

Отчет по лабораторной работе №6

Дисциплина: Информационная безопасность

Выполнила: Афтаева Ксения Васильевна

Содержание

1	Цель работы	5
2	Задачи	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	9
4.1	Подготовка лабораторного стенда	9
4.2	Работа с SELinux и Apache	11
5	Выводы	23
	Список литературы	24

Список иллюстраций

4.1	Проверка политики и режима	9
4.2	Обновление системы	10
4.3	Обновление системы, установка веб-сервера Apache	10
4.4	Установка веб-сервера Apache	10
4.5	Задание параметра ServerName	11
4.6	Добавление разрешающих правил	11
4.7	Проверка режима и политики работы	12
4.8	Запуск веб-сервера	12
4.9	Контекст безопасности Apache	13
4.10	Состояние переключателей	14
4.11	Статистика по политике	15
4.12	Тип файлов и круг пользователей	15
4.13	Создание файла /var/www/html/test.html	16
4.14	Проверка контекста файла	16
4.15	Проверка отображения файла в веб-браузере	16
4.16	Проверка контекста файла	17
4.17	Изменение контекста файла	18
4.18	Отказ в доступе к файлу через веб-сайт	18
4.19	Просмотр прав доступа на файл	19
4.20	Отказ в доступе к файлу через веб-сайт	19
4.21	Изменение на порт 81	19
4.22	Перезапуск с портом 81	19
4.23	Лог-файлы, просмотр сообщений об ошибках	20
4.24	Просмотр сообщений об ошибках	20
4.25	Список портов и запуск веб-сервера	21
4.26	Доступ к файлу через веб-сервер	21
4.27	Редактирование конфигурационного файла apache	22
4.28	Удаление файла	22

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 Задачи

1. Подготовить лабораторный стенд.
2. Ознакомиться с технологией SELinux.
3. Выполнить задания по работе с SELinux совместно с веб-сервером Apache.

3 Теоретическое введение

SELinux — это система принудительного контроля доступа, реализованная на уровне ядра [1].

SELinux имеет три основных режима работы, при этом по умолчанию установлен режим Enforcing. Это довольно жесткий режим, и в случае необходимости он может быть изменен на более удобный для конечного пользователя.

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.

Политики SELinux бывают тоже нескольких типов [2]. Могут использоваться три основные политики:

- **targeted** - защищает основные системные сервисы, например, веб-сервер, DHCP, DNS, но не трогает все остальные программы;
- **strict** - самая строгая политика, управляет не только сетевыми службами, но и программами пользователя;
- **mls** - содержит не только правила, но и различные уровни безопасности; она позволяет реализовать многоуровневую систему безопасности на основе SELinux.

Также можно добавить свои политики.

Все процессы и файлы в рамках SELinux имеют контекст безопасности.

Apache – это свободное программное обеспечение для размещения веб-сервера [3]. 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.

Система конфигурации Apache работает на текстовых файлах с прописанными настройками. Она подразделяется на три условных уровня, для каждого из которых имеется свой конфигурационный файл:

1. Уровень конфигурации сервера (файл `httpd.conf`) – основной конфигурационный файл. Действие распространяется на весь механизм веб-сервера.
2. Уровень каталога (файл `.htaccess`) – дополнительный конфигурационный файл. Его директивы охватывают только каталог, где расположен файл, а также вложенные подкаталоги.
3. Уровень виртуального хоста (файл `httpd.conf` или `extra/httpd-vhosts.conf`).

Обычно конфигурационные файлы Apache находятся в папке «`conf`», а дополнительные конфигурационные файлы во вложенной в нее папке «`extra`». Внести изменения можно как через редактирование самого файла, так и через командную строку.

