Отчет по лабораторной работе №6

Дисциплина: Информационная безопасность

Выполнила: Афтаева Ксения Васильевна

Содержание

# 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:

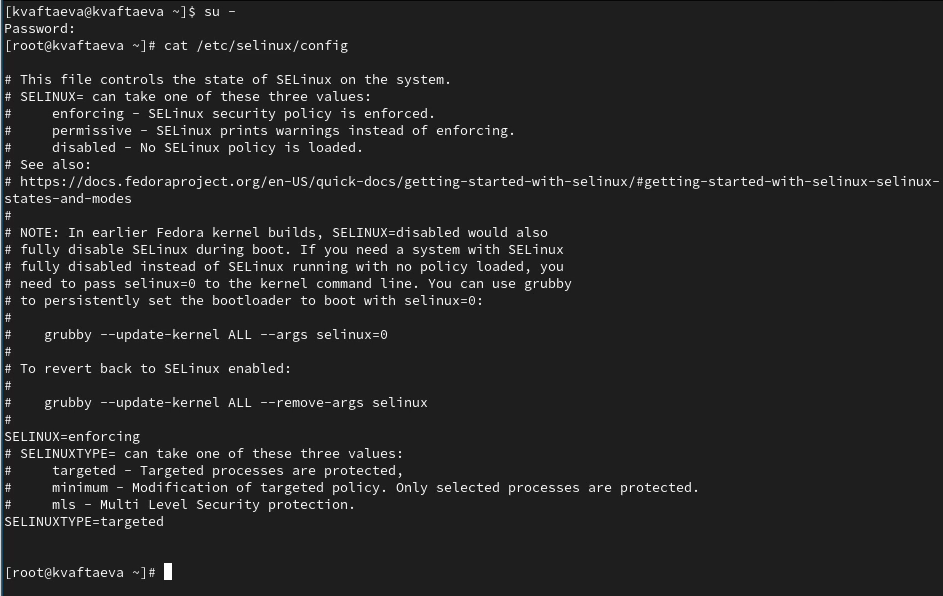
1. Уровень конфигурации сервера (файл httpd.conf) – основной конфигурационный файл. Действие распространяется на весь механизм веб-сервера.
2. Уровень каталога (файл .htaccess) – дополнительный конфигурационный файл. Его директивы охватывают только каталог, где расположен файл, а также вложенные подкаталоги.
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. ??). Видим, что установлены политика targeted и режим enforcing, поэтому специальных настроек не требуется.

