

Лабораторная работа №6

Дисциплина: Информационная безопасность

Губина Ольга Вячеславовна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	9
4.1	Подготовительный этап	9
4.2	Порядок выполнения работы	12
5	Выводы	26
	Список литературы	27

Список иллюстраций

4.1	Проверка политики и режима	9
4.2	Обновление	10
4.3	Загрузка Apache	10
4.4	Конец загрузки Apache	10
4.5	Расположение конфигурационного файла httpd	11
4.6	Задаем ServerName	11
4.7	Задаем ServerName	12
4.8	Режим enforcing политики targeted	12
4.9	Запуск	13
4.10	Контекст безопасности	13
4.11	Переключатели SELinux для Apache	14
4.12	Статистика по политике	15
4.13	Типы файлов	15
4.14	Типы файлов	16
4.15	Создание html файла	16
4.16	Создание html файла	16
4.17	Создание html файла	17
4.18	Обращение к файлу через веб-сервер	17
4.19	Контекст безопасности html файла	18
4.20	Смена контекста безопасности html файла	18
4.21	Попытка получения доступа к файлу через веб-сервер	19
4.22	tail /var/log/messages	19
4.23	/var/log/audit/audit.log	20
4.24	Смена прослушиваемого порта	20
4.25	Перезапуск Apache	21
4.26	tail -nl /var/log/messages	21
4.27	/var/log/http/error_log	21
4.28	/var/log/http/access_log	22
4.29	/var/log/audit/audit.log	22
4.30	Список портов	22
4.31	Запуск веб-сервера	23
4.32	Возвращаем контекст безопасности	23
4.33	Получение доступа к файлу через веб-сервер	24
4.34	Изменение прослушиваемого порта	24
4.35	Удаление файла	25

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

- Изучить на практике работу SELinux и Apache.

3 Теоретическое введение

SELinux (SELinux) — это система принудительного контроля доступа, реализованная на уровне ядра. Впервые эта система появилась в четвертой версии CentOS, а в 5 и 6 версии реализация была существенно дополнена и улучшена. Эти улучшения позволили SELinux стать универсальной системой, способной эффективно решать массу актуальных задач. Стоит помнить, что классическая система прав Unix применяется первой, и управление перейдет к SELinux только в том случае, если эта первичная проверка будет успешно пройдена.

Для того, чтобы понять, в чем состоит практическая ценность SELinux, рассмотрим несколько примеров, когда стандартная система контроля доступа недостаточна. Если SELinux отключен, то вам доступна только классическая дискреционная система контроля доступа, которая включает в себя DAC (избирательное управление доступом) или ACL(списки контроля доступа). То есть речь идет о манипулировании правами на запись, чтение и исполнение на уровне пользователей и групп пользователей, чего в некоторых случаях может быть совершенно недостаточно. Например:

- Администратор не может в полной мере контролировать действия пользователя. Например, пользователь вполне способен дать всем остальным пользователям права на чтение собственных конфиденциальных файлов, таких как ключи SSH.
- Процессы могут изменять настройки безопасности. Например, файлы, содержащие в себе почту пользователя должны быть доступны для чтения

только одному конкретному пользователю, но почтовый клиент вполне может изменить права доступа так, что эти файлы будут доступны для чтения всем.

- Процессы наследуют права пользователя, который их запустил. Например, зараженная трояном версия браузера Firefox в состоянии читать SSH-ключи пользователя, хотя не имеет для того никаких оснований.[1]

Apache – это свободное программное обеспечение для размещения веб-сервера. Он хорошо показывает себя в работе с масштабными проектами, поэтому заслуженно считается одним из самых популярных веб-серверов. Кроме того, Apache очень гибок в плане настройки, что даёт возможность реализовать все особенности размещаемого веб-ресурса.[2]

Установить веб-сервер Apache можно следующим образом. Откройте окно терминала и обновите списки пакетов репозитория, введя следующее: `sudo yum update`

Теперь вы можете установить Apache с помощью команды: `sudo yum -y install httpd`

httpd - это имя службы Apache в CentOS. Опция `-y` автоматически отвечает да на запрос подтверждения.[3]

4 Выполнение лабораторной работы

4.1 Подготовительный этап

Сперва проверим конфигурационный файл SELinux - видим, что политика targeted и режим enforcing используются в данном дистрибутиве по умолчанию, т.е. каких-то специальных настроек не требуется (рис. 4.1).

```
(root@ovgubina ~) cat /etc/selinux/config
# This file controls the state of SELinux on the system.
# SELinux can take one of three values:
#   enforcing - SELinux security policy is enforced.
#   permissive - SELinux prints warnings instead of enforcing.
#   disabled - No SELinux policy is loaded.
# See also:
#   https://docs.fedoraproject.org/en-US/quick-docs/getting-started-with-selinux/#getting-started-with-selinux-states-and-modes
#
# NOTE: In earlier Fedora kernel builds, SELINUX=disabled would also
#       fully disable SELinux during boot. If you need a system with SELinux
#       fully disabled instead of SELinux running with no policy loaded, you
#       need to pass selinux=0 to the kernel command line. You can use grubby
#       to persistently set the bootloader to boot with selinux=0:
#
#       grubby --update-kernel ALL --args selinux=0
#
# To revert back to SELinux enabled:
#
#       grubby --update-kernel ALL --remove-args selinux
#
SELINUX=enforcing
# SELINUXTYPE can take one of three values:
#   targeted - Targeted processes are protected,
#   minimum - Modification of targeted policy. Only selected processes are protected.
#   mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

Рис. 4.1: Проверка политики и режима

Дальше потребуется установить менеджер Apache, для этого предварительно обновим систему (рис. 4.2), только после этого устанавливаем Apache (httpd) (рис. 4.3-4.4).