4 Выполнение лабораторной работы

4.1 Подготовка лабораторного стенда

1. Посмотрела конфигурационный файл `/etc/selinux/config`, чтобы проверить используемый режим и политику (рис. 4.1). Видим, что установлены политика `targeted` и режим `enforcing`, поэтому специальных настроек не требуется.

```
[kvaftaeva@kvaftaeva ~]$ su -
Password:
[root@kvaftaeva ~]# cat /etc/selinux/config

# This file controls the state of SELinux on the system.
# SELINUX= can take one of these 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 these three values:
#   targeted - Targeted processes are protected,
#   minimum - Modification of targeted policy. Only selected processes are protected.
#   mls - Multi Level Security protection.
SELINUXTYPE=targeted

[root@kvaftaeva ~]#
```

Рис. 4.1: Проверка политики и режима

2. Обновила систему командой `yum update` (рис. 4.2 - 4.3). После этого, установила веб-сервер Apache командой `yum install httpd` (рис. 4.3 - 4.4). Видим, что все установилось успешно.

```
[root@kvaftaeva ~]# yum update
Rocky Linux 9 - BaseOS                               5.6 kB/s | 4.1 kB    00:00
Rocky Linux 9 - BaseOS                               1.0 MB/s | 1.9 MB    00:01
Rocky Linux 9 - AppStream                             9.5 kB/s | 4.5 kB    00:00
Rocky Linux 9 - AppStream                             ] 185 kB/s | 63 kB    00:09 ETA
3% [=
```

Рис. 4.2: Обновление системы

```
systemd-libs-252-14.el9_2.3.0.1.x86_64
systemd-pam-252-14.el9_2.3.0.1.x86_64
systemd-rpm-macros-252-14.el9_2.3.0.1.noarch
systemd-udev-252-14.el9_2.3.0.1.x86_64
texlive-lib-9:20200406-26.el9_2.x86_64
webkitgtk3-2.38.5-1.el9_2.3.x86_64
webkitgtk3-jsc-2.38.5-1.el9_2.3.x86_64
Installed:
  grub2-tools-efi-1:2.06-61.el9_2.1.rocky.0.2.x86_64
  kernel-5.14.0-284.30.1.el9_2.x86_64
  kernel-devel-5.14.0-284.30.1.el9_2.x86_64
  kernel-modules-core-5.14.0-284.30.1.el9_2.x86_64
  grub2-tools-extra-1:2.06-61.el9_2.1.rocky.0.2.x86_64
  kernel-core-5.14.0-284.30.1.el9_2.x86_64
  kernel-modules-5.14.0-284.30.1.el9_2.x86_64
Complete!
[root@kvaftaeva ~]# yum install httpd
Last metadata expiration check: 0:00:43 ago on Thu 12 Oct 2023 01:04:29 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            14 k
httpd-tools            x86_64            2.4.53-11.el9_2.5  appstream            81 k
rocky-logos-httpd     noarch            90.14-1.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: Обновление системы, установка веб-сервера Apache

```
verifying : httpd-core-2.4.53-11.el9_2.5.x86_64
Installed:
apr-1.7.0-11.el9.x86_64
apr-util-1.6.1-20.el9_2.1.x86_64
httpd-2.4.53-11.el9_2.5.x86_64
httpd-filesystem-2.4.53-11.el9_2.5.noarch
mod_http2-1.15.19-4.el9_2.4.x86_64
rocky-logos-httpd-90.14-1.el9.noarch
apr-util-1.6.1-20.el9_2.1.x86_64
apr-util-openssl-1.6.1-20.el9_2.1.x86_64
httpd-core-2.4.53-11.el9_2.5.x86_64
httpd-tools-2.4.53-11.el9_2.5.x86_64
mod_lua-2.4.53-11.el9_2.5.x86_64
Complete!
```

Рис. 4.4: Установка веб-сервера Apache

- Задала в конфигурационном файле `/etc/httpd/conf/httpd.conf` параметр `ServerName test.ru` чтобы при запуске веб-сервера не выдавались лишние сообщения об ошибках, не относящихся к лабораторной работе (рис. 4.5).

