Отчет по лабораторной работе №6

Дисциплина: Информационная безопасность

Выполнила: Афтаева Ксения Васильевна

Содержание

Список иллюстраций

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.

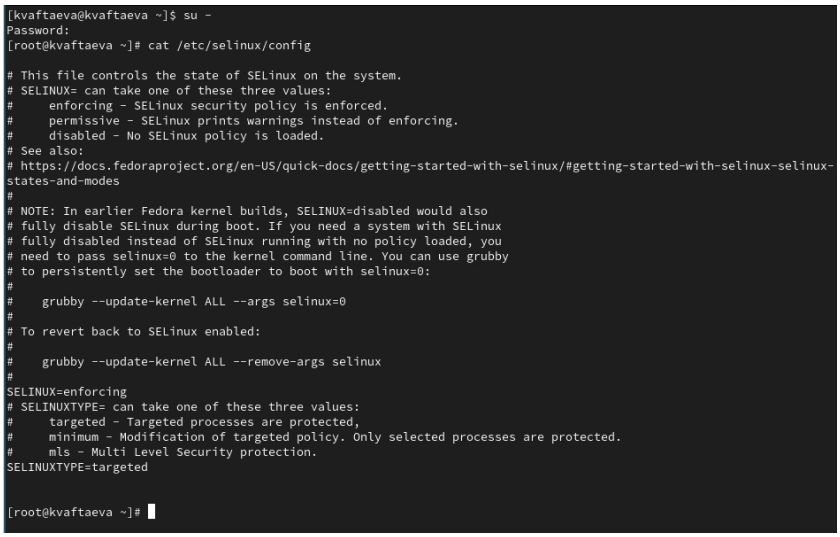

Рис. 4.1: Проверка политики и режима

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.

Рис. 4.2: Обновление системы

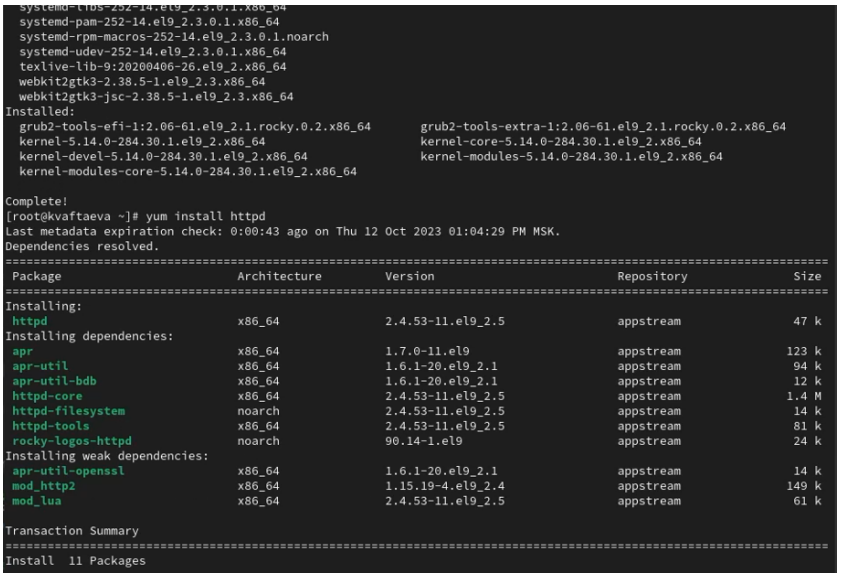

Рис. 4.3: Обновление системы, установка веб-сервера Apache

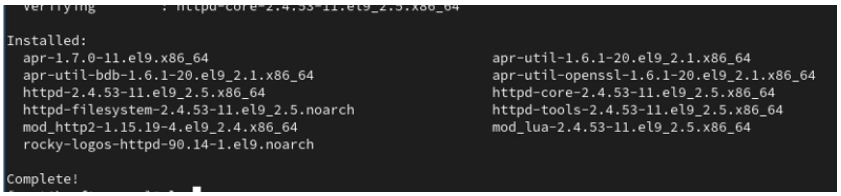

Рис. 4.4: Установка веб-сервера Apache

3. 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).