```

root@ovgubina ~# yum update
Rocky Linux 9 - BaseOS                               9.9 kB/s | 4.1 kB  00:00
Rocky Linux 9 - BaseOS                               1.4 MB/s | 1.9 MB  00:01
Rocky Linux 9 - AppStream                             12 kB/s | 4.5 kB  00:00
Rocky Linux 9 - AppStream                             4.6 MB/s | 7.1 MB  00:01
Rocky Linux 9 - Extras                               7.6 kB/s | 2.9 kB  00:00
Dependencies resolved.
=====
Package                                Architecture      Version            Repository          Size
=====
Upgrading:
firefox                                x86_64            115.3.1-1.el9_2    appstream           110 M
ghostscript                            x86_64            9.54.0-10.el9_2    appstream           36 k
ghostscript-tools-fonts                x86_64            9.54.0-10.el9_2    appstream           11 k
ghostscript-tools-printing              x86_64            9.54.0-10.el9_2    appstream           11 k
glibc                                  x86_64            2.34-60.el9_2.7    baseos              1.9 M
glibc-all-langpacks                   x86_64            2.34-60.el9_2.7    baseos              18 M
glibc-common                           x86_64            2.34-60.el9_2.7    baseos              386 k
glibc-devel                             x86_64            2.34-60.el9_2.7    appstream           46 k

```

Рис. 4.2: Обновление

```

root@ovgubina ~# yum install httpd
Last metadata expiration check: 0:01:36 ago on Mon 09 Oct 2023 07:06:55 PM MSK.
Dependencies resolved.
=====
Package                                Architecture      Version            Repository          Size
=====
Installing:
httpd                                  x86_64            2.4.53-11.el9_2.5  appstream           47 k
Installing dependencies:
apr                                    x86_64            1.7.0-11.el9       appstream           123 k
apr-util                              x86_64            1.6.1-20.el9_2.1   appstream           94 k
apr-util-bdb                          x86_64            1.6.1-20.el9_2.1   appstream           12 k
httpd-core                            x86_64            2.4.53-11.el9_2.5  appstream           1.4 M
httpd-filesystem                      noarch            2.4.53-11.el9_2.5  appstream           19 k
httpd-tools                           x86_64            2.4.53-11.el9_2.5  appstream           81 k
rocky-logos-httpd                     noarch            98.14-1.el9        appstream           24 k
Installing weak dependencies:
apr-util-openssl                      x86_64            1.6.1-20.el9_2.1   appstream           14 k
mod_http2                             x86_64            1.15.19-4.el9_2.4  appstream           149 k
mod_lua                               x86_64            2.4.53-11.el9_2.5  appstream           61 k
Transaction Summary
=====
Install 11 Packages

```

Рис. 4.3: Загрузка Apache

```

Verifying      : apr-util-bdb-1.6.1-20.el9_2.1.x86_64      7/11
Verifying      : apr-util-1.6.1-20.el9_2.1.x86_64         8/11
Verifying      : mod_http2-1.15.19-4.el9_2.4.x86_64       9/11
Verifying      : apr-1.7.0-11.el9.x86_64                 10/11
Verifying      : httpd-core-2.4.53-11.el9_2.5.x86_64      11/11
Installed:
apr-1.7.0-11.el9.x86_64      apr-util-1.6.1-20.el9_2.1.x86_64      apr-util-bdb-1.6.1-20.el9_2.1.x86_64      apr-util-openssl-1.6.1-20.el9_2.1.x86_64
httpd-2.4.53-11.el9_2.5.x86_64      httpd-core-2.4.53-11.el9_2.5.x86_64      httpd-filesystem-2.4.53-11.el9_2.5.noarch      httpd-tools-2.4.53-11.el9_2.5.x86_64
mod_http2-1.15.19-4.el9_2.4.x86_64      mod_lua-2.4.53-11.el9_2.5.x86_64      rocky-logos-httpd-98.14-1.el9.noarch
Complete!

```

Рис. 4.4: Конец загрузки Apache

Далее зададим `ServerName test.ru` в конфигурационном файле `httpd` (рис. 4.6), для этого сперва найдем, где он находится (рис. 4.5).

```

[root@ovgubina ~]# ls /etc | grep httpd
httpd
[root@ovgubina ~]# ls /etc/httpd
conf conf.d conf.modules.d logs modules run state
[root@ovgubina ~]# ls /etc/httpd/conf
httpd.conf magic
[root@ovgubina ~]# cat /etc/httpd/conf/httpd.conf
#
# This is the main Apache HTTP server configuration file. It contains the
# configuration directives that give the server its instructions.
# See <URL:http://httpd.apache.org/docs/2.4/> for detailed information.
# In particular, see
# <URL:http://httpd.apache.org/docs/2.4/mod/directives.html>
# for a discussion of each configuration directive.
#
# See the httpd.conf(5) man page for more information on this configuration,
# and httpd.service(8) on using and configuring the httpd service.
#
# Do NOT simply read the instructions in here without understanding
# what they do. They're here only as hints or reminders. If you are unsure
# consult the online docs. You have been warned.
#
# Configuration and logfile names: If the filenames you specify for many
# of the server's control files begin with "/" (or "drive:/" for Win32), the
# server will use that explicit path. If the filenames do *not* begin
# with "/", the value of ServerRoot is prepended -- so 'log/access_log'
# with ServerRoot set to '/www' will be interpreted by the
# server as '/www/log/access_log', where as '/log/access_log' will be
# interpreted as '/log/access_log'.

```

Рис. 4.5: Расположение конфигурационного файла httpd