```
GNU nano 5.6.1 /etc/httpd/conf/httpd.conf
#
# ServerAdmin: Your address, where problems with the server should be
# e-mailed. This address appears on some server-generated pages, such
# as error documents. e.g. admin@your-domain.com
#
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 test.ru
```

Рис. 4.5: Задание параметра ServerName

4. Чтобы пакетный фильтр в своей рабочей конфигурации позволял подключаться к 80-у и 81-у портам протокола tcp, добавила разрешающие правила (рис. 4.6):

```
iptables -I INPUT -p tcp --dport 80 -j ACCEPT
iptables -I INPUT -p tcp --dport 81 -j ACCEPT
iptables -I OUTPUT -p tcp --sport 80 -j ACCEPT
iptables -I OUTPUT -p tcp --sport 81 -j ACCEPT
```

```
[root@kvaftaeva ~]# nano /etc/httpd/conf/httpd.conf
[root@kvaftaeva ~]# iptables -I INPUT -p tcp --dport 80 -j ACCEPT
[root@kvaftaeva ~]# iptables -I INPUT -p tcp --dport 81 -j ACCEPT
[root@kvaftaeva ~]# iptables -I OUTPUT -p tcp --sport 80 -j ACCEPT
[root@kvaftaeva ~]# iptables -I OUTPUT -p tcp --sport 81 -j ACCEPT
[root@kvaftaeva ~]#
```

Рис. 4.6: Добавление разрешающих правил

4.2 Работа с SELinux и Apache

1. Вошла в систему и убедилась, что SELinux работает в режиме enforcing политики targeted с помощью команд getenforce и sestatus (рис. 4.7).

```
[kvaftaeva@kvaftaeva ~]$ getenforce
Enforcing
[kvaftaeva@kvaftaeva ~]$ 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. 4.7: Проверка режима и политики работы

2. Обратилась с помощью браузера к веб-серверу, запущенному на моем компьютере, с помощью команды `service httpd status` (рис. 4.8). Видим, что он не работает. Запустила его с помощью команды `service httpd start` (рис. 4.8). Убедилась, что он работает с помощью команды `service httpd status` (рис. 4.8).

```
[kvaftaeva@kvaftaeva ~]$ 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)
[kvaftaeva@kvaftaeva ~]$ service httpd start
Redirecting to /bin/systemctl start httpd.service
[kvaftaeva@kvaftaeva ~]$ 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 Thu 2023-10-12 13:15:28 MSK; 3s ago
     Docs: man:httpd.service(8)
  Main PID: 100332 (httpd)
    Status: "Started, listening on: port 80"
    Tasks: 213 (Limit: 50455)
    Memory: 27.5M
      CPU: 104ms
    CGroup: /system.slice/httpd.service
            └─100332 /usr/sbin/httpd -DFOREGROUND
              └─100340 /usr/sbin/httpd -DFOREGROUND
                └─100341 /usr/sbin/httpd -DFOREGROUND
                  └─100342 /usr/sbin/httpd -DFOREGROUND
                    └─100345 /usr/sbin/httpd -DFOREGROUND

Oct 12 13:15:28 kvaftaeva.localdomain systemd[1]: Starting The Apache HTTP Server...
Oct 12 13:15:28 kvaftaeva.localdomain httpd[100332]: Server configured, listening on: port 80
Oct 12 13:15:28 kvaftaeva.localdomain systemd[1]: Started The Apache HTTP Server.
[kvaftaeva@kvaftaeva ~]$
```

Рис. 4.8: Запуск веб-сервера

3. Нашла веб-сервер Apache в списке процессов командой `ps auxZ | grep httpd` (рис. 4.9). Видим, что контекст безопасности здесь

system_u:system_r:httpd_t:s0. Контекст безопасности состоит из четырех полей: пользователь, роль, тип и уровень.

```
[kvaftaeva@kvaftaeva ~]$ ps auxZ | grep httpd
system_u:system_r:httpd_t:s0 root 100332 0.1 0.1 20116 11516 ? Ss 13:15 0:00 /usr/sbin/httpd -D
FOREGROUND
system_u:system_r:httpd_t:s0 apache 100340 0.0 0.0 21600 7312 ? S 13:15 0:00 /usr/sbin/httpd -D
FOREGROUND
system_u:system_r:httpd_t:s0 apache 100341 0.0 0.1 2259020 13048 ? Sl 13:15 0:00 /usr/sbin/httpd -D
FOREGROUND
system_u:system_r:httpd_t:s0 apache 100342 0.0 0.1 2455692 15096 ? Sl 13:15 0:00 /usr/sbin/httpd -D
FOREGROUND
system_u:system_r:httpd_t:s0 apache 100345 0.0 0.1 2259020 11004 ? Sl 13:15 0:00 /usr/sbin/httpd -D
FOREGROUND
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 kvaftae+ 100591 0.0 0.0 221664 2312 pts/1 S+ 13:16 0:00 grep
--color=auto httpd
[kvaftaeva@kvaftaeva ~]$
```

