Отчет по лабораторной работе №6

Мандатное разграничение прав в Linux

Евсеева Дарья Олеговна

15 октября, 2022

Содержание

# Цель работы

Целью данной работы является развитие навыков администрирования ОС Linux, ознакомление с технологией SELinux, а также проверка работы SELinux на практике совместно с веб-сервером Apache.

# Задание

Провести работу с SELinux и веб-сервером Apache.

# Теоретическое введение

Мандатное управление доступом (Mandatory Access Control, MAC) - это система разграничения доступа на основе уровня доступа субъекта и защитной метки объекта. Смысл состоит в том, что субъект может получить доступ к тем объектам, у которых метка безопасности имеет тот же уровень или ниже, что и у объекта.

Субъект - это пользователь, вернее процесс, который он инициализирует. Объект - это файл, программа, база данных и любой из ее объектов, даже сетевой пакет. Также предусмотрена иерархическая структура уровней доступа. Всем субъектам и объектам назначаются так называемые метки - значение уровня доступа у субъекта и значение уровня конфиденциальности для объекта.

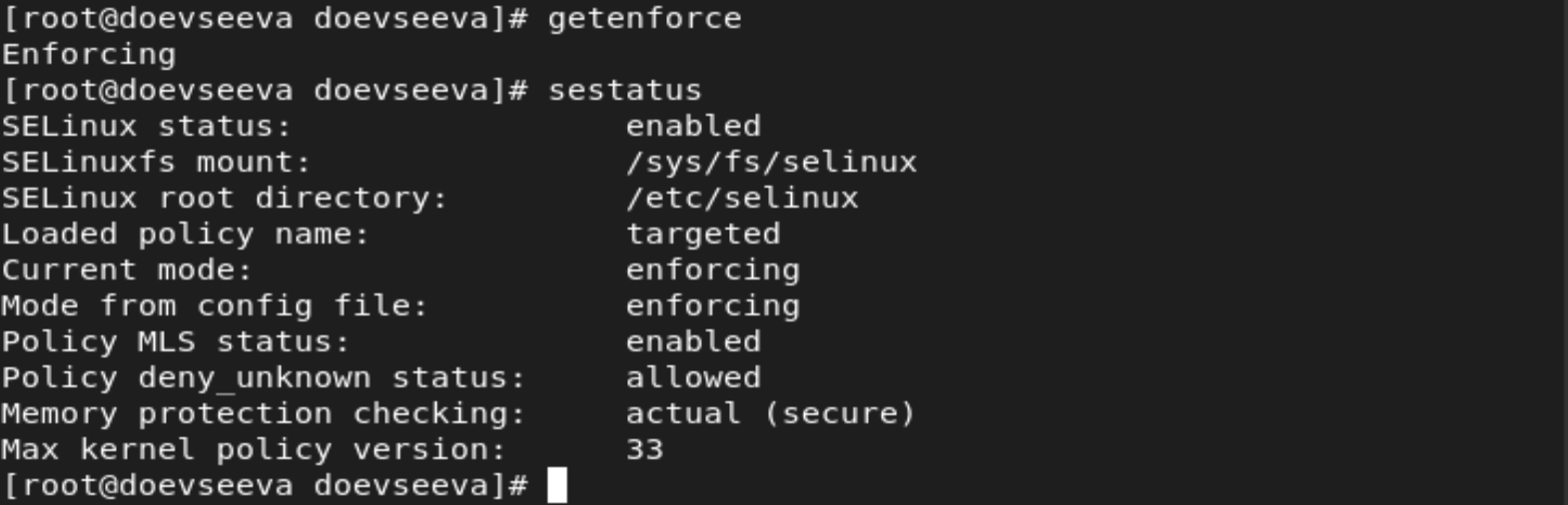
Каждый раз, когда субъект запрашивает объект происходит проверка соответствия меток и принимается решение о разрешении или запрете доступа. Так как структура уровней доступа иерархическая, то субъект имеет доступ к объектам соответствующего уровня конфиденциальности, а также ко всем другим уровням, находящимся по иерархии ниже. Проверка уровня доступа это вертикальная безопасность, но в MAC предусмотрена и горизонтальная. В дополнение к уровням безопасности, существуют категории. Благодаря им можно разграничивать доступ среди субъектов с одинаковым уровнем доступа.

# Выполнение лабораторной работы

Выполнять работу будем в операционной системе, установленной при выполнении первой лабораторной работы.

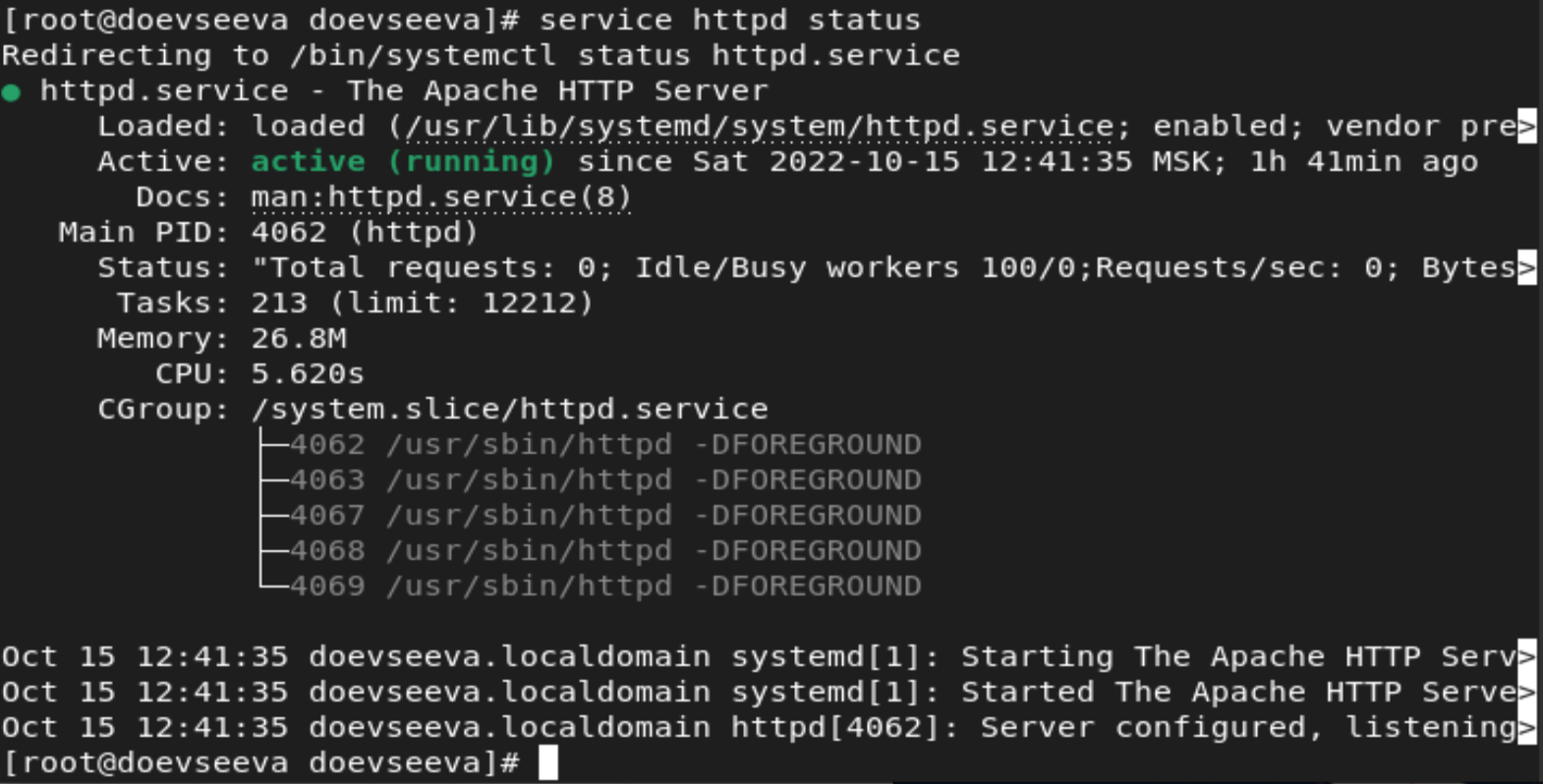
## Работа с SELinux и веб-сервером Apache