```

root@ovgubina:~
#
ServerAdmin root@localhost
#
# ServerName gives the name and port that the server uses to identify itself.
# This can often be determined automatically, but we recommend you specify
# it explicitly to prevent problems during startup.
#
# If your host doesn't have a registered DNS name, enter its IP address here.
#
#ServerName www.example.com:80
ServerName test.ru
#
# Deny access to the entirety of your server's filesystem. You must
# explicitly permit access to web content directories in other
# <Directory> blocks below.
#
<Directory />
    AllowOverride none
    Require all denied
</Directory>
#
# Note that from this point forward you must specifically allow
# particular features to be enabled - so if something's not working as
# you might expect, make sure that you have specifically enabled it
# below.
#
# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this directory, but
:~wq

```

Рис. 4.6: Задаем ServerName

Чтобы пакетный фильтр в своей рабочей конфигурации позволял подклю-

чаться к 80-у и 81-у портам протокола tcp добавим разрешающие правила (рис. 4.7):

```
[root@ovgubina ~]#  
[root@ovgubina ~]# iptables -I INPUT -p tcp --dport 80 -j ACCEPT  
[root@ovgubina ~]# iptables -I INPUT -p tcp --dport 81 -j ACCEPT  
[root@ovgubina ~]# iptables -I OUTPUT -p tcp --sport 80 -j ACCEPT  
[root@ovgubina ~]# iptables -I OUTPUT -p tcp --sport 81 -j ACCEPT  
[root@ovgubina ~]#
```

Рис. 4.7: Задаем ServerName

4.2 Порядок выполнения работы

1. Войдите в систему с полученными учётными данными и убедитесь, что SELinux работает в режиме enforcing политики targeted с помощью команд `getenforce` и `sestatus` (рис. 4.8).

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

Рис. 4.8: Режим enforcing политики targeted

2. Обратитесь с помощью браузера к веб-серверу, запущенному на вашем компьютере, и убедитесь, что последний работает: `service httpd status`. Видим, что он неактивен, поэтому запускаем его командой `service httpd start`, после чего снова проверяем, в этот раз сервис активен (рис. 4.9).

```
[ovgubina@ovgubina ~]$ service httpd status
Redirecting to /bin/systemctl status httpd.service
o httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
   Active: inactive (dead)
   Docs: man:httpd.service(8)
[ovgubina@ovgubina ~]$ service httpd start
Redirecting to /bin/systemctl start httpd.service
[ovgubina@ovgubina ~]$ 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 Mon 2023-10-09 19:37:46 MSK; 7s ago
     Docs: man:httpd.service(8)
  Main PID: 6096 (httpd)
    Status: "Started, listening on: port 80"
     Tasks: 213 (limit: 12145)
    Memory: 27.5M
       CPU: 191ms
    CGroup: /system.slice/httpd.service
            └─6096 /usr/sbin/httpd -DFOREGROUND
              └─6105 /usr/sbin/httpd -DFOREGROUND
                └─6106 /usr/sbin/httpd -DFOREGROUND
                  └─6110 /usr/sbin/httpd -DFOREGROUND
                    └─6111 /usr/sbin/httpd -DFOREGROUND

Oct 09 19:37:46 ovgubina.localdomain systemd[1]: Starting The Apache HTTP Server...
Oct 09 19:37:46 ovgubina.localdomain httpd[6096]: Server configured, listening on: port 80
Oct 09 19:37:46 ovgubina.localdomain systemd[1]: Started The Apache HTTP Server.
[ovgubina@ovgubina ~]$
```

Рис. 4.9: Запуск

- Найдите веб-сервер Apache в списке процессов, определите его контекст безопасности и занесите эту информацию в отчёт. Например, можно использовать команду `ps auxZ | grep httpd` (4.10).

```
[ovgubina@ovgubina ~]$ ps auxZ | grep httpd
system_u:system_r:httpd_t:s0 root        6096  0.3  0.5 20116 11344 ?        Ss   19:37   0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache    6105  0.0  0.3 21608 7344 ?        S    19:37   0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache    6106  0.0  0.5 1538124 11004 ?      Sl   19:37   0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache    6110  0.0  0.6 1669200 13052 ?      Sl   19:37   0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache    6111  0.0  0.5 1538124 11004 ?      Sl   19:37   0:00 /usr/sbin/httpd -DFOREGROUND
unconfined_u:unconfined_r:unconfined_t:s0-c0.c1023 ovgubina 6365  0.0  0.1 221664 2268 pts/0 S+  19:38   0:00 grep --color=auto httpd
[ovgubina@ovgubina ~]$ ps -eZ | grep httpd
system_u:system_r:httpd_t:s0 6096 ? 00:00:00 httpd
system_u:system_r:httpd_t:s0 6105 ? 00:00:00 httpd
system_u:system_r:httpd_t:s0 6106 ? 00:00:00 httpd
system_u:system_r:httpd_t:s0 6110 ? 00:00:00 httpd
system_u:system_r:httpd_t:s0 6111 ? 00:00:00 httpd
[ovgubina@ovgubina ~]$
```

Рис. 4.10: Контекст безопасности

Видим, что веб-сервер имеет контекст безопасности `httpd_t`.

- Посмотрите текущее состояние переключателей SELinux для Apache с помощью команды `sestatus -b httpd` (рис. 4.11). Многие из них находятся в положении «off».