Рис. 4.9: Контекст безопасности Apache

4. Посмотрела текущее состояние переключателей SELinux для Apache с помощью команды `sestatus -b | grep httpd` (рис. 4.10). Видим, что большие из них находятся в положении «off».

```

[kvaftaeva@kvaftaeva ~]$ sestatus -b | grep httpd
httpd_anon_write off
httpd_builtin_scripting on
httpd_can_check_spam off
httpd_can_connect_ftp off
httpd_can_connect_ldap off
httpd_can_connect_mythtv off
httpd_can_connect_zabbix off
httpd_can_manage_courier_spool off
httpd_can_network_connect off
httpd_can_network_connect_cobbler off
httpd_can_network_connect_db off
httpd_can_network_memcache off
httpd_can_network_relay off
httpd_can_sendmail off
httpd_dbus_avahi off
httpd_dbus_sssd off
httpd_dontaudit_search_dirs off
httpd_enable_cgi on
httpd_enable_ftp_server off
httpd_enable_homedirs off
httpd_execmem off
httpd_graceful_shutdown off
httpd_manage_ipa off
httpd_mod_auth_ntlm_winbind off
httpd_mod_auth_pam off
httpd_read_user_content off
httpd_run_ipa off
httpd_run_preupgrade off
httpd_run_stickshift off
httpd_serve_cobbler_files off
httpd_setrlimit off
httpd_ssi_exec off
httpd_sys_script_anon_write off
httpd_tmp_exec off
httpd_tty_comm off
httpd_unified off
httpd_use_cifs off
httpd_use_fusefs off
httpd_use_gpg off
httpd_use_nfs off
httpd_use_opencryptoki off
httpd_use_openstack off
httpd_use_sasl off
httpd_verify_dns off
[kvaftaeva@kvaftaeva ~]$

```

Рис. 4.10: Состояние переключателей

5. Посмотрела статистику по политике с помощью команды `seinfo` (рис. 4.11).
Видим, что пользователей 8, типов 5100, ролей 14.

```

[kvaftaeva@kvaftaeva ~]$ 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
Validatetrans:           0
MLS Val. Tran:           0
Polcap:                  6
Typebounds:              0
Neverallowxperm:         0
Dontauditxperm:          0
Ibpkeycon:               0
Fs_use:                  35
Portcon:                 660
Nodecon:                 0

```

Рис. 4.11: Статистика по политике

6. Посмотрела содержимое директории /var/www, с помощью команды `ls -lZ /var/www` (рис. 4.12). Видим, что здесь находятся две папки, с типами `httpd_sys_script_exec_t` и `httpd_sys_content_t`.
7. Посмотрела содержимое директории /var/www/html, командой `ls -lZ /var/www/html` (рис. 4.12). Видим, что папка пуста.
8. Определила круг пользователей, которым разрешено создание файлов в директории /var/www/html. Создавать файлы может только владелец (root)

```

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

```

Рис. 4.12: Тип файлов и круг пользователей

9. Создала 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. 4.13):

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

```
[root@kvaftaeva ~]# nano /var/www/html/test.html
[root@kvaftaeva ~]# cat /var/www/html/test.html
<html>
<body>test</body>
</html>
[root@kvaftaeva ~]#
```