Для начала войдем в систему и убедимся, что SELinux работает в режиме enforcing политики targeted.



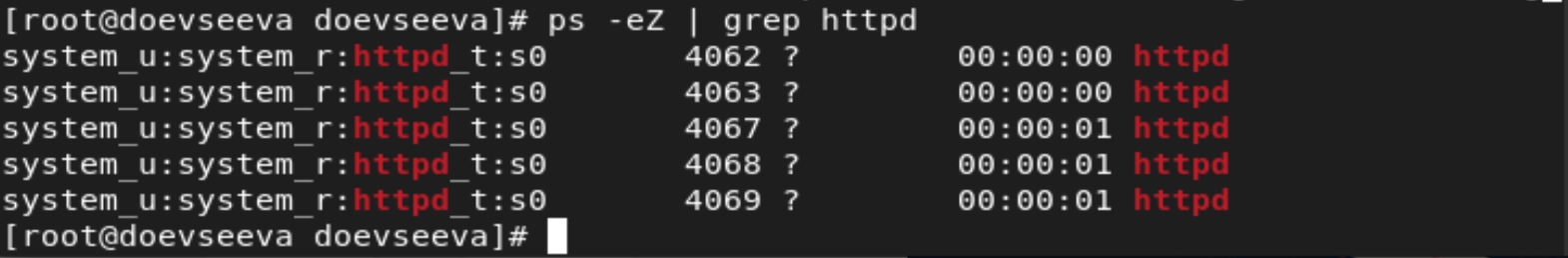
Проверка режима работы SELinux

Далее, обратимся с помощью браузера к веб-серверу и убедимся, что последний работает.



Проверка работы веб-сервера

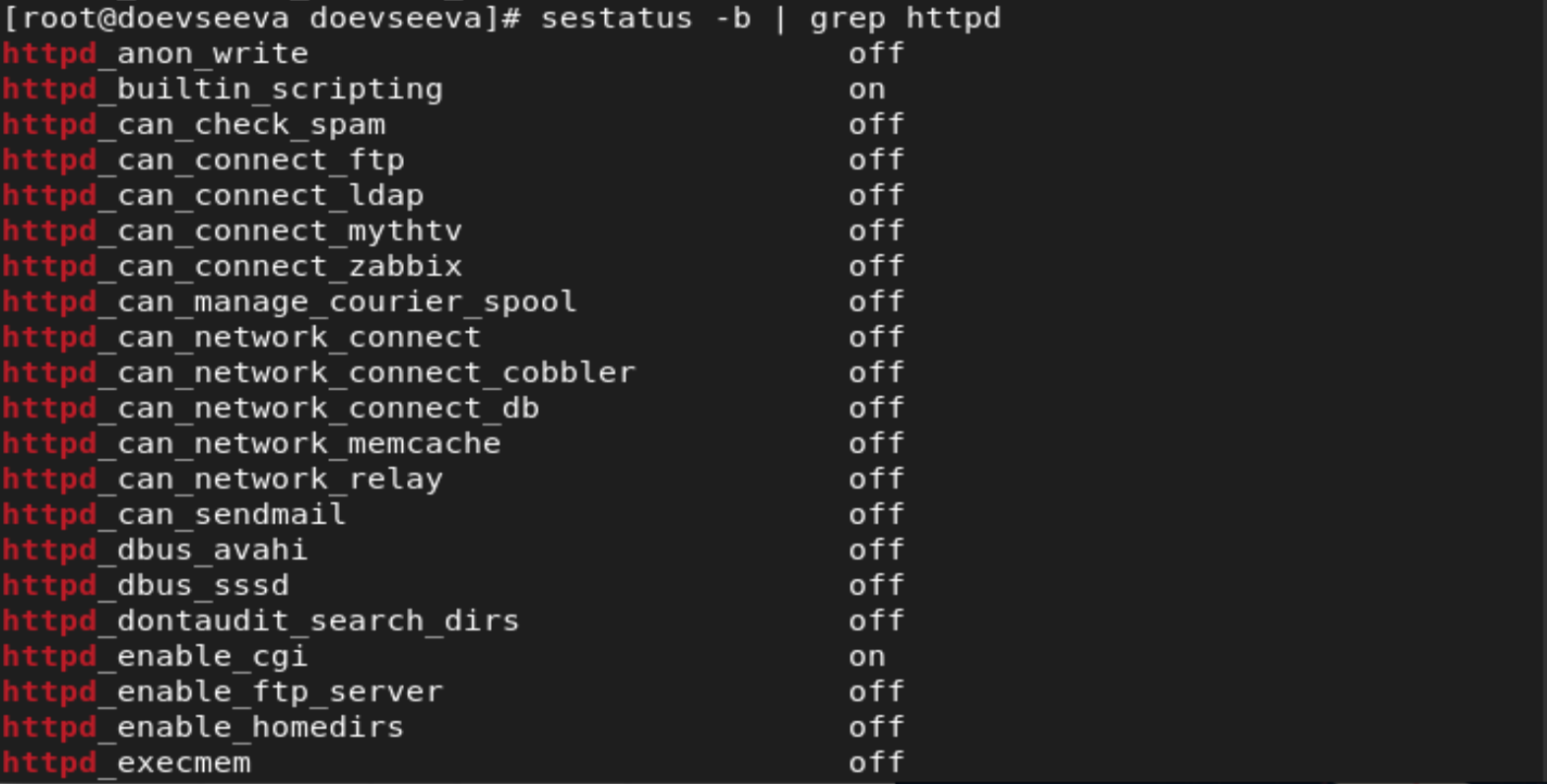
Найдем веб-сервер Apache в списке процессов и определим его контекст безопасности.



Определение контекста

Как мы видим, контекст веб-сервера Apache — system\_u:system\_r:httpd\_t:s0.