```
[ovgubina@ovgubina ~]$ 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
cdrecord_read_content off
cluster_can_network_connect off
cluster_manage_all_files off
cluster_use_execmem off
cobbler_anon_write off
cobbler_can_network_connect off
cobbler_use_cifs off
```

Рис. 4.11: Переключатели SELinux для Apache

5. Посмотрите статистику по политике с помощью команды `seinfo`, также определите множество пользователей, ролей, типов (рис. 4.12).

```
[ovgubina@ovgubina ~]$ 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:            65008    Neverallow:         0
Auditallow:       170     Dontaudit:          8572
Type_trans:       265344  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

[ovgubina@ovgubina ~]$
```

Рис. 4.12: Статистика по политике

Число типов = 5100, ролей = 14, пользователей = 8.

6. Определите тип файлов и поддиректорий, находящихся в директории /var/www, с помощью команды `ls -lZ /var/www` (рис. 4.13).

```
[ovgubina@ovgubina ~]$ ls -lZ /var/www
total 0
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_script_exec_t:s0 6 May 16 23:21 cgi-bin
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_content_t:s0 6 May 16 23:21 html
```

Рис. 4.13: Типы файлов

В каталоге находятся только директории.

7. Определите тип файлов, находящихся в директории /var/www/html: `ls -lZ /var/www/html` (рис. 4.14).

```
[ovgubina@ovgubina ~]$ ls -lZ /var/www/html
total 0
[ovgubina@ovgubina ~]$ ls /var/www/html
[ovgubina@ovgubina ~]$
```

Рис. 4.14: Типы файлов

Директория пуста.

8. Определите круг пользователей, которым разрешено создание файлов в директории /var/www/html - это только пользователь root.
9. Создайте от имени суперпользователя (так как в дистрибутиве после установки только ему разрешена запись в директорию) html-файл /var/www/html/test.html следующего содержания (рис. 4.15-4.16):

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

```
[root@ovgubina ~]# vim /var/www/html/test.html
[root@ovgubina ~]#
```

Рис. 4.15: Создание html файла

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

Рис. 4.16: Создание html файла

10. Проверьте контекст созданного вами файла. Занесите в отчёт контекст, присваиваемый по умолчанию вновь созданным файлам в директории `/var/www/html` (рис. 4.17).

```
ovgubina@ovgubina ~]$ ls -lZ /var/www/html
total 4
-rw-r--r--. 1 root root unconfined_u:object_r:httpd_sys_content_t:s0 33 Oct  9 19:50 test.html
ovgubina@ovgubina ~]$
```

Рис. 4.17: Создание html файла

Контекст безопасности - **`httpd_sys_content_t`**.

11. Обратитесь к файлу через веб-сервер, введя в браузере адрес `http://127.0.0.1/test.html` (рис. 4.18). Файл был успешно отображён.

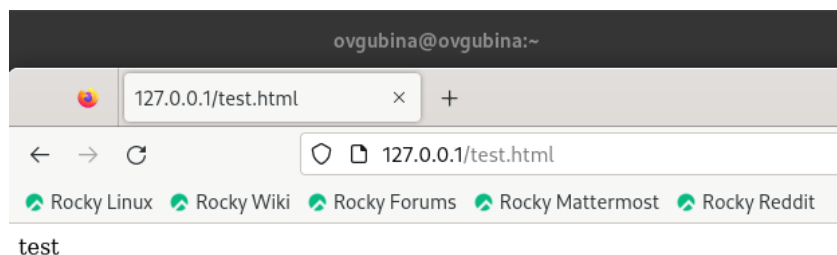


Рис. 4.18: Обращение к файлу через веб-сервер

12. Изучите справку `man httpd_selinux` и выясните, какие контексты файлов определены для `httpd`.

Вызвать данную справку не удалось, контексты безопасности `httpd` были просмотрены через интернет.

Сопоставьте их с типом файла `test.html`. Проверить контекст файла можно командой `ls -Z /var/www/html/test.html` (рис. 4.19). Контекст безопасности

файла - **httpd_sys_content_t**. Данный контекст входит в перечень контекстов безопасности httpd.

Роль **object_r** используется по умолчанию для файлов на «постоянных» носителях и на сетевых файловых системах. 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.