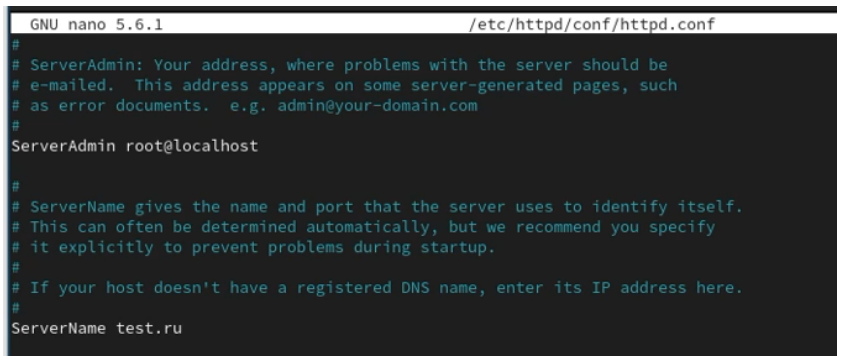

Рис. 4.5: Задание параметра ServerName

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
```

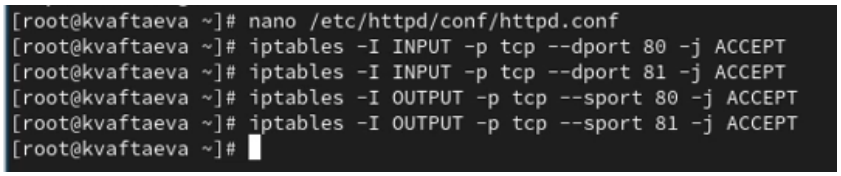

Рис. 4.6: Добавление разрешающих правил

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).

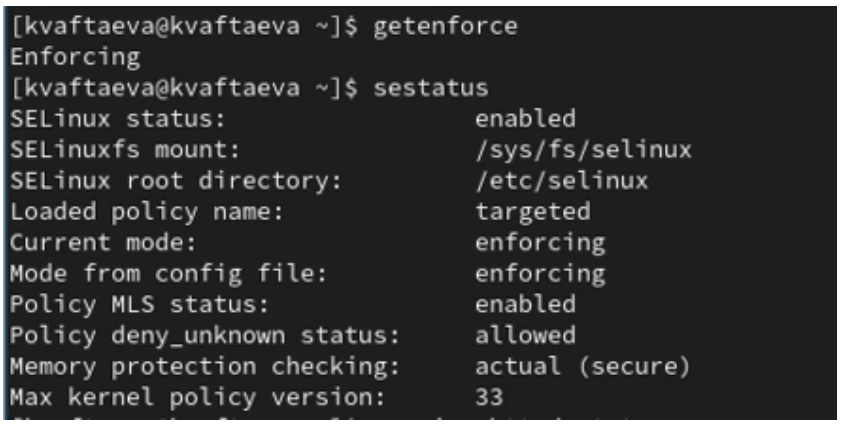