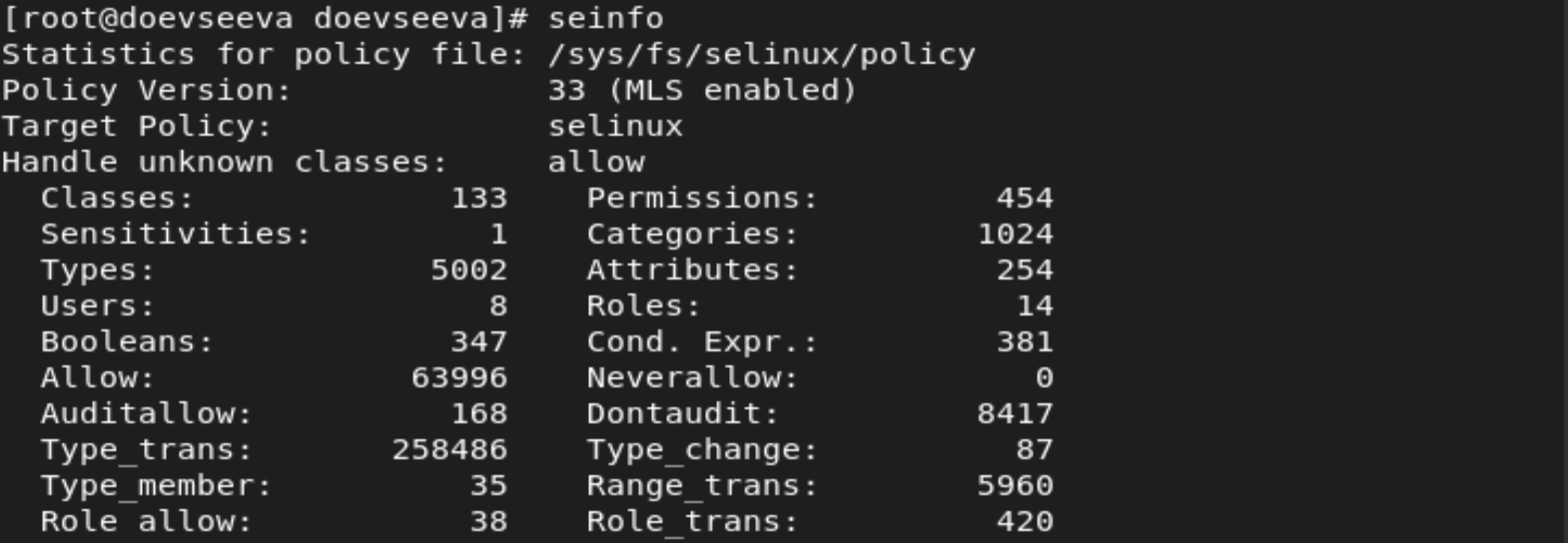
Посмотрим текущее состояние переключателей SELinux для Apache.



Просмотр состояния переключателей

Мы видим, что многие выключатели находятся в положении ‘off’.

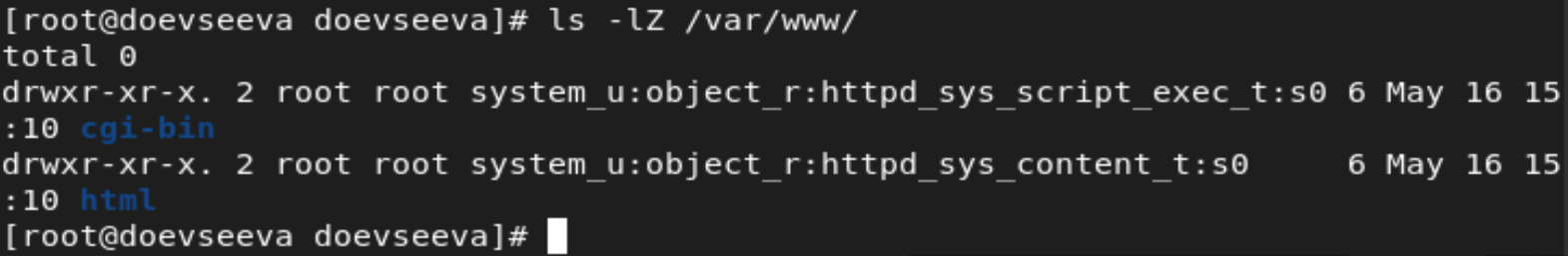
Посмотрим статистику по политике с помощью команды seinfo и определим множество пользователей, ролей и типов.



Просмотр статистики по политике

Здесь мы видим, что множество пользователей состоит из 4 пользователей, во множестве ролей находится 14 ролей, а во множестве типов — 5002 типа.

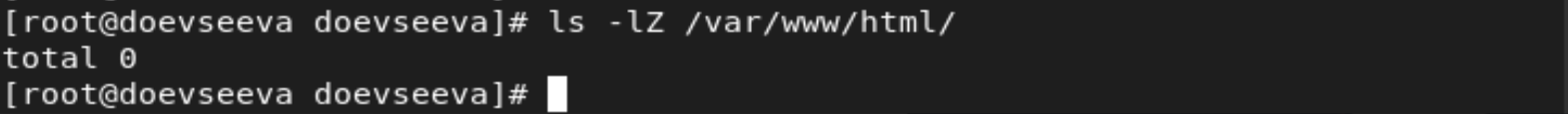
Далее посмотрим на содержимое директории /var/www.



Просмотр содержимого директории /var/www

Мы видим, что в директории находятся две поддиректории.

Также определим содержимое директории /var/www/html.



Просмотр содержимого директории /var/www/html

Мы видим, что в данной директории нет файлов или поддиректорий.

Чтобы определить круг пользователей, который разрешено создание файлов в директории /var/www/html, посмотрим на вывод команды на рис.6. Как мы видим, создание файлов в данной директории разрешено только для пользователя, который является владельцем директории.

Создадим от имени суперпользователя html-файл /var/www/html/test.html.



Создание html-файла



Содержимое файла

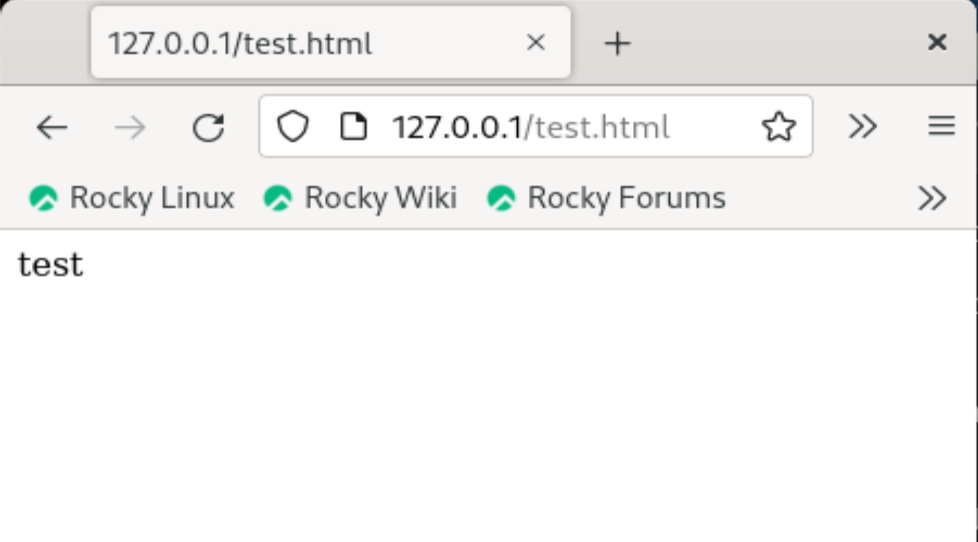
Далее проверим контекст созданного файла.