```
[ovgubina@ovgubina ~]$ ls -lZ /var/www/html
total 4
-rw-r--r--. 1 root root unconfined_u:object_r:httpd_sys_content_t:s0 33 Oct  9 19:50 test.html
[ovgubina@ovgubina ~]$
```

Рис. 4.19: Контекст безопасности html файла

13. Измените 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 (рис. 4.20):

```
chcon -t samba_share_t /var/www/html/test.html
ls -Z /var/www/html/test.html
```

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

Рис. 4.20: Смена контекста безопасности html файла

Видим, что контекст безопасности действительно изменился.

14. Попробуйте ещё раз получить доступ к файлу через веб-сервер, введя в браузере адрес <http://127.0.0.1/test.html> (рис. 4.21).

Мы получили сообщение об ошибке: **Forbidden You don't have permission to access /test.html on this server.**

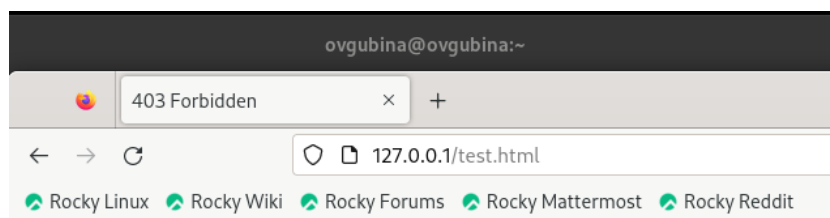
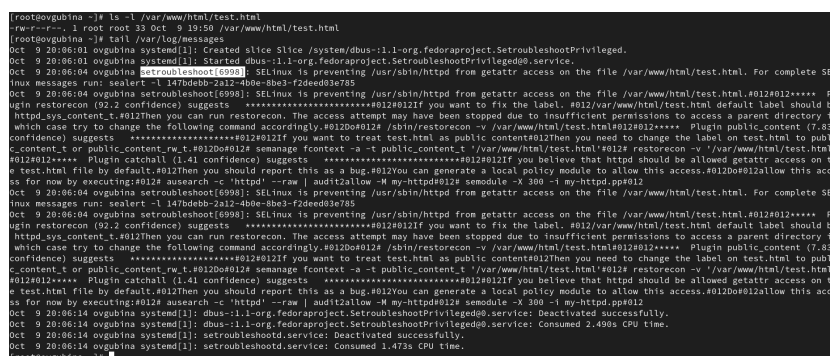


Рис. 4.21: Попытка получения доступа к файлу через веб-сервер

Просмотрите log-файлы веб-сервера Apache. Также просмотрите системный лог-файл: `tail /var/log/messages` (рис. 4.22).



```

root@ovgubina:~# cat /var/log/audit/audit.log
type=DAEMON_START msg=audit(1694100940.504:6778): op=star ver=3.0.7 format=enriched kernel=5.14.0-284.11.1.el9_2.x86_64 audit=4294967295 pid=712 uid=0 ses=4294967295 subj=system_u:system_r:auditd_t:s0 res=successAUID="unset" UID="root"
type=CONFIG_CHANGE msg=audit(1694100940.574:5): op=set audit_backlog_limit=8192 old=64 audit=4294967295 ses=4294967295 subj=system_u:system_r:unconfined_service_t:s0 res=IAUID="unset"
type=SYSCALL msg=audit(1694100940.574:5): arch=c000003e syscall=44 success=yes exit=60 a0=3 a1=fffff09031d90 a2=3c a3=0 items=0 ppid=717 pid=727 audit=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(1694100940.574:5): proctitle=2F7362696E2F617564697463746C002052092F6574632F61756469742F61756469742E72756C6573
type=CONFIG_CHANGE msg=audit(1694100940.575:5): op=set audit_failure=1 old=1 audit=4294967295 ses=4294967295 subj=system_u:system_r:unconfined_service_t:s0 res=IAUID="unset"
type=SYSCALL msg=audit(1694100940.575:5): arch=c000003e syscall=44 success=yes exit=60 a0=3 a1=fffff09031d90 a2=3c a3=0 items=0 ppid=717 pid=727 audit=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(1694100940.575:5): proctitle=2F7362696E2F617564697463746C002052092F6574632F61756469742F61756469742E72756C6573
type=CONFIG_CHANGE msg=audit(1694100940.576:7): op=set audit_backlog_wait_time=60000 old=60000 audit=4294967295 ses=4294967295 subj=system_u:system_r:unconfined_service_t:s0 res=IAUID="unset"
type=SYSCALL msg=audit(1694100940.576:7): arch=c000003e syscall=44 success=yes exit=60 a0=3 a1=fffff09031d90 a2=3c a3=0 items=0 ppid=717 pid=727 audit=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(1694100940.576:7): proctitle=2F7362696E2F617564697463746C002052092F6574632F61756469742F61756469742E72756C6573
type=SERVICE_START msg=audit(1694100940.578:8): pid=1 uid=0 audit=4294967295 ses=4294967295 subj=system_u:system_r:init_t:s0 msg='unit=auditd comm="systemd" exe="/usr/lib/systemd/systemd" hostname=? addr=? terminal=? res=successUID="root" AUID="unset"
type=SYSTEM_BOOT msg=audit(1694100940.592:9): pid=734 uid=0 audit=4294967295 ses=4294967295 subj=system_u:system_r:init_t:s0 msg=' comm="systemd-update-utmp" exe="/usr/lib/systemd/systemd-update-utmp" hostname=? addr=? terminal=? res=successUID="root" AUID="unset"
type=SERVICE_START msg=audit(1694100940.602:10): pid=1 uid=0 audit=4294967295 ses=4294967295 subj=system_u:system_r:init_t:s0 msg='unit=systemd-update-utmp comm="systemd" exe="/usr/lib/systemd/systemd" hostname=? addr=? terminal=? res=successUID="root" AUID="unset"

```

Рис. 4.23: /var/log/audit/audit.log

16. Попробуйте 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 найдите строчку Listen 80 и замените её на Listen 81 (рис. 4.24).