Рис. 4.13: Создание файла /var/www/html/test.html

10. Проверила контекст созданного файла. Видим, что контекст, присваиваемый по умолчанию вновь созданным файлам в директории /var/www/html - unconfined_u:object_r:httpd_sys_content_t:s0 (рис. 4.14).

```
[kvaftaeva@kvaftaeva ~]$ ls -lZ /var/www/html
total 4
-rw-r--r--. 1 root root unconfined_u:object_r:httpd_sys_content_t:s0 33 Oct 12 13:28 test.html
[kvaftaeva@kvaftaeva ~]$
```

Рис. 4.14: Проверка контекста файла

11. Обратилась к файлу через веб-сервер, введя в браузере адрес <http://127.0.0.1/test.html> (рис. 4.15). Видим, что файл был успешно отображён.

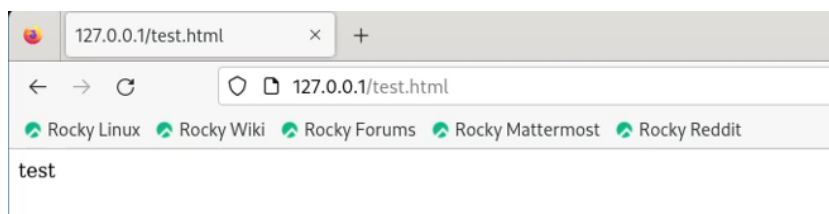


Рис. 4.15: Проверка отображения файла в веб-браузере

12. Изучила справочную информацию. SELinux требует наличия у файлов расширенных атрибутов, определяющих тип файла. 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

Проверила контекст файла командой `ls -Z /var/www/html/test.html` (рис. 4.16). Видим, что здесь `httpd_sys_content_t`, то есть содержимое должно быть доступно для всех скриптов httpd и для самого демона. 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. 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.

```
[kvaftaeva@kvaftaeva ~]$ ls -Z /var/www/html/test.html
unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/test.html
[kvaftaeva@kvaftaeva ~]$
```

Рис. 4.16: Проверка контекста файла

13. Изменила контекст файла `/var/www/html/test.html` с `httpd_sys_content_t` на `samba_share_t`, к которому процесс `httpd` не имеет доступа, командой `chcon -t samba_share_t /var/www/html/test.html` (рис. 4.17). Затем проверила, что контекст поменялся командой `ls -Z /var/www/html/test.html` (рис. 4.17).

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

Рис. 4.17: Изменение контекста файла

14. Попробовала ещё раз получить доступ к файлу через веб-сервер, введя в браузере адрес `http://127.0.0.1/test.html` (рис. 4.18). Получено сообщение об ошибке.

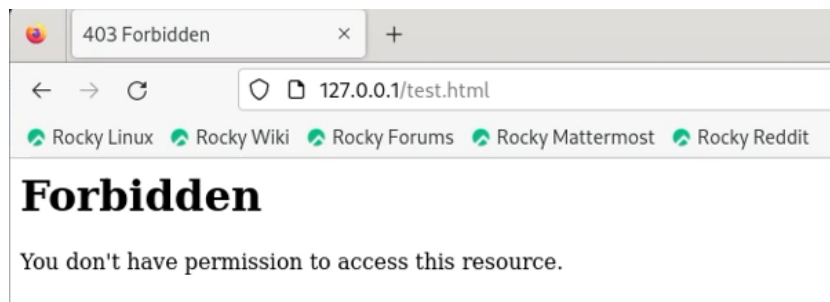


Рис. 4.18: Отказ в доступе к файлу через веб-сайт

15. Файл не был отображен из-за недопустимого для `httpd` контекста безопасности, несмотря на то, что права доступа позволяют читать этот файл (рис. 4.19). Просмотрела системный лог-файл командой `tail /var/log/messages` (рис. 4.20). Процессы `setroubleshootd` и `audtd` не запущены.

```
[kvaftaeva@kvaftaeva ~]$ ls -l /var/www/html/test.html
-rw-r--r--. 1 root root 33 Oct 12 13:28 /var/www/html/test.html
[kvaftaeva@kvaftaeva ~]$
```