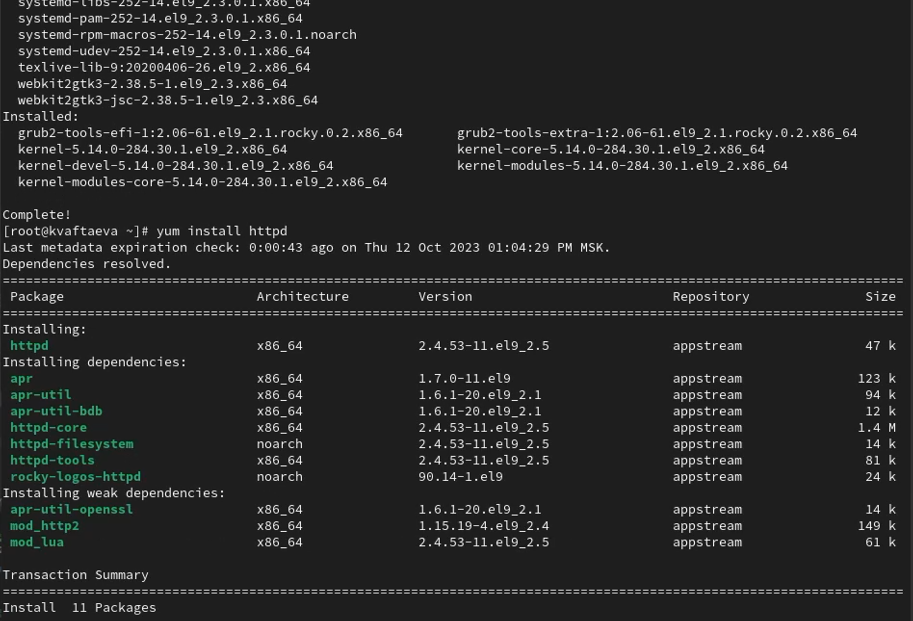


Проверка политики и режима

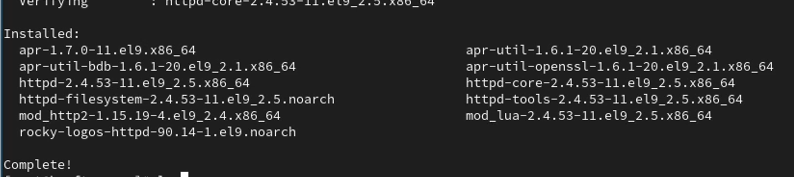
1. Обновила систему командой yum update (рис. ?? - ??). После этого, установила веб-сервер Apache командой yum install httpd (рис. ?? - ??). Видим, что все установилось успешно.



Обновление системы

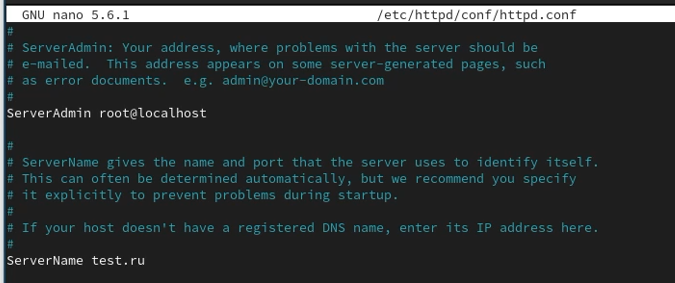


Обновление системы, установка веб-сервера Apache



Установка веб-сервера Apache

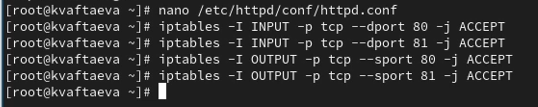
1. Задала в конфигурационном файле /etc/httpd/conf/httpd.conf параметр ServerName test.ru чтобы при запуске веб-сервера не выдавались лишние сообщения об ошибках, не относящихся к лабораторной работе (рис. ??).



Задание параметра ServerName

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

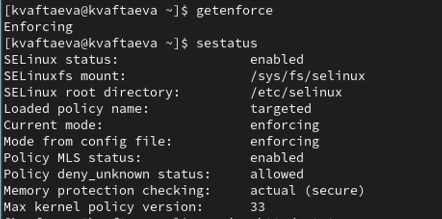
iptables -I INPUT -p tcp --dport 80 -j ACCEPT  
iptables -I INPUT -p tcp --dport 81 -j ACCEPT  
iptables -I OUTPUT -p tcp --sport 80 -j ACCEPT  
iptables -I OUTPUT -p tcp --sport 81 -j ACCEPT