```

root@ovgubina:~# cat /etc/httpd/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 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
# to be loaded here.
#
# Example:
# LoadModule foo_module modules/mod_foo.so
#
Include conf.modules.d/*.*

#
# If you wish httpd to run as a different user or group, you must run
# httpd as root initially and it will switch.
:~#

```

Рис. 4.24: Смена прослушиваемого порта

17. Выполните перезапуск веб-сервера Apache (рис. 4.25). Произошёл сбой?
Поясните почему?

```
[root@ovgubina ~]# service httpd restart
Redirecting to /bin/systemctl restart httpd.service
[root@ovgubina ~]#
```

Рис. 4.25: Перезапуск Apache

Никакой ошибки не возникает, поскольку в изначальных настройках системы порт 81 уже был прописан в рекомендациях системы.

18. Проанализируйте лог-файлы: `tail -nl /var/log/messages` - нет никаких ошибок (рис. 4.26). Просмотрите файлы `/var/log/http/error_log` (рис. 4.27), `/var/log/http/access_log` (рис. 4.28) и `/var/log/audit/audit.log` (рис. 4.29) и выясните, в каких файлах появились записи - нет записей об ошибках, т.к. нет ошибок.

```
[root@ovgubina ~]# tail /var/log/messages
Oct  9 20:14:08 ovgubina systemd[1]: Starting The Apache HTTP Server...
Oct  9 20:14:08 ovgubina systemd[1]: Started The Apache HTTP Server.
Oct  9 20:14:08 ovgubina httpd[7156]: Server configured, listening on: port 81
Oct  9 20:14:27 ovgubina systemd[1]: Stopping The Apache HTTP Server...
Oct  9 20:14:28 ovgubina systemd[1]: httpd.service: Deactivated successfully.
Oct  9 20:14:28 ovgubina systemd[1]: Stopped The Apache HTTP Server.
Oct  9 20:14:28 ovgubina systemd[1]: Starting The Apache HTTP Server...
Oct  9 20:14:28 ovgubina httpd[7400]: Server configured, listening on: port 81
Oct  9 20:14:28 ovgubina systemd[1]: Started The Apache HTTP Server.
Oct  9 20:14:34 ovgubina systemd[1]: fprintd.service: Deactivated successfully.
[root@ovgubina ~]#
```

Рис. 4.26: `tail -nl /var/log/messages`

```
[root@ovgubina ~]# cat /var/log/http/error_log
[Mon Oct 09 19:37:46.335070 2023] [core:notice] [pid 6896:tid 6896] SELinux policy enabled; httpd running as context system_u:system_r:httpd_t:s0
[Mon Oct 09 19:37:46.337001 2023] [suexec:notice] [pid 6896:tid 6896] AH02232: suEXEC mechanism enabled (wrapper: /usr/sbin/suexec)
[Mon Oct 09 19:37:46.364554 2023] [lbmethod:heartbeat:notice] [pid 6896:tid 6896] AH02282: No slotmem from mod_heartbeat
[Mon Oct 09 19:37:46.381672 2023] [mpm_event:notice] [pid 6896:tid 6896] AH00489: Apache/2.4.53 (Rocky Linux) configured -- resuming normal operations
[Mon Oct 09 19:37:46.383824 2023] [core:notice] [pid 6896:tid 6896] AH00004: Command Line: '/usr/sbin/httpd -D FOREGROUND'
[Mon Oct 09 20:14:28.204611 2023] [core:error] [pid 6186:tid 6235] (13)Permission denied: [client 127.0.0.1:46850] AH00035: access to /test.html denied (files
system path '/var/www/html/test.html') because search permissions are missing on a component of the path
[Mon Oct 09 20:14:28.381718 2023] [mpm_event:notice] [pid 6896:tid 6896] AH00492: caught SLOWINCH, shutting down gracefully
[Mon Oct 09 20:14:28.472888 2023] [core:notice] [pid 7156:tid 7156] SELinux policy enabled; httpd running as context system_u:system_r:httpd_t:s0
[Mon Oct 09 20:14:28.475792 2023] [suexec:notice] [pid 7156:tid 7156] AH02232: suEXEC mechanism enabled (wrapper: /usr/sbin/suexec)
[Mon Oct 09 20:14:28.497138 2023] [lbmethod:heartbeat:notice] [pid 7156:tid 7156] AH02282: No slotmem from mod_heartbeat
[Mon Oct 09 20:14:28.510043 2023] [mpm_event:notice] [pid 7156:tid 7156] AH00489: Apache/2.4.53 (Rocky Linux) configured -- resuming normal operations
[Mon Oct 09 20:14:28.531003 2023] [core:notice] [pid 7156:tid 7156] AH00004: Command Line: '/usr/sbin/httpd -D FOREGROUND'
[Mon Oct 09 20:14:27.553478 2023] [mpm_event:notice] [pid 7156:tid 7156] AH00492: caught SLOWINCH, shutting down gracefully
[Mon Oct 09 20:14:28.641464 2023] [core:notice] [pid 7400:tid 7400] SELinux policy enabled; httpd running as context system_u:system_r:httpd_t:s0
[Mon Oct 09 20:14:28.642912 2023] [suexec:notice] [pid 7400:tid 7400] AH02232: suEXEC mechanism enabled (wrapper: /usr/sbin/suexec)
[Mon Oct 09 20:14:28.663487 2023] [lbmethod:heartbeat:notice] [pid 7400:tid 7400] AH02282: No slotmem from mod_heartbeat
[Mon Oct 09 20:14:28.676642 2023] [mpm_event:notice] [pid 7400:tid 7400] AH00489: Apache/2.4.53 (Rocky Linux) configured -- resuming normal operations
[Mon Oct 09 20:14:28.676690 2023] [core:notice] [pid 7400:tid 7400] AH00004: Command Line: '/usr/sbin/httpd -D FOREGROUND'
[root@ovgubina ~]#
```

Рис. 4.27: `/var/log/http/error_log`