Определение контекста файла

Мы видим, что контекст, присваиваемый по умолчанию файлам, созданным в данной директории — это unconfined\_u:object\_r:httpd\_system\_content\_t:s0.

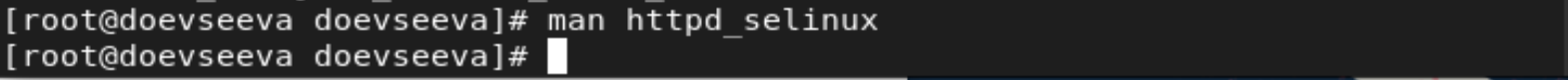
Обратимся к файлу через веб-сервер.



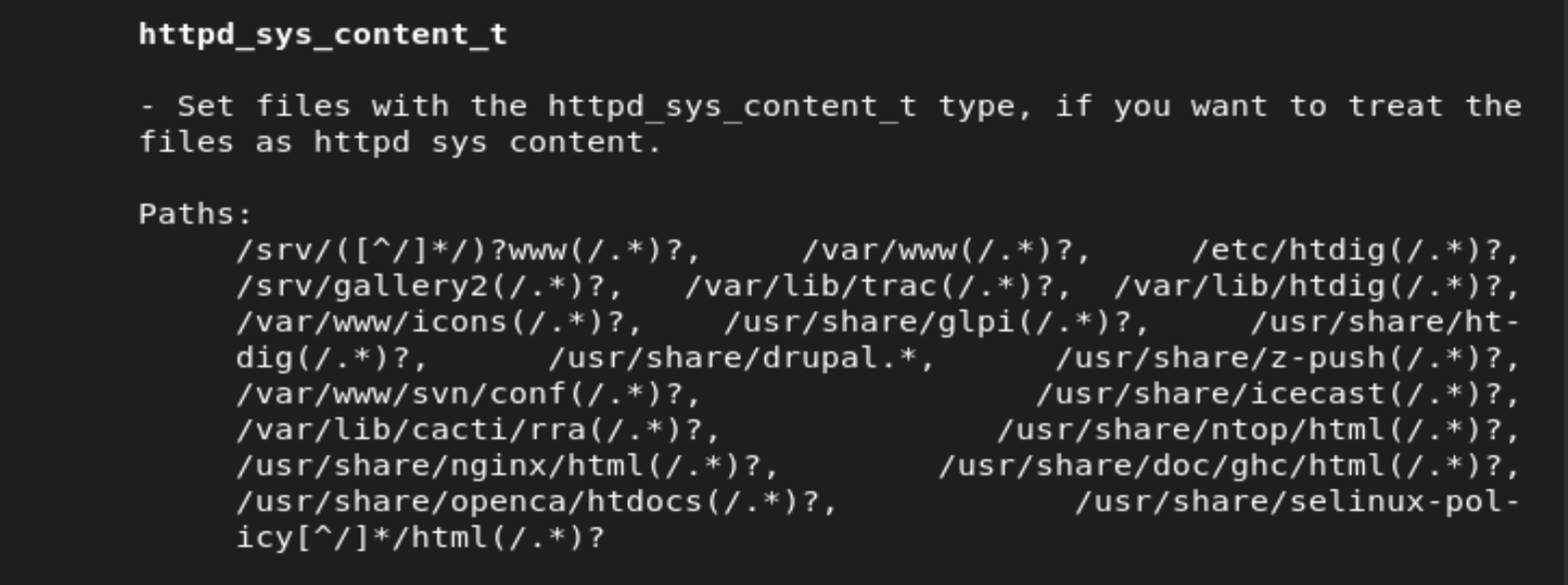
Обращение к файлу через веб-сервер

Как мы видим, файл успешно отображается.

Изучим справку man httpd\_selinux и найдем в ней контекст, соответствующий созданному файлу.

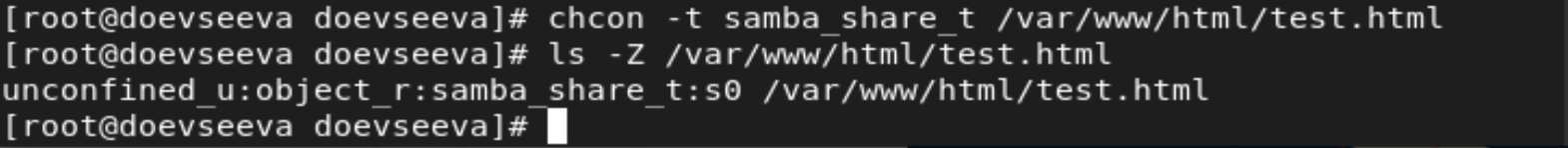


Просмотр справки httpd\_selinux



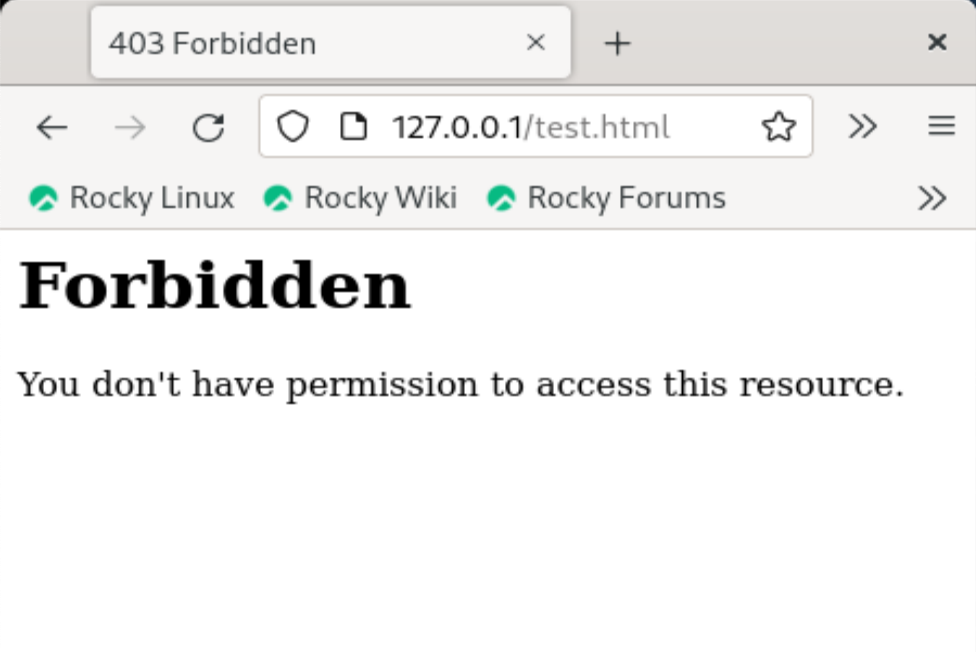
Соответствующий файлу контекст

Изменим контекст файла на такой, к которому процесс httpd не должен иметь доступа, и проверим успешность изменения контекста.



Изменение контекста

Еще раз попробуем получиться доступ к файлу через веб-сервер, а также проверим права доступа к файлу.