Рис. 4.7: Проверка режима и политики работы

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).

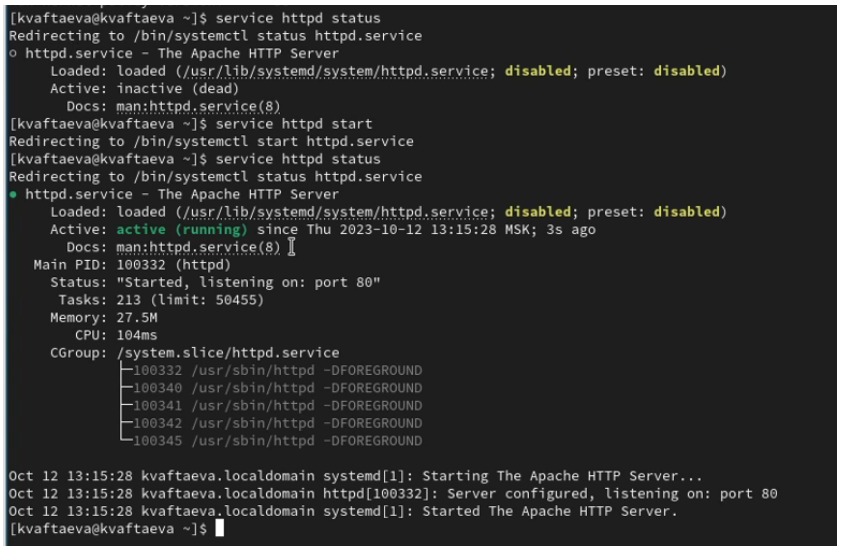

Рис. 4.8: Запуск веб-сервера

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.

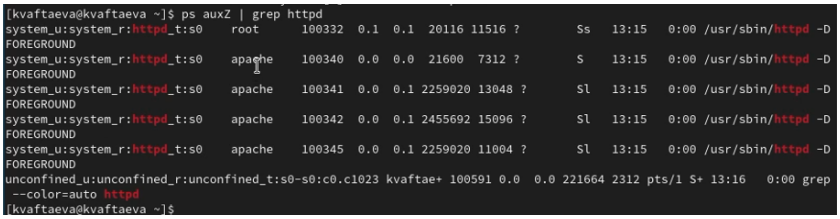

Рис. 4.9: Контекст безопасности Apache

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».