Добавление разрешающих правил

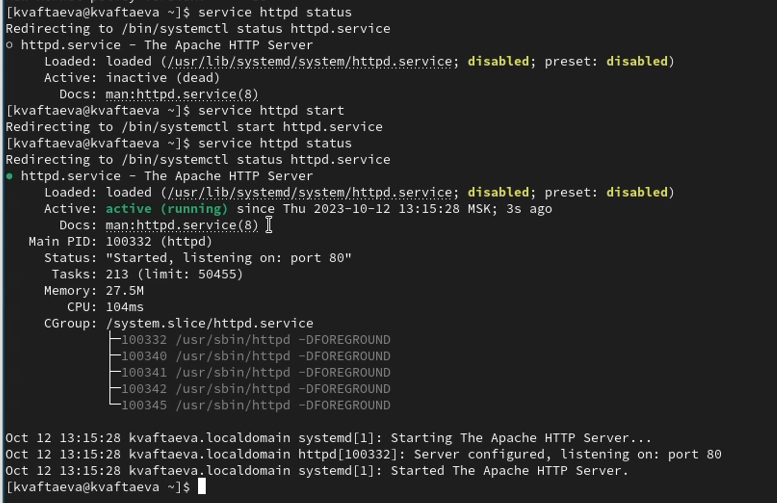
## 4.2 Работа с SELinux и Apache

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



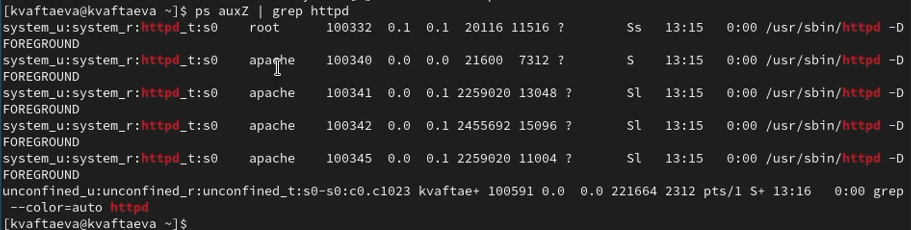
Проверка режима и политики работы

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



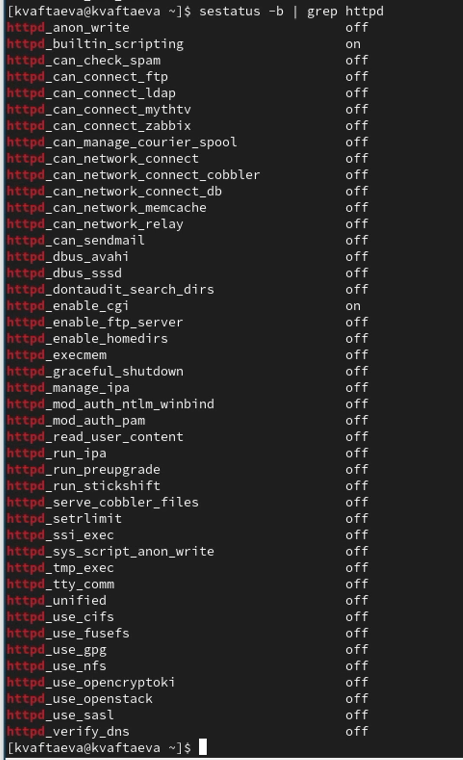
Запуск веб-сервера

1. Нашла веб-сервер Apache в списке процессов командой ps auxZ | grep httpd (рис. ??). Видим, что контекст безопасности здесь system\_u:system\_r:httpd\_t:s0. Контекст безопасности состоит из четырех полей: пользователь, роль, тип и уровень.



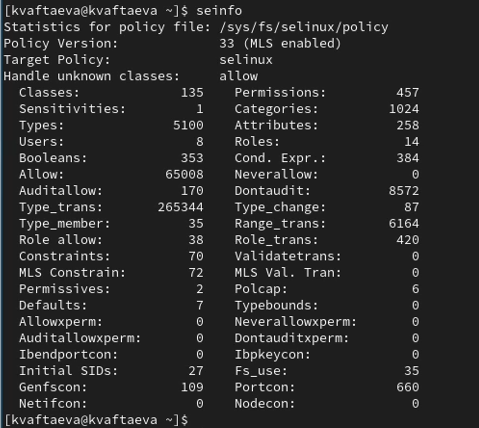
Контекст безопасности Apache

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



Состояние переключателей

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



Статистика по политике

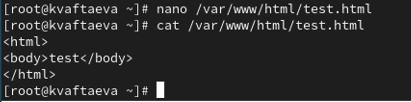
1. Посмотрела содержимое директории /var/www, с помощью команды ls -lZ /var/www (рис. ??). Видим, что здесь находятся две папки, с типами httpd\_sys\_script\_exec\_t и httpd\_sys\_content\_t.
2. Посмотрела содержимое директории /var/www/html, командой ls -lZ /var/www/html (рис. ??). Видим, что папка пуста.
3. Определила круг пользователей, которым разрешено создание файлов в директории /var/www/html. Создавать файлы может только владелец (root)



Тип файлов и круг пользователей

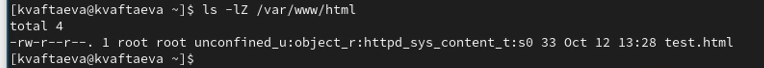
1. Создала от имени суперпользователя (так как в дистрибутиве после установки только ему разрешена запись в директорию) html-файл /var/www/html/test.html следующего содержания (рис. ??):

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



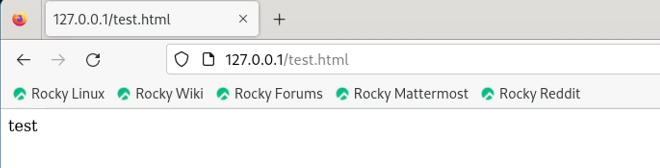
Создание файла /var/www/html/test.html

1. 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. ??).