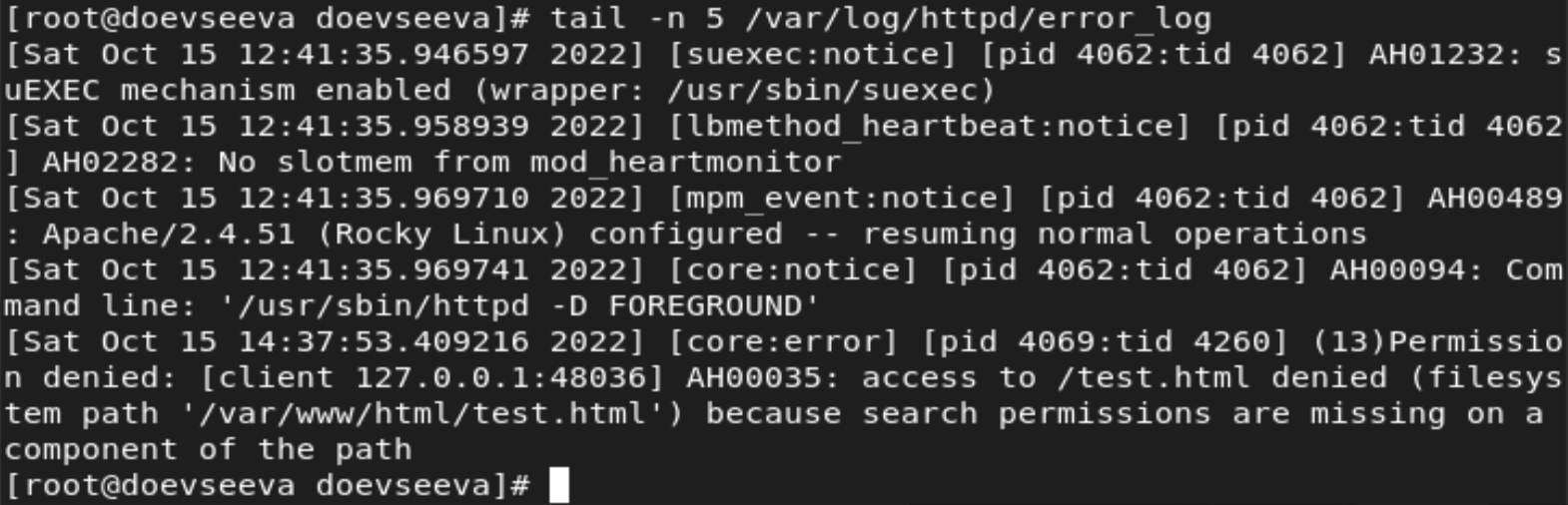
Попытка получения доступа к файлу



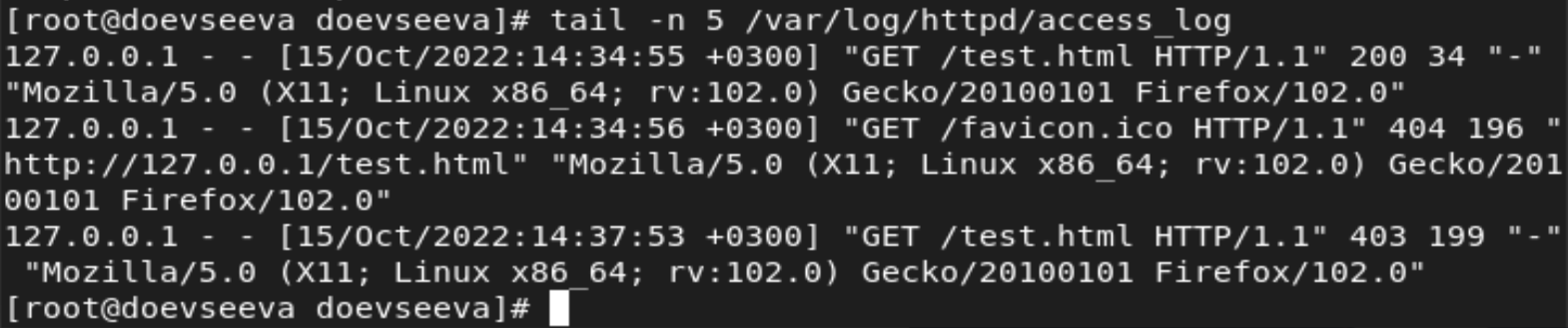
Проверка прав доступа

Мы видим, что на этот раз в доступе было отказано, и файл не был отображен. Несмотря на то, что доступ на чтение файла является открытым, файл не отображается из-за того, что ранее мы поменяли его контекст на такой, к которому нет доступа у процесса httpd.

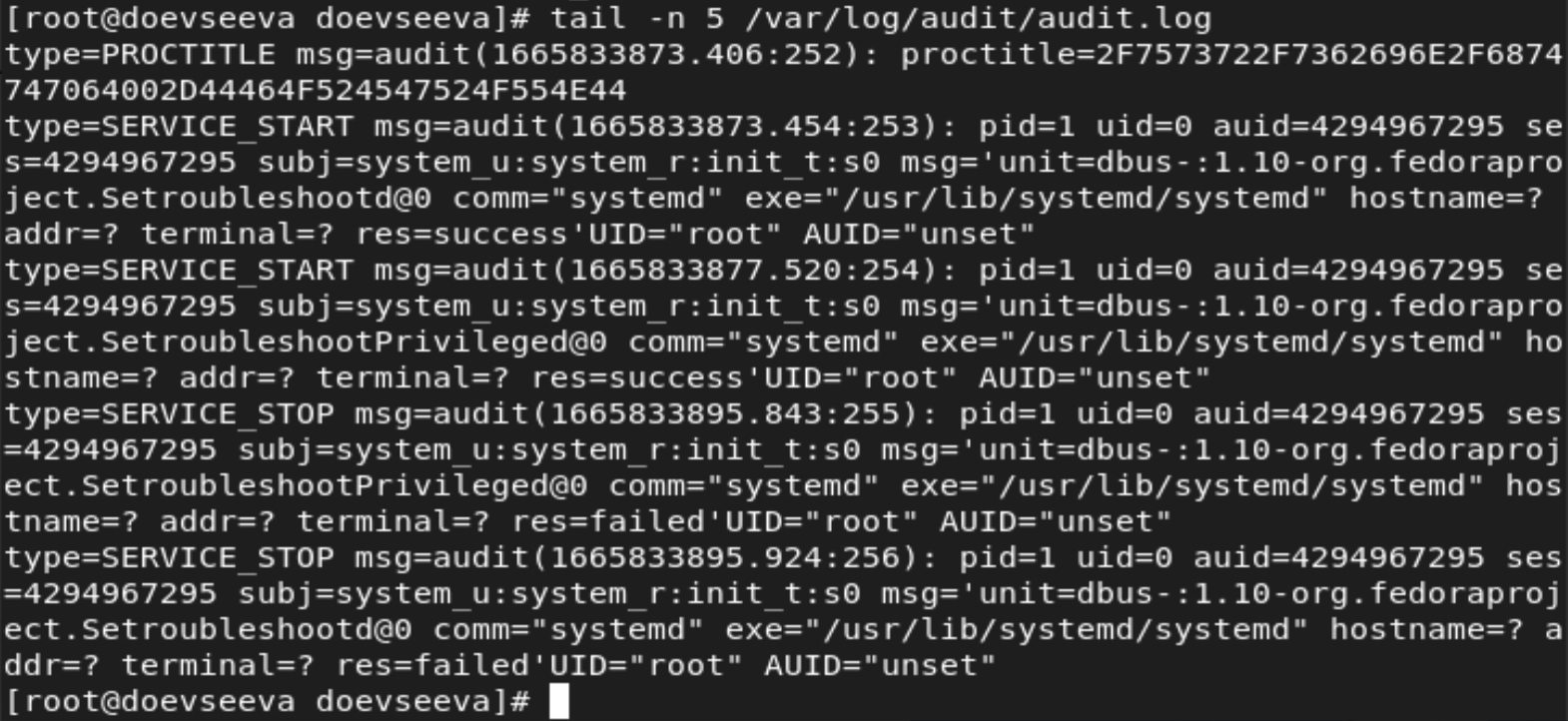
Просмотрим лог-файлы веб-сервера Apache, а также системный лог-файл.