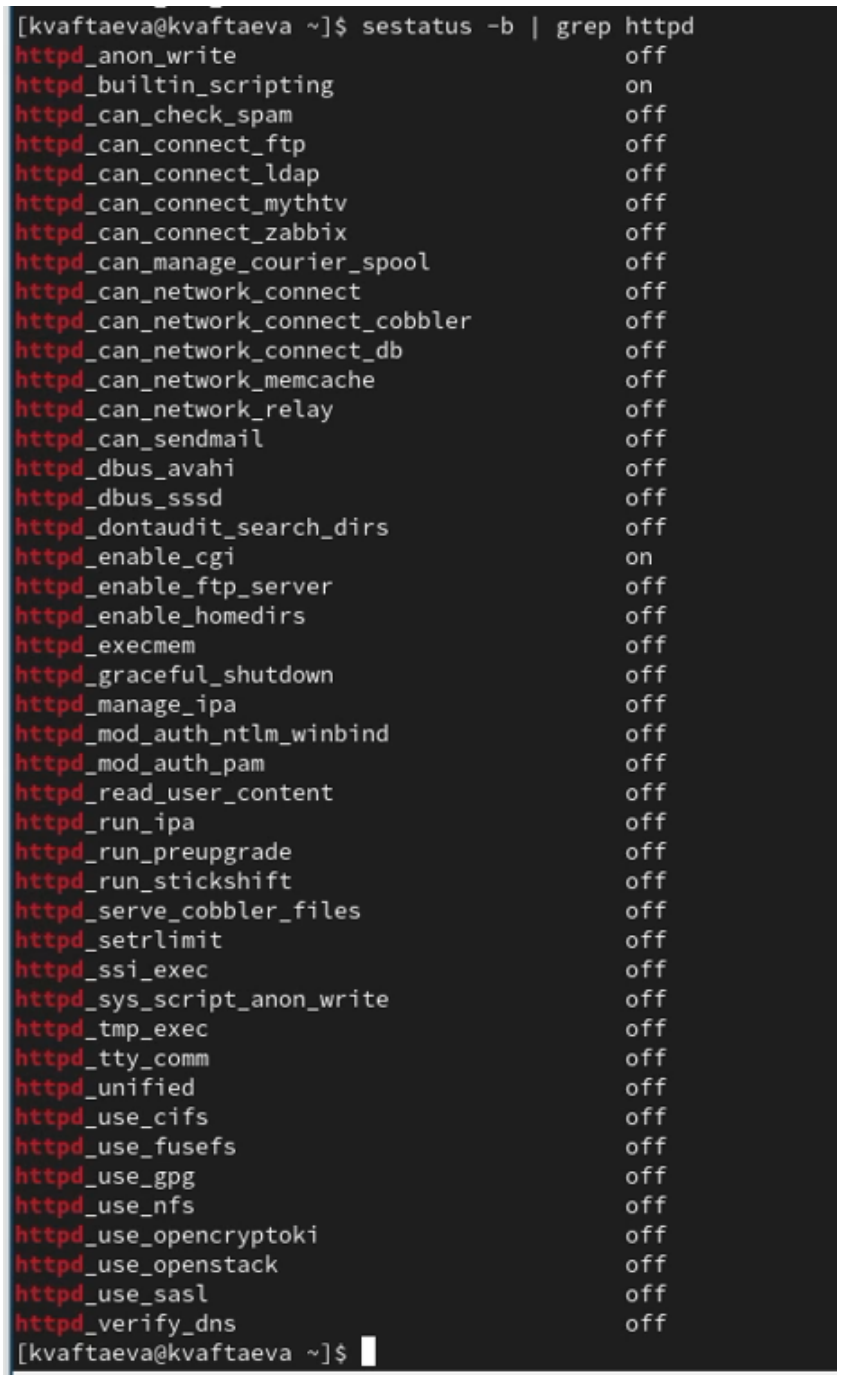

Рис. 4.10: Состояние переключателей

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

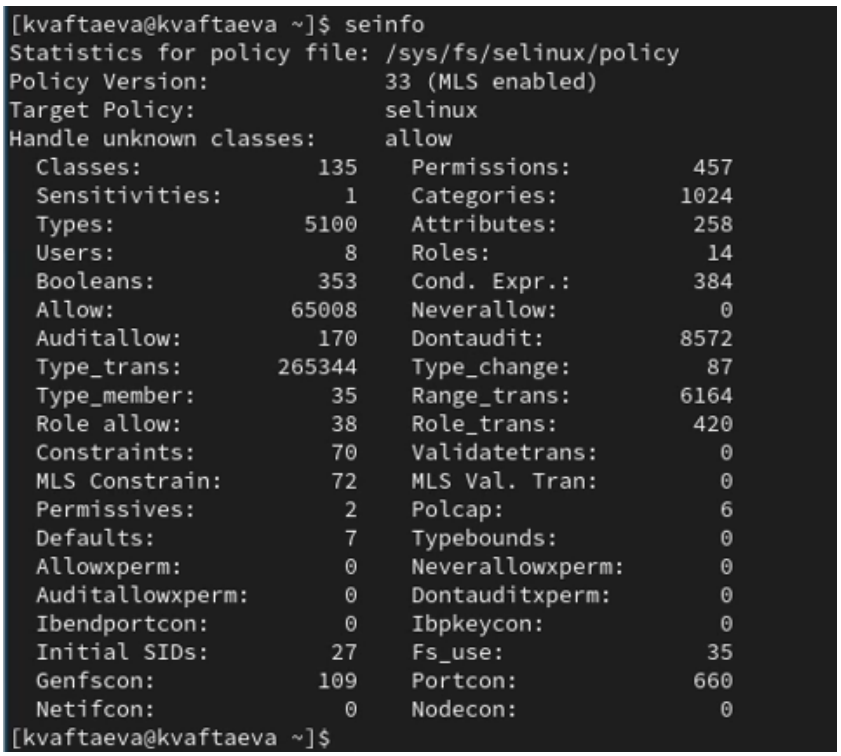

Рис. 4.11: Статистика по политике

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)

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>
```