Рис. 4.19: Просмотр прав доступа на файл

```
[root@kvaftaeva ~]# tail /var/log/messages
Oct 12 13:55:27 kvaftaeva systemd[1]: Starting Fingerprint Authentication Daemon...
Oct 12 13:55:27 kvaftaeva systemd[1]: Started Fingerprint Authentication Daemon.
Oct 12 13:55:31 kvaftaeva su[101834]: (to root) kvaftaeva on pts/2
Oct 12 13:55:31 kvaftaeva systemd[1]: Starting Hostname Service...
Oct 12 13:55:31 kvaftaeva systemd[1]: Started Hostname Service.
Oct 12 13:55:58 kvaftaeva systemd[1]: fprintd.service: Deactivated successfully.
Oct 12 13:56:01 kvaftaeva systemd[1]: systemd-hostnamed.service: Deactivated successfully.
Oct 12 13:56:41 kvaftaeva chronyd[802]: Selected source 94.247.111.10 (2.rocky.pool.ntp.org)
Oct 12 13:57:46 kvaftaeva chronyd[802]: Source 91.200.94.10 replaced with 192.36.143.130 (2.rocky.pool.ntp.org)
Oct 12 13:59:54 kvaftaeva chronyd[802]: Selected source 176.215.178.239 (2.rocky.pool.ntp.org)
[root@kvaftaeva ~]#
```

Рис. 4.20: Отказ в доступе к файлу через веб-сайт

16. Попробовала запустить веб-сервер Apache на прослушивание TCP-порта 81. Для этого в файле /etc/httpd/conf/httpd.conf заменила строчку Listen 80 на Listen 81 (рис. 4.21).

```
# page for more information.
#
#Listen 12.34.56.78:80
Listen 81
#
```

Рис. 4.21: Изменение на порт 81

17. Выполнила перезапуск веб-сервера Apache командой `service httpd restart` (рис. 4.22). Сбоя не произошло, так как возможность прослушивания 81 порта была прописана в виде разрешающих правил в разделе подготовки стенда.

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

Рис. 4.22: Перезапуск с портом 81

18. Проанализировала лог-файлы (команда `tail -nl /var/log/messages`) (рис. 4.23). Просмотрела файлы `/var/log/http/error_log` (рис. 4.23), `/var/log/http/access_log` (рис. 4.23) и `/var/log/audit/audit.log` (рис. 4.24). Нигде нет записей об ошибках, так как ошибки не было.

```
[root@kvaftaeva ~]# tail -nl /var/log/messages
Oct 12 14:04:24 kvaftaeva systemd[1]: Started The Apache HTTP Server.
[root@kvaftaeva ~]# cat /var/log/http/error_log
cat: /var/log/http/error_log: No such file or directory
[root@kvaftaeva ~]# cat /var/log/httpd/error_log
[Thu Oct 12 13:15:28.815155 2023] [core:notice] [pid 100332:tid 100332] SELinux policy enabled; httpd running as context
system_u:system_r:httpd_t:s0
[Thu Oct 12 13:15:28.816880 2023] [suexec:notice] [pid 100332:tid 100332] AH01232: suEXEC mechanism enabled (wrapper: /
usr/sbin/suexec)
[Thu Oct 12 13:15:28.832057 2023] [lbmethod_heartbeat:notice] [pid 100332:tid 100332] AH02282: No slotmem from mod_heart
bmonitor
[Thu Oct 12 13:15:28.840782 2023] [mpm_event:notice] [pid 100332:tid 100332] AH00489: Apache/2.4.53 (Rocky Linux) confi
gured -- resuming normal operations
[Thu Oct 12 13:15:28.840836 2023] [core:notice] [pid 100332:tid 100332] AH00094: Command line: '/usr/sbin/httpd -D FORE
GROUND'
[Thu Oct 12 13:44:11.766194 2023] [core:error] [pid 100345:tid 100516] (13)Permission denied: [client 127.0.0.1:32918]
AH00835: access to /test.html denied (filesystem path '/var/www/html/test.html') because search permissions are missing
on a component of the path
[Thu Oct 12 14:04:22.996524 2023] [mpm_event:notice] [pid 100332:tid 100332] AH00492: caught SIGWINCH, shutting down gr
acefully
[Thu Oct 12 14:04:24.054326 2023] [core:notice] [pid 101987:tid 101987] SELinux policy enabled; httpd running as context
system_u:system_r:httpd_t:s0
[Thu Oct 12 14:04:24.055240 2023] [suexec:notice] [pid 101987:tid 101987] AH01232: suEXEC mechanism enabled (wrapper: /
usr/sbin/suexec)
[Thu Oct 12 14:04:24.072005 2023] [lbmethod_heartbeat:notice] [pid 101987:tid 101987] AH02282: No slotmem from mod_heart
bmonitor
[Thu Oct 12 14:04:24.080506 2023] [mpm_event:notice] [pid 101987:tid 101987] AH00489: Apache/2.4.53 (Rocky Linux) confi
gured -- resuming normal operations
[Thu Oct 12 14:04:24.080643 2023] [core:notice] [pid 101987:tid 101987] AH00094: Command line: '/usr/sbin/httpd -D FORE
GROUND'
[root@kvaftaeva ~]# cat /var/log/httpd/access_log
127.0.0.1 - - [12/Oct/2023:13:32:29 +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 - - [12/Oct/2023:13:32:29 +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 - - [12/Oct/2023:13:44:11 +0300] "GET /test.html HTTP/1.1" 403 199 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:10
9.0) Gecko/20100101 Firefox/115.0"
[root@kvaftaeva ~]# cat
```