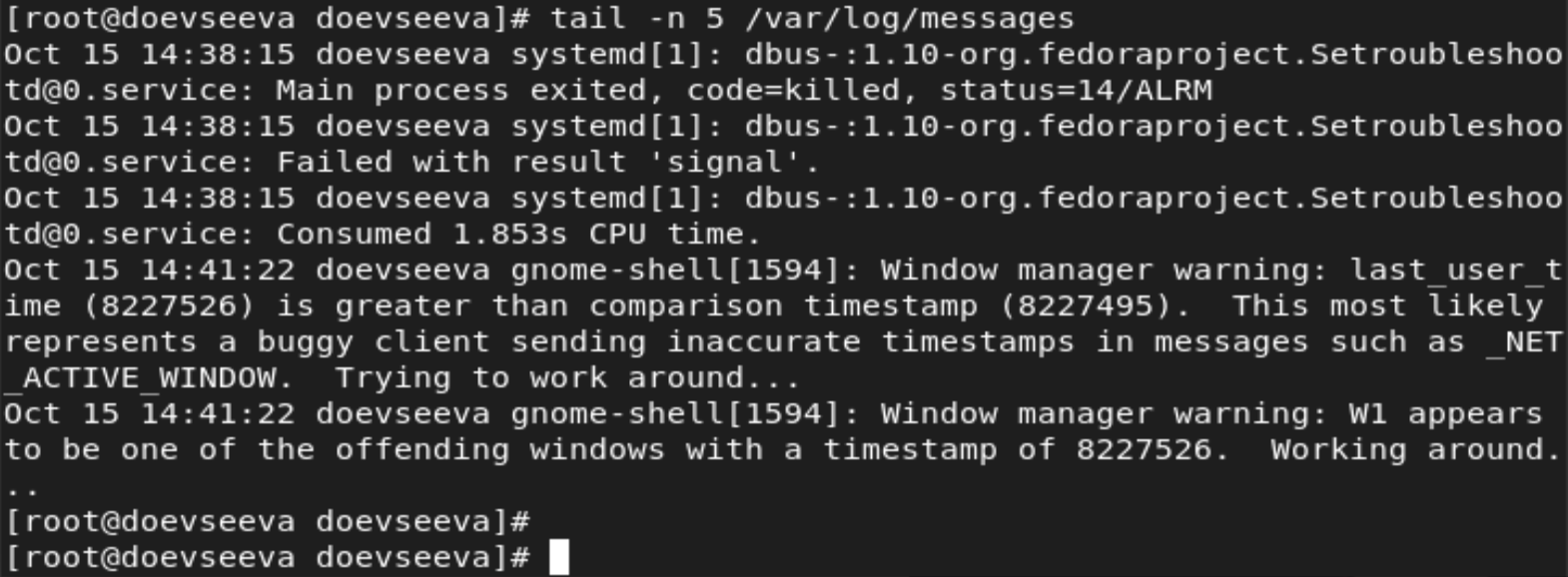
Просмотр лог-файлов (1)



Просмотр лог-файлов (2)



Просмотр лог-файлов (3)

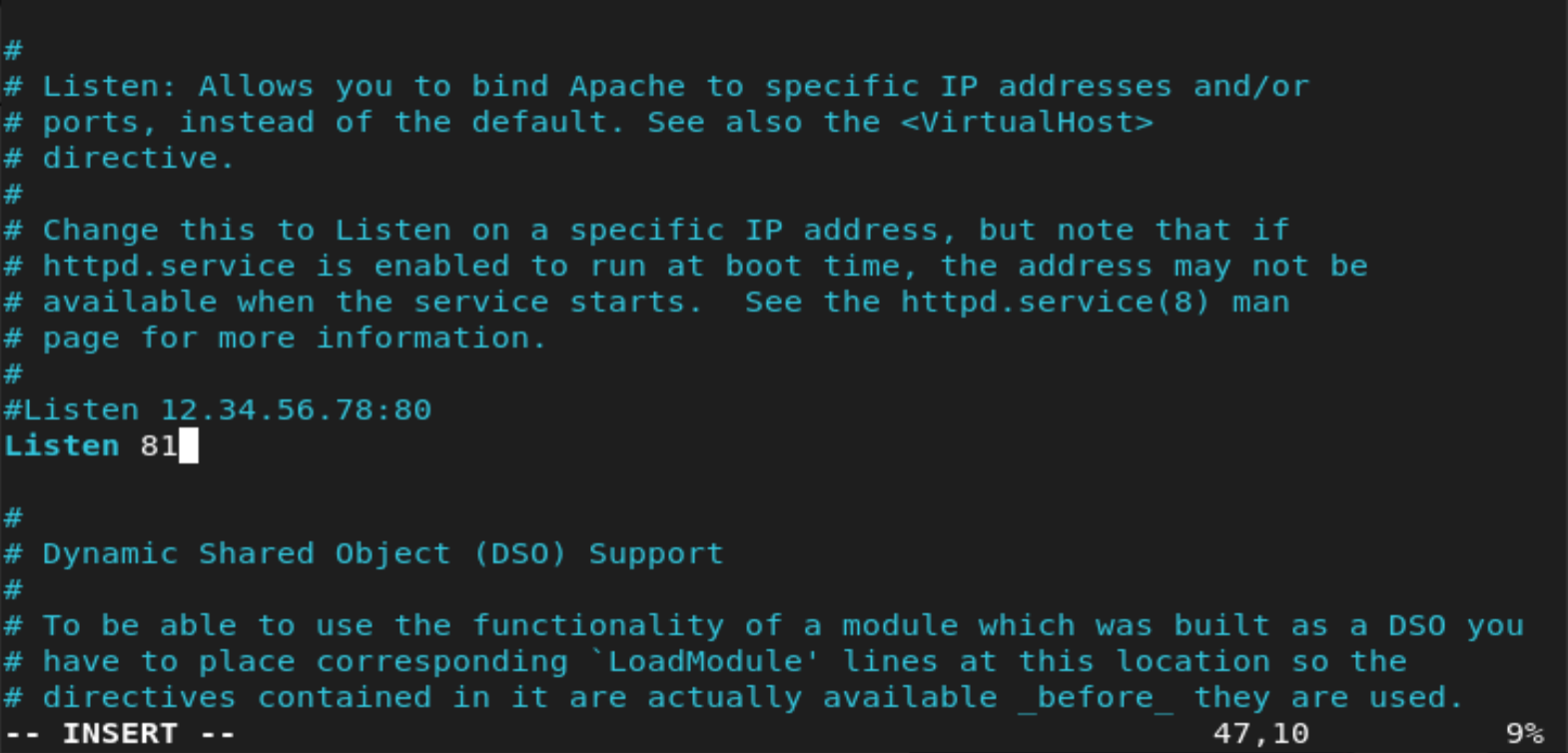


Просмотр системного лог-файла

Теперь попробуем запустить веб-сервер Apache на прослушивание TCP-порта 81, для чего внесем изменения в файл /etc/httpd/conf/httpd.conf.