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).

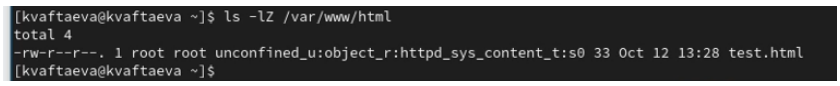

Рис. 4.14: Проверка контекста файла

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.

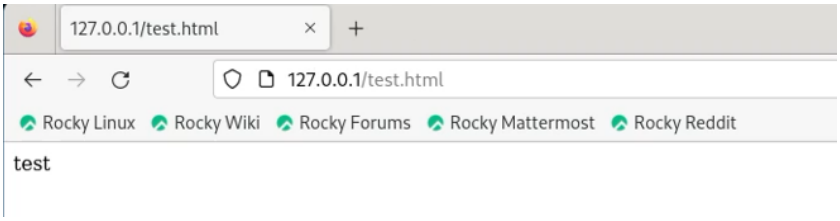

Рис. 4.15: Проверка отображения файла в веб-браузере

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.

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).

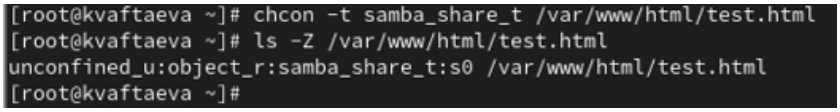

Рис. 4.17: Изменение контекста файла

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.

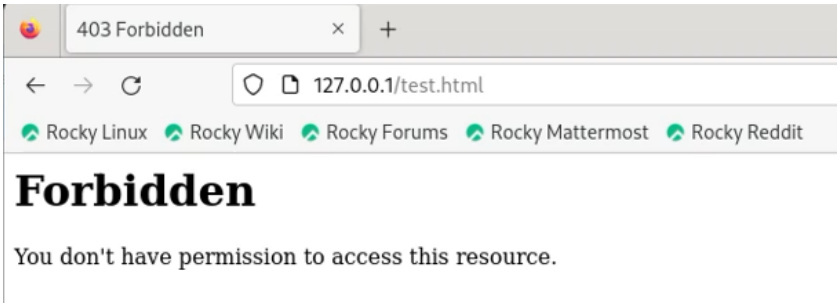

Рис. 4.18: Отказ в доступе к файлу через веб-сайт

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.

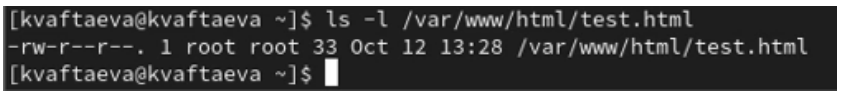

Рис. 4.19: Просмотр прав доступа на файл

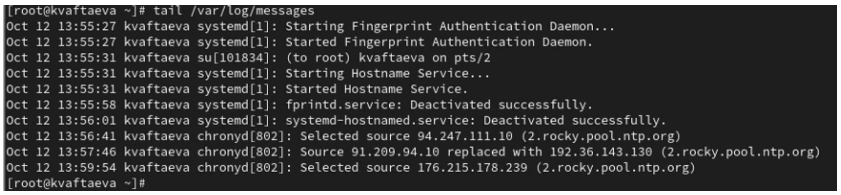

Рис. 4.20: Отказ в доступе к файлу через веб-сайт

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).