Проверка контекста файла

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



Проверка отображения файла в веб-браузере

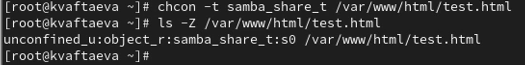
1. Изучила справочную информацию. SELinux требует наличия у файлов расширенных атрибутов, определяющих тип файла. Для httpd определены следующие контексты файлов:

* 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. ??). 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.

Проверка контекста файла

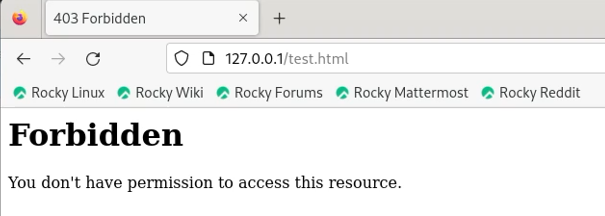
Проверка контекста файла

1. 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. ??). Затем проверила, что контекст поменялся командой ls -Z /var/www/html/test.html (рис. ??).



Изменение контекста файла

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

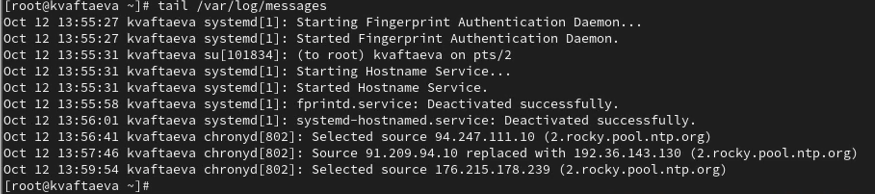


Отказ в доступе к файлу через веб-сайт

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

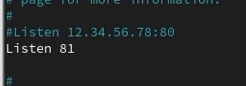
Просмотр прав доступа на файл

Просмотр прав доступа на файл



Отказ в доступе к файлу через веб-сайт

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



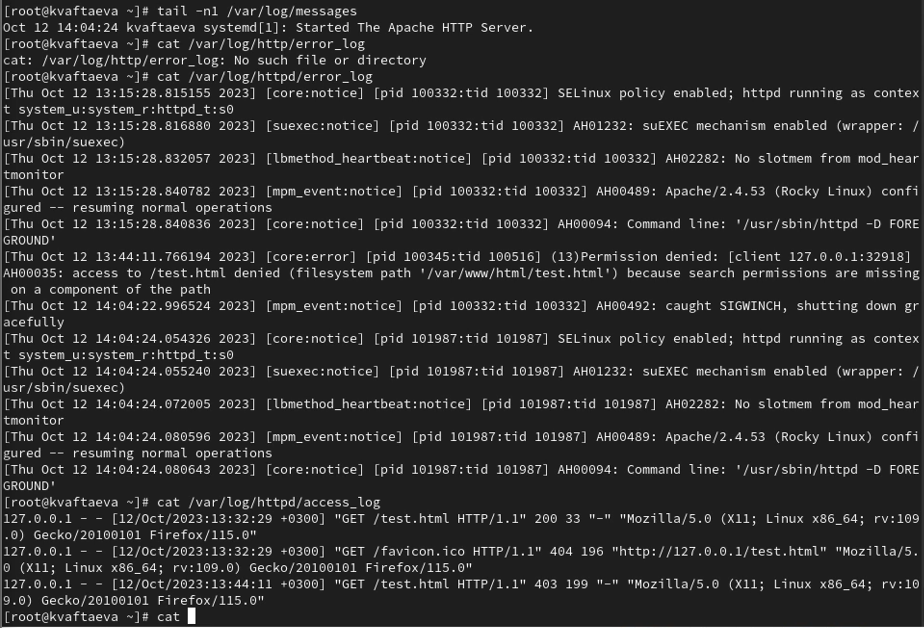
Изменение на порт 81

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

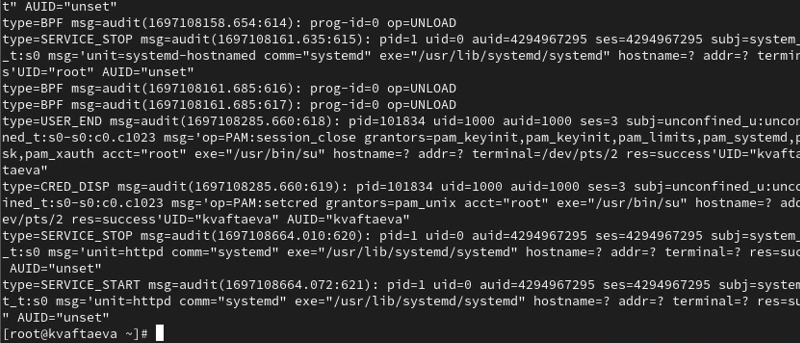
Перезапуск с портом 81

Перезапуск с портом 81

1. Проанализировала лог-файлы (команда tail -nl /var/log/messages) (рис. ??). Просмотрела файлы /var/log/http/error\_log (рис. ??), /var/log/http/access\_log (рис. ??) и /var/log/audit/audit.log (рис. ??). Нигде нет записей об ошибках, так как ошибки не было.