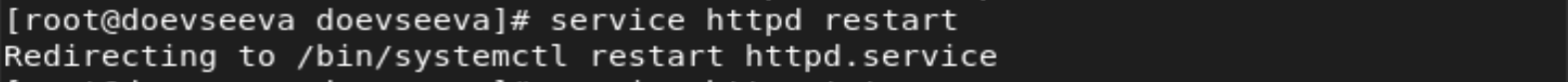


Открытие файла /etc/httpd/conf/httpd.conf



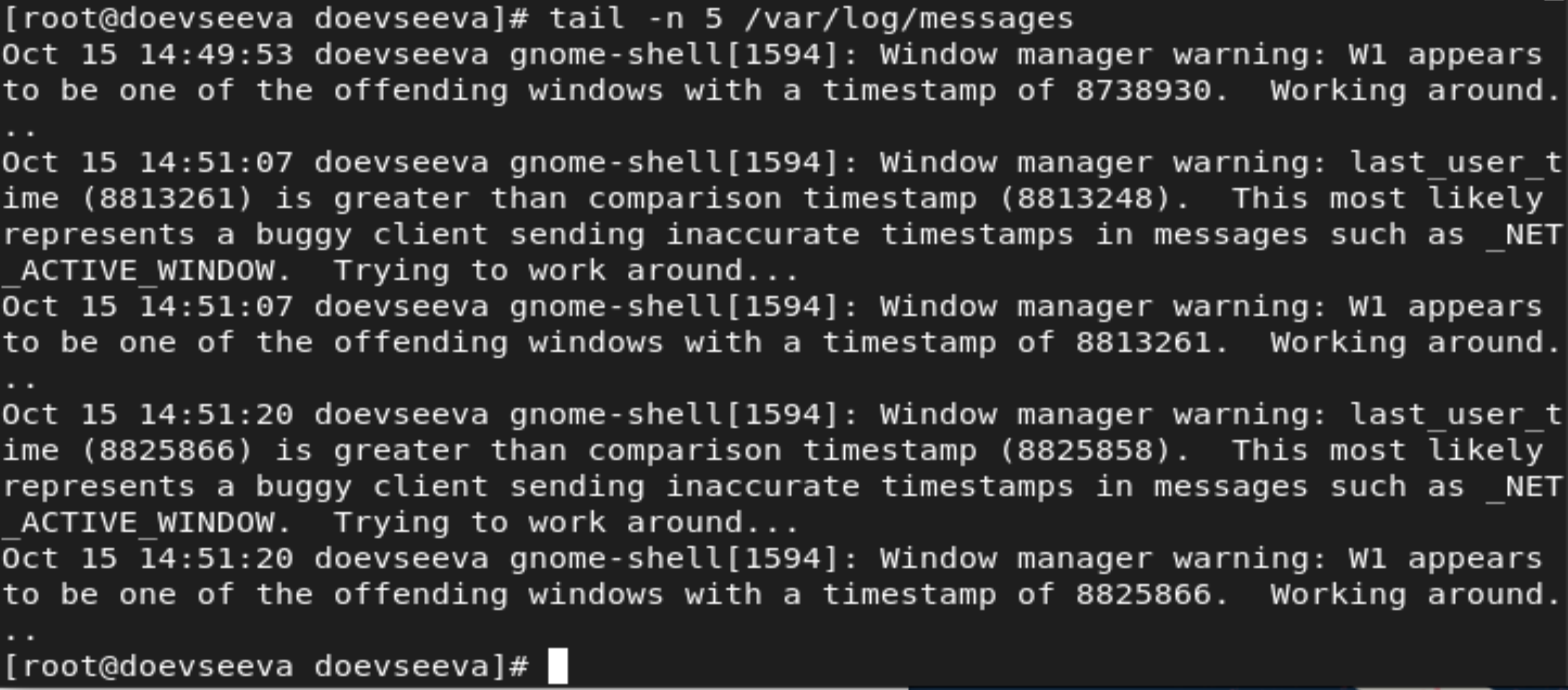
Внесение изменений в файл

Выполним перезапуск веб-сервера Apache.