```

[root@ovgubina ~]# cat /var/log/http/access_log
127.0.0.1 - - [09/Oct/2023:19:54:25 +0300] "GET /test.html HTTP/1.1" 200 33 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0"
127.0.0.1 - - [09/Oct/2023:19:54:25 +0300] "GET /favicon.ico HTTP/1.1" 404 196 "http://127.0.0.1/test.html" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0"
127.0.0.1 - - [09/Oct/2023:20:06:00 +0300] "GET /test.html HTTP/1.1" 403 199 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0"
[root@ovgubina ~]#

```

Рис. 4.28: /var/log/http/access_log

```

[root@ovgubina ~]# cat /var/log/audit/audit.log
type=BAEMON_START msg=audit(1694100940.584:6778): op=start ver=2.0.7 format=enriched kernel=5.14.0-284.11.1.el9_2.x86_64 auid=4294967295 pid=712 uid=0 ses=4294967295 subj=system_u:system_r:auditd_t:s0 res=successAUID="unset" UID="root"
type=CONFIG_CHANGE msg=audit(1694100940.574:5): op=set audit_backlog_l1mit=8192 old=64 auid=4294967295 ses=4294967295 subj=system_u:system_r:unconfined_service_t:s0 res=IAUID="unset"
type=SYSCALL msg=audit(1694100940.574:5): arch=c000003e syscall=44 success=yes exit=60 a0=3 a1=7ffff09831d90 a2=3c a3=0 items=0 ppid=717 pid=727 auid=4294967295 uid=0 gid=0 euid=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(1694100940.574:5): proctitle=2F7362696E2F617564697463746C002D52002F6574632F61756469742F61756469742E72756C6573
type=CONFIG_CHANGE msg=audit(1694100940.575:6): op=set audit_failure=1 old=1 auid=4294967295 ses=4294967295 subj=system_u:system_r:unconfined_service_t:s0 res=IAUID="unset"
type=SYSCALL msg=audit(1694100940.575:6): arch=c000003e syscall=44 success=yes exit=60 a0=3 a1=7ffff09831d90 a2=3c a3=0 items=0 ppid=717 pid=727 auid=4294967295 uid=0 gid=0 euid=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(1694100940.575:6): proctitle=2F7362696E2F617564697463746C002D52002F6574632F61756469742F61756469742E72756C6573
type=CONFIG_CHANGE msg=audit(1694100940.576:7): op=set audit_backlog_wait_time=60000 old=60000 auid=4294967295 ses=4294967295 subj=system_u:system_r:unconfined_service_t:s0 res=IAUID="unset"
type=SYSCALL msg=audit(1694100940.576:7): arch=c000003e syscall=44 success=yes exit=60 a0=3 a1=7ffff09831d90 a2=3c a3=0 items=0 ppid=717 pid=727 auid=4294967295 uid=0 gid=0 euid=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(1694100940.576:7): proctitle=2F7362696E2F617564697463746C002D52002F6574632F61756469742F61756469742E72756C6573
type=SERVICE_START msg=audit(1694100940.578:8): pid=1 uid=0 auid=4294967295 ses=4294967295 subj=system_u:system_r:init_t:s0 msg="unit=auditd comm='systemd' exe='/usr/lib/systemd/systemd' hostname=? addr=? terminal=? res=success"UID="root" AUID="unset"
type=SYSTEM_BOOT msg=audit(1694100940.593:9): pid=734 uid=0 auid=4294967295 ses=4294967295 subj=system_u:system_r:init_t:s0 msg="comm='systemd-update-utmp' exe='/usr/lib/systemd/systemd-update-utmp' hostname=? addr=? terminal=? res=success"UID="root" AUID="unset"
type=SERVICE_START msg=audit(1694100940.602:10): pid=1 uid=0 auid=4294967295 ses=4294967295 subj=system_u:system_r:init_t:s0 msg="unit=systemd-update-utmp comm='systemd' exe='/usr/lib/systemd/systemd' hostname=? addr=? terminal=? res=success"UID="root" AUID="unset"

```

Рис. 4.29: /var/log/audit/audit.log

19. Выполните команду `semanage port -a -t http_port_t -p tcp 81` После этого проверьте список портов командой `semanage port -l | grep http_port_t` (рис. 4.30). Добавление порта не производим, т.к. нам известно, что он и так уже добавлен - сразу смотрим список.

Порт 81 есть в списке.

```

[root@ovgubina ~]# 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@ovgubina ~]#

```

Рис. 4.30: Список портов

20. Попробуйте запустить веб-сервер Apache ещё раз. Поняли ли вы, почему он сейчас запустился, а в предыдущем случае не смог?