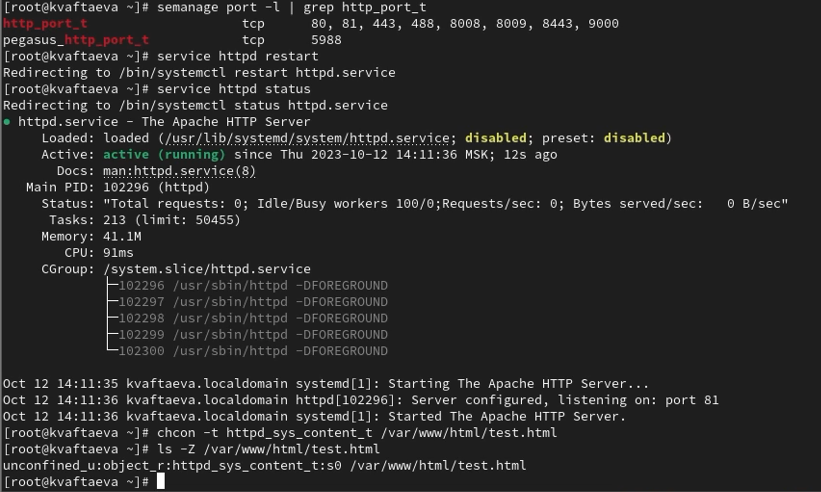


Лог-файлы, просмотр сообщений об ошибках

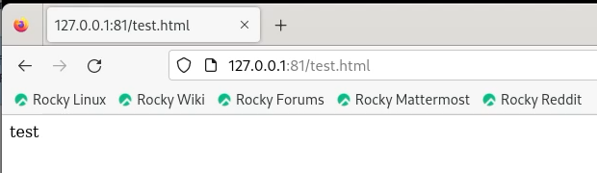


Просмотр сообщений об ошибках

1. Проверила список портов командой semanage port -l | grep http\_port\_t (рис. ??). Видим, что порт 81 есть в списке.
2. Запустила веб-сервер Apache ещё раз (рис. ??). Он снова успешно запустился, так как мы ничего и не меняли (рис. ??). Он запустился и в первый и во второй раз, так как порт 81 был в списке портов.
3. Вернула контекст httpd\_sys\_cоntent\_\_t к файлу /var/www/html/ test.html командой chcon -t httpd\_sys\_content\_t /var/www/html/test.html (рис. ??). После этого попробовала получить доступ к файлу через веб-сервер, введя в браузере адрес http://127.0.0.1:81/test.html (рис. ??). Видим содержимое файла — слово «test».

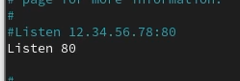


Список портов и запуск веб-сервера



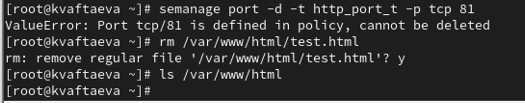
Доступ к файлу через веб-сервер

1. Исправила обратно конфигурационный файл apache, вернув Listen 80 (рис. ??).



Редактирование конфигурационного файла apache

1. Удалить привязку к порту командой semanage port -d -t http\_port\_t -p tcp 81 не удалось, так как она определена в политике (рис. ??). Исправлять это я не стала, сделаю это при необходимости позже.
2. Удалила файл /var/www/html/test.html командой rm /var/www/html/test.html (рис. ??).



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>.