Рис. 4.23: Лог-файлы, просмотр сообщений об ошибках

```
t" AUID="unset"
type=BPF msg=audit(1697108158.654:614): prog-id=0 op=UNLOAD
type=SERVICE_STOP msg=audit(1697108161.635:615): pid=1 uid=0 auid=4294967295 ses=4294967295 subj=system_
_t:s0 msg='unit=systemd-hostnamed comm="systemd" exe="/usr/lib/systemd/systemd" hostname=? addr=? termin
s'UID="root" AUID="unset"
type=BPF msg=audit(1697108161.685:616): prog-id=0 op=UNLOAD
type=BPF msg=audit(1697108161.685:617): prog-id=0 op=UNLOAD
type=USER_END msg=audit(1697108285.660:618): pid=101834 uid=1000 auid=1000 ses=3 subj=unconfined_u:uncor
ned_t:s0-s0:c0.c1023 msg='op=PAM:session_close grantors=pam_keyinit,pam_keyinit,pam_limits,pam_systemd,p
sk,pam_xauth acct="root" exe="/usr/bin/su" hostname=? addr=? terminal=/dev/pts/2 res=success'UID="kvafta
eva"
type=CRED_DISP msg=audit(1697108285.660:619): pid=101834 uid=1000 auid=1000 ses=3 subj=unconfined_u:uncor
ned_t:s0-s0:c0.c1023 msg='op=PAM:setcred grantors=pam_unix acct="root" exe="/usr/bin/su" hostname=? ad
ev/pts/2 res=success'UID="kvaftaeva" AUID="kvaftaeva"
type=SERVICE_STOP msg=audit(1697108664.010:620): pid=1 uid=0 auid=4294967295 ses=4294967295 subj=system_
_t:s0 msg='unit=httpd comm="systemd" exe="/usr/lib/systemd/systemd" hostname=? addr=? terminal=? res=suc
AUID="unset"
type=SERVICE_START msg=audit(1697108664.072:621): pid=1 uid=0 auid=4294967295 ses=4294967295 subj=system
_t:s0 msg='unit=httpd comm="systemd" exe="/usr/lib/systemd/systemd" hostname=? addr=? terminal=? res=su
AUID="unset"
[root@kvaftaeva ~]#
```

Рис. 4.24: Просмотр сообщений об ошибках

19. Проверила список портов командой `semanage port -l | grep http_port_t` (рис. 4.25). Видим, что порт 81 есть в списке.

