ago
     Docs: man:httpd.service(8)
   Main PID: 57556 (httpd)
    Status: "Total requests: 0; Idle/Busy workers 100/0; Requests/sec: 0; Bytes served/sec: 0 B/sec"
     Tasks: 177 (limit: 11741)
    Memory: 27.9M
       CPU: 223ms
    CGroup: /system.slice/httpd.service
            └─57556 /usr/sbin/httpd -DFOREGROUND
              └─57559 /usr/sbin/httpd -DFOREGROUND
                └─57560 /usr/sbin/httpd -DFOREGROUND
                  └─57561 /usr/sbin/httpd -DFOREGROUND
                    └─57562 /usr/sbin/httpd -DFOREGROUND

сен 01 18:03:10 nskarmatskiy.localdomain systemd[1]: Starting The Apache HTTP Server...
сен 01 18:03:10 nskarmatskiy.localdomain httpd[57556]: Server configured, listening on: port 81
сен 01 18:03:10 nskarmatskiy.localdomain systemd[1]: Started The Apache HTTP Server.
```

Рис. 3.21: Перезапуск сервера

22. Теперь он работает, ведь мы внесли порт 81 в список портов `httpd_port_t` (рис. 3.22).

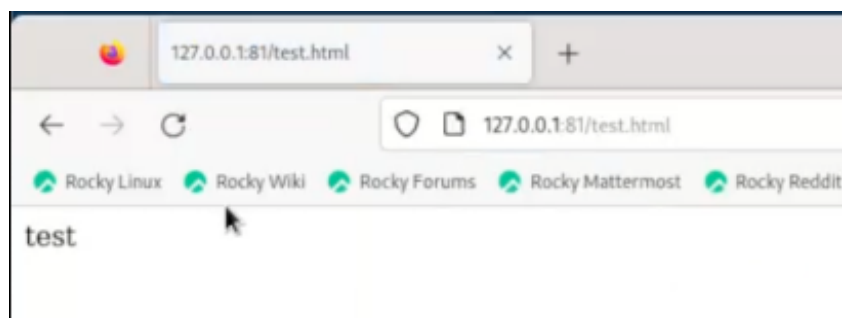
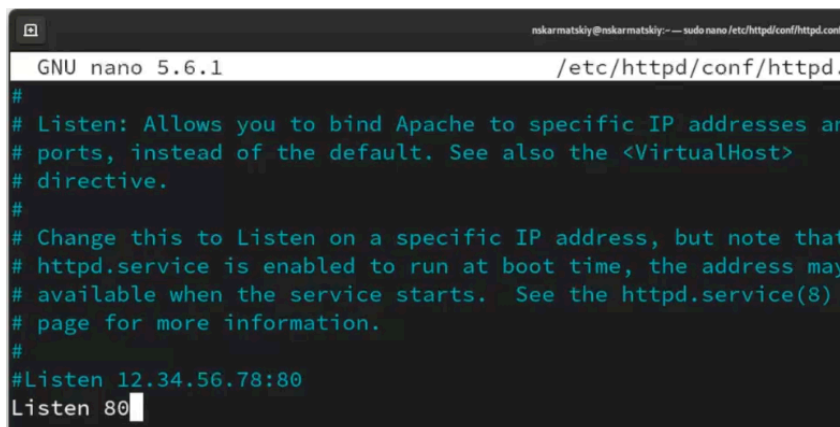


Рис. 3.22: Проверка сервера

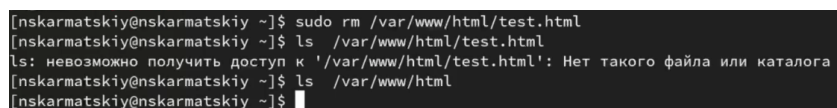
23. Возвращаем в файл `/etc/httpd/httpd.conf` порт 80, вместо 81. Проверяем, что порт 81 удален, это правда. (рис. 3.23).



```
nskarmatskiy@nskarmatskiy:~$ sudo nano /etc/httpd/conf/httpd.conf
GNU nano 5.6.1 /etc/httpd/conf/httpd.conf
#
# Listen: Allows you to bind Apache to specific IP addresses and
# ports, instead of the default. See also the <VirtualHost>
# directive.
#
# Change this to Listen on a specific IP address, but note that
# httpd.service is enabled to run at boot time, the address may
# not be available when the service starts. See the httpd.service(8)
# page for more information.
#
#Listen 12.34.56.78:80
Listen 80
```

Рис. 3.23: Проверка порта 81

24. Далее удаляем файл test.html, проверяем, что он удален(рис. 3.24).



```
[nskarmatskiy@nskarmatskiy ~]$ sudo rm /var/www/html/test.html
[nskarmatskiy@nskarmatskiy ~]$ ls /var/www/html/test.html
ls: невозможно получить доступ к '/var/www/html/test.html': Нет такого файла или каталога
[nskarmatskiy@nskarmatskiy ~]$ ls /var/www/html
[nskarmatskiy@nskarmatskiy ~]$
```

Рис. 3.24: 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.