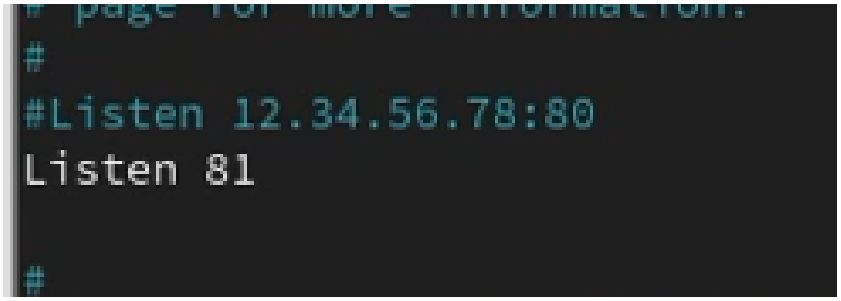

Рис. 4.21: Изменение на порт 81

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.

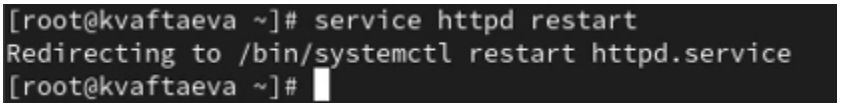

Рис. 4.22: Перезапуск с портом 81

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.

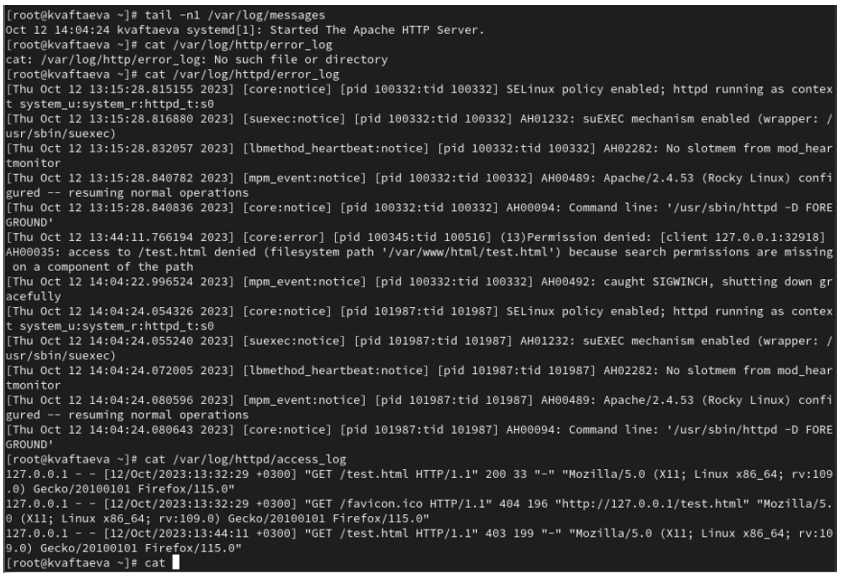

Рис. 4.23: Лог-файлы, просмотр сообщений об ошибках

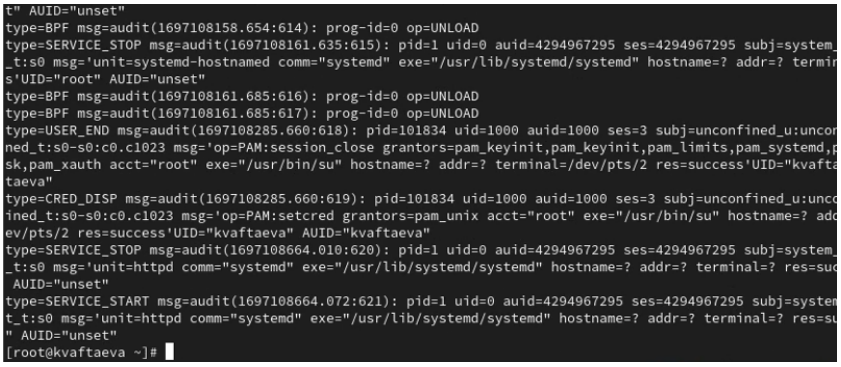

Рис. 4.24: Просмотр сообщений об ошибках

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».