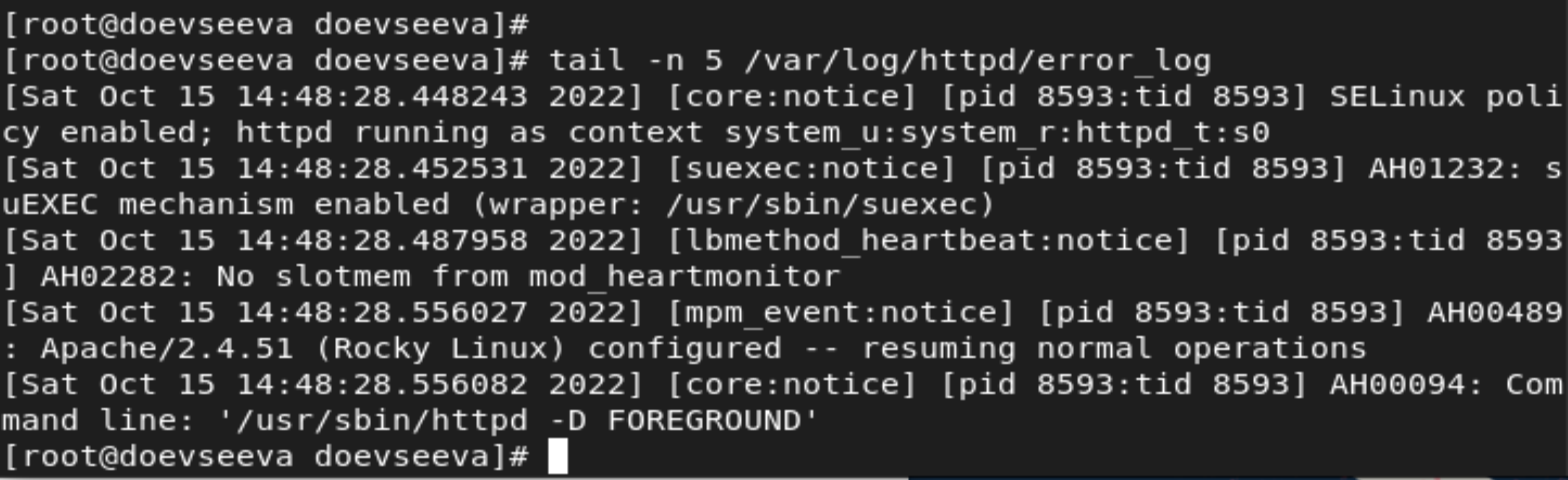


Перезапуск веб-сервера Apache

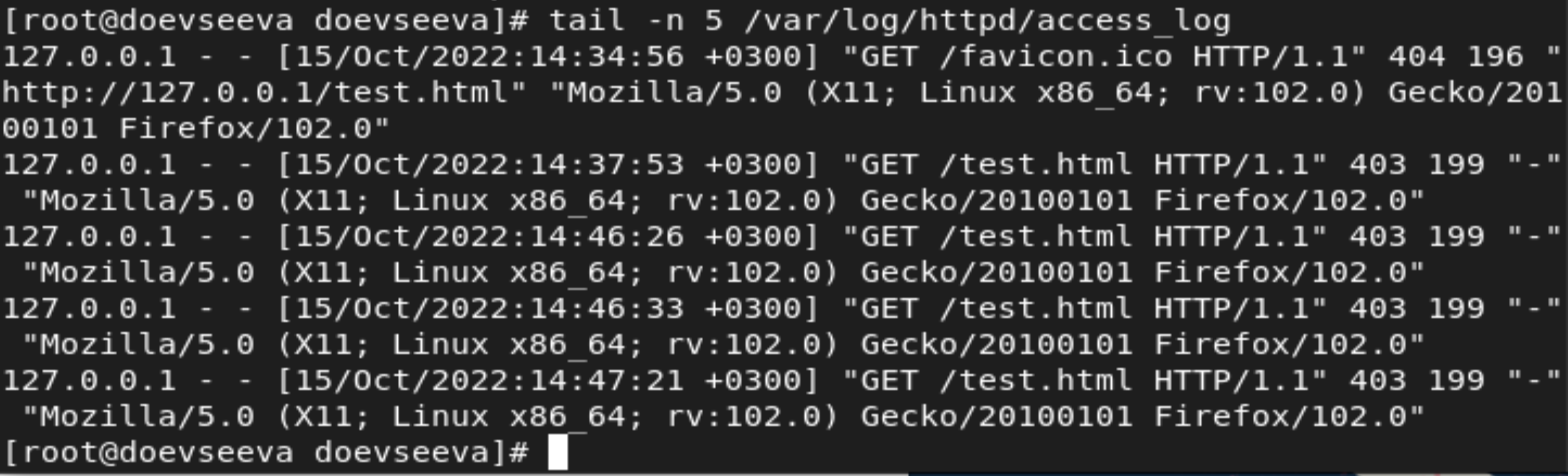
Далее просмотрим лог-файлы.



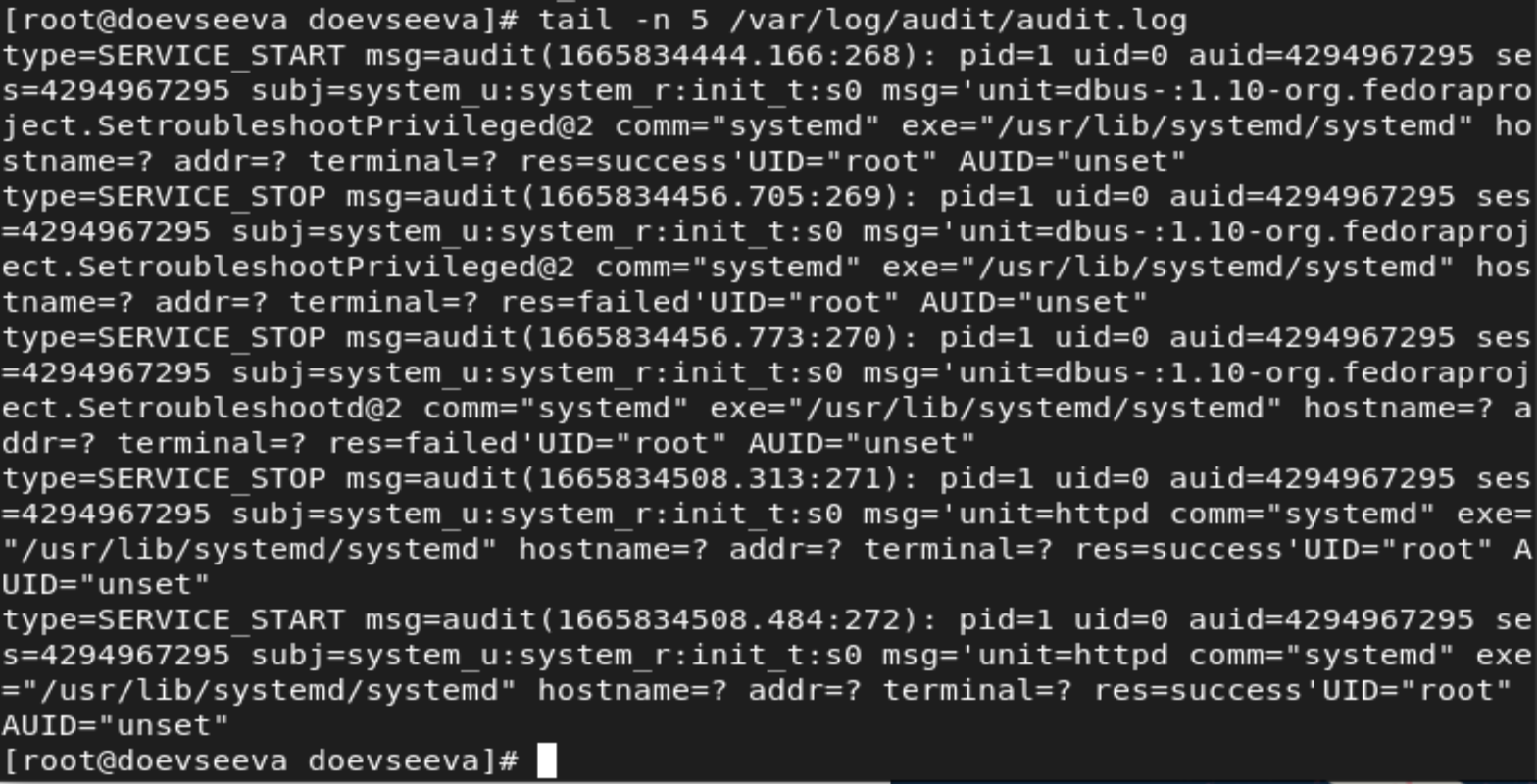
Просмотр системного лог-файла



Просмотр лог-файлов (1)



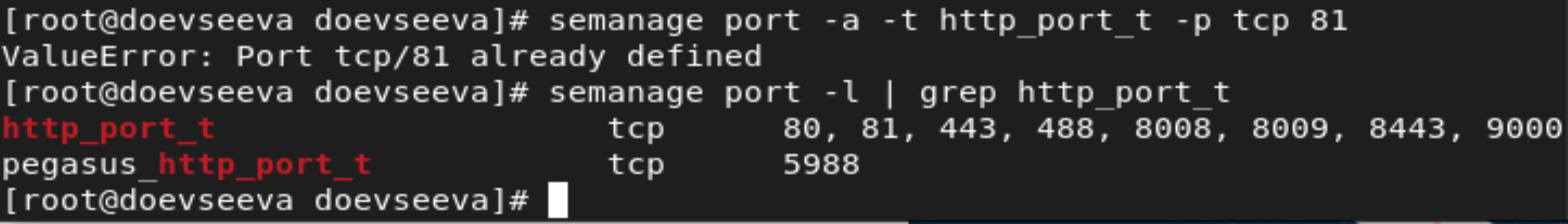
Просмотр лог-файлов (2)



Просмотр лог-файлов (3)

Мы можем заметить, что в каждом из лог-файлов появились записи.