20. Запустила веб-сервер Apache ещё раз (рис. 4.25). Он снова успешно запустился, так как мы ничего и не меняли (рис. 4.25). Он запустился и в первый и во второй раз, так как порт 81 был в списке портов.
21. Вернула контекст `httpd_sys_content_t` к файлу `/var/www/html/ test.html` командой `chcon -t httpd_sys_content_t /var/www/html/test.html` (рис. 4.25). После этого попробовала получить доступ к файлу через веб-сервер, введя в браузере адрес `http://127.0.0.1:81/test.html` (рис. 4.26). Видим содержимое файла — слово «test».

```
[root@kvaftaeva ~]# 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@kvaftaeva ~]# service httpd restart
Redirecting to /bin/systemctl restart httpd.service
[root@kvaftaeva ~]# 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 Thu 2023-10-12 14:11:36 MSK; 12s ago
     Docs: man:httpd.service(8)
   Main PID: 102296 (httpd)
    Status: "Total requests: 0; Idle/Busy workers 100/0; Requests/sec: 0; Bytes served/sec: 0 B/sec"
     Tasks: 213 (limit: 50455)
    Memory: 41.1M
       CPU: 91ms
   CGroup: /system.slice/httpd.service
           └─102296 /usr/sbin/httpd -DFOREGROUND
             └─102297 /usr/sbin/httpd -DFOREGROUND
               └─102298 /usr/sbin/httpd -DFOREGROUND
                 └─102299 /usr/sbin/httpd -DFOREGROUND
                   └─102300 /usr/sbin/httpd -DFOREGROUND

Oct 12 14:11:35 kvaftaeva.localdomain systemd[1]: Starting The Apache HTTP Server...
Oct 12 14:11:36 kvaftaeva.localdomain httpd[102296]: Server configured, listening on: port 81
Oct 12 14:11:36 kvaftaeva.localdomain systemd[1]: Started The Apache HTTP Server.
[root@kvaftaeva ~]# chcon -t httpd_sys_content_t /var/www/html/test.html
[root@kvaftaeva ~]# ls -Z /var/www/html/test.html
unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/test.html
[root@kvaftaeva ~]#
```

Рис. 4.25: Список портов и запуск веб-сервера

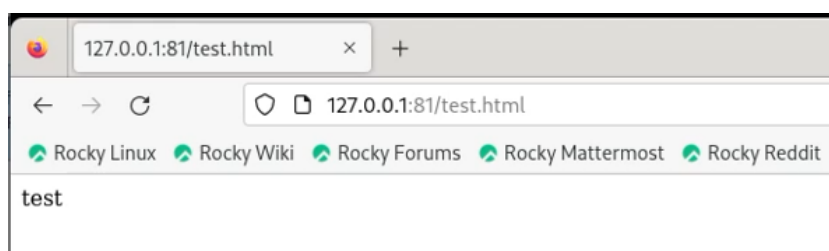


Рис. 4.26: Доступ к файлу через веб-сервер

22. Исправила обратно конфигурационный файл `apache`, вернув `Listen 80` (рис. 4.27).

```
# page for more information:
#
#Listen 12.34.56.78:80
Listen 80
#
```

Рис. 4.27: Редактирование конфигурационного файла apache

23. Удалить привязку к порту командой `semanage port -d -t http_port_t -p tcp 81` не удалось, так как она определена в политике (рис. 4.28). Исправлять это я не стала, сделаю это при необходимости позже.
24. Удалила файл `/var/www/html/test.html` командой `rm /var/www/html/test.html` (рис. 4.28).

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

Рис. 4.28: Удаление файла

5 Выводы

Я развила навыки администрирования ОС Linux. Получила первое практическое знакомство с технологией SELinux. Проверила работу SELinux на практике совместно с веб-сервером Apache.

Список литературы

1. SELinux - описание и особенности работы с системой [Электронный ресурс]. 2014. URL: <https://habr.com/ru/companies/kingservers/articles/209644/>.
2. Настройка SELinux [Электронный ресурс]. 2021. URL: <https://losst.pro/nastroyka-selinux>.
3. Что такое Apache [Электронный ресурс]. 2021. URL: <https://eternalhost.net/blog/hosting/web-server-apache>.