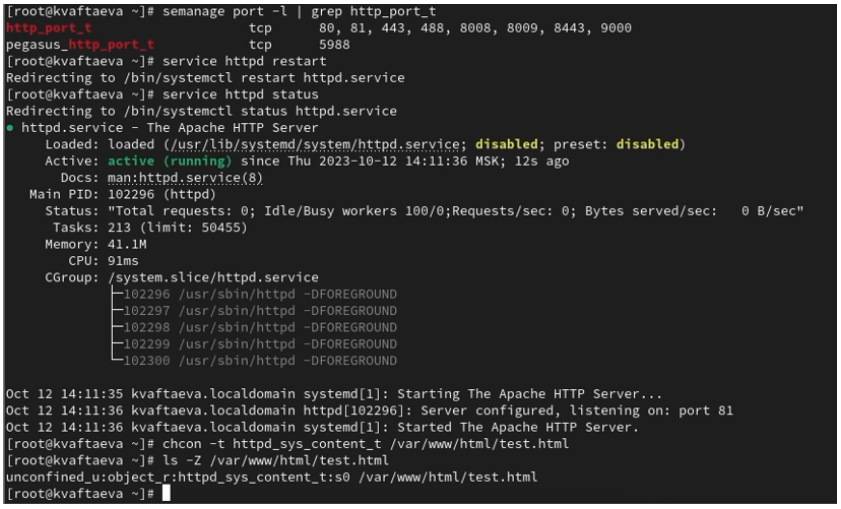

Рис. 4.25: Список портов и запуск веб-сервера

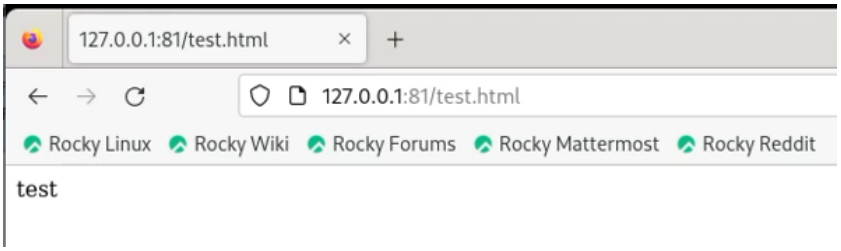

Рис. 4.26: Доступ к файлу через веб-сервер

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

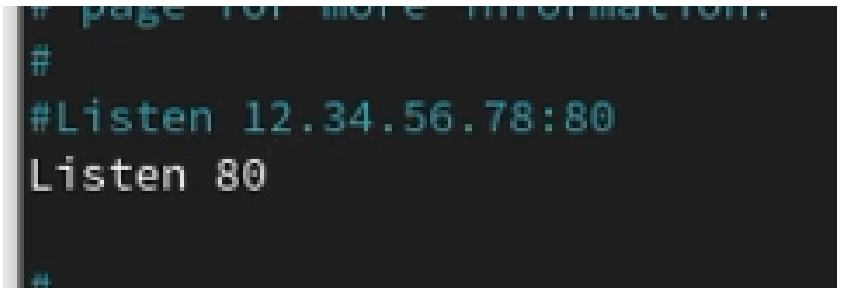

Рис. 4.27: Редактирование конфигурационного файла apache

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).

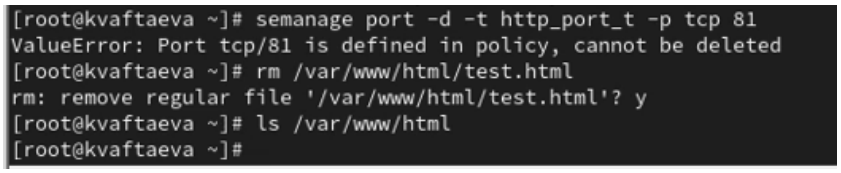

Рис. 4.28: Удаление файла

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/nastrojka-selinux.
3. Что такое Apache [Электронный ресурс]. 2021. URL: https://eternalhost.net/blog/hosting/web-server-apache.