```

[root@ovgubina ~]# service httpd restart
Redirecting to /bin/systemctl restart httpd.service
[root@ovgubina ~]# 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 Mon 2023-10-09 20:23:24 MSK; 11s ago
     Docs: man:httpd.service(8)
  Main PID: 7711 (httpd)
    Status: "Total requests: 0; Idle/Busy workers 100/0; Requests/sec: 0; Bytes served/sec: 0 B/sec"
    Tasks: 213 (limit: 12145)
   Memory: 26.9M
      CPU: 123ms
   CGroup: /system.slice/httpd.service
           └─7711 /usr/sbin/httpd -DFOREGROUND
             └─7712 /usr/sbin/httpd -DFOREGROUND
               └─7713 /usr/sbin/httpd -DFOREGROUND
                 └─7714 /usr/sbin/httpd -DFOREGROUND
                   └─7715 /usr/sbin/httpd -DFOREGROUND

Oct 09 20:23:24 ovgubina.localdomain systemd[1]: Starting The Apache HTTP Server...
Oct 09 20:23:24 ovgubina.localdomain httpd[7711]: Server configured, listening on: port 81
Oct 09 20:23:24 ovgubina.localdomain systemd[1]: Started The Apache HTTP Server.
[root@ovgubina ~]#

```

Рис. 4.31: Запуск веб-сервера

Сервер запустился также успешно, как и в тот раз, поскольку оба раза порт 81 был в списке портов.

21. Верните контекст `httpd_sys_content_t` к файлу `/var/www/html/test.html`:
`chcon -t httpd_sys_content_t /var/www/html/test.html` (рис. 4.32).

После этого попробуйте получить доступ к файлу через веб-сервер, введя в браузере адрес `http://127.0.0.1:81/test.html` (рис. 4.33).

Втидим содержимое файла — слово «test».

```

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

```

Рис. 4.32: Возвращаем контекст безопасности

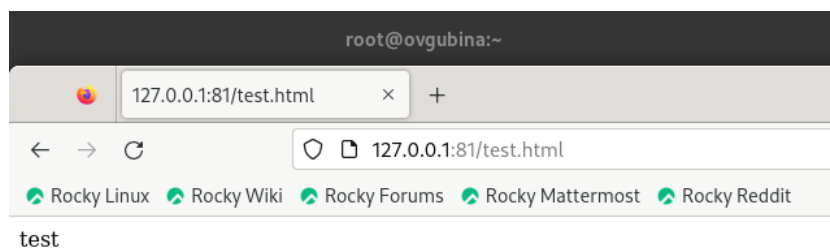


Рис. 4.33: Получение доступа к файлу через веб-сервер

22. Исправьте обратно конфигурационный файл `apache`, вернув `Listen 80` (рис. 4.34).

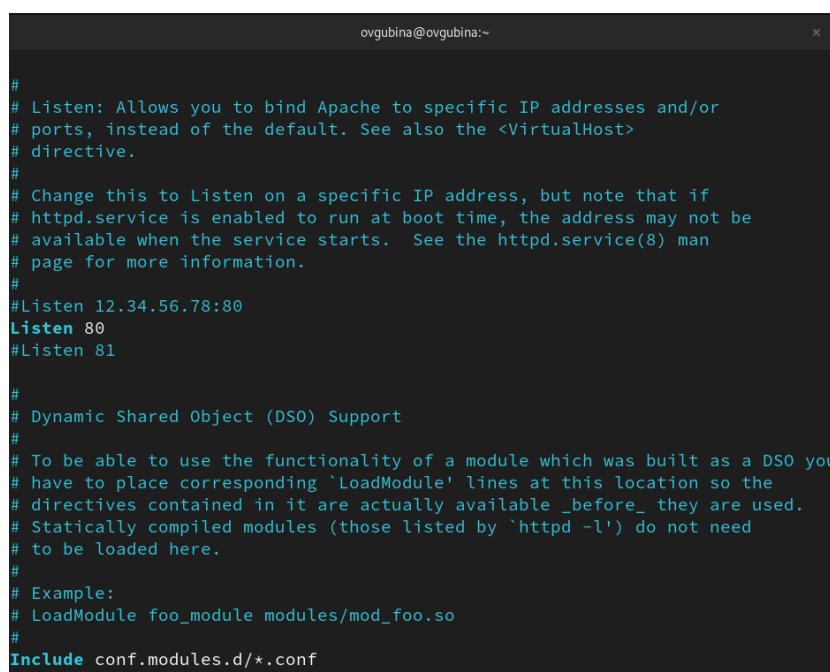


Рис. 4.34: Изменение прослушиваемого порта

23. Удалите привязку `http_port_t` к 81 порту: `semanage port -d -t http_port_t -p tcp 81` и проверьте, что порт 81 удалён.

Мы не можем этого сделать, поскольку это была изначальная настройка системы, работает - не трогай, поэтому мы пока не станем этого делать (рис. 4.35).

24. Удалите файл /var/www/html/test.html (рис. 4.35): `rm /var/www/html/test.html`

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

Рис. 4.35: Удаление файла

5 Выводы

Развила навыки администрирования ОС Linux. Получила первое практическое знакомство с технологией SELinux¹. Проверила работу SELinx на практике совместно с веб-сервером Apache.

Список литературы

1. SELinux – описание и особенности работы с системой [Электронный ресурс]. 2023. URL: <https://habr.com/ru/companies/kingservers/articles/209644/>.
2. Что такое Apache [Электронный ресурс]. 2023. URL: <https://eternalhost.net/blog/hosting/web-server-apache>.
3. Установка веб-сервера Apache на Linux [Электронный ресурс]. 2023. URL: <https://wiki.merionet.ru/articles/ustanovka-apache-v-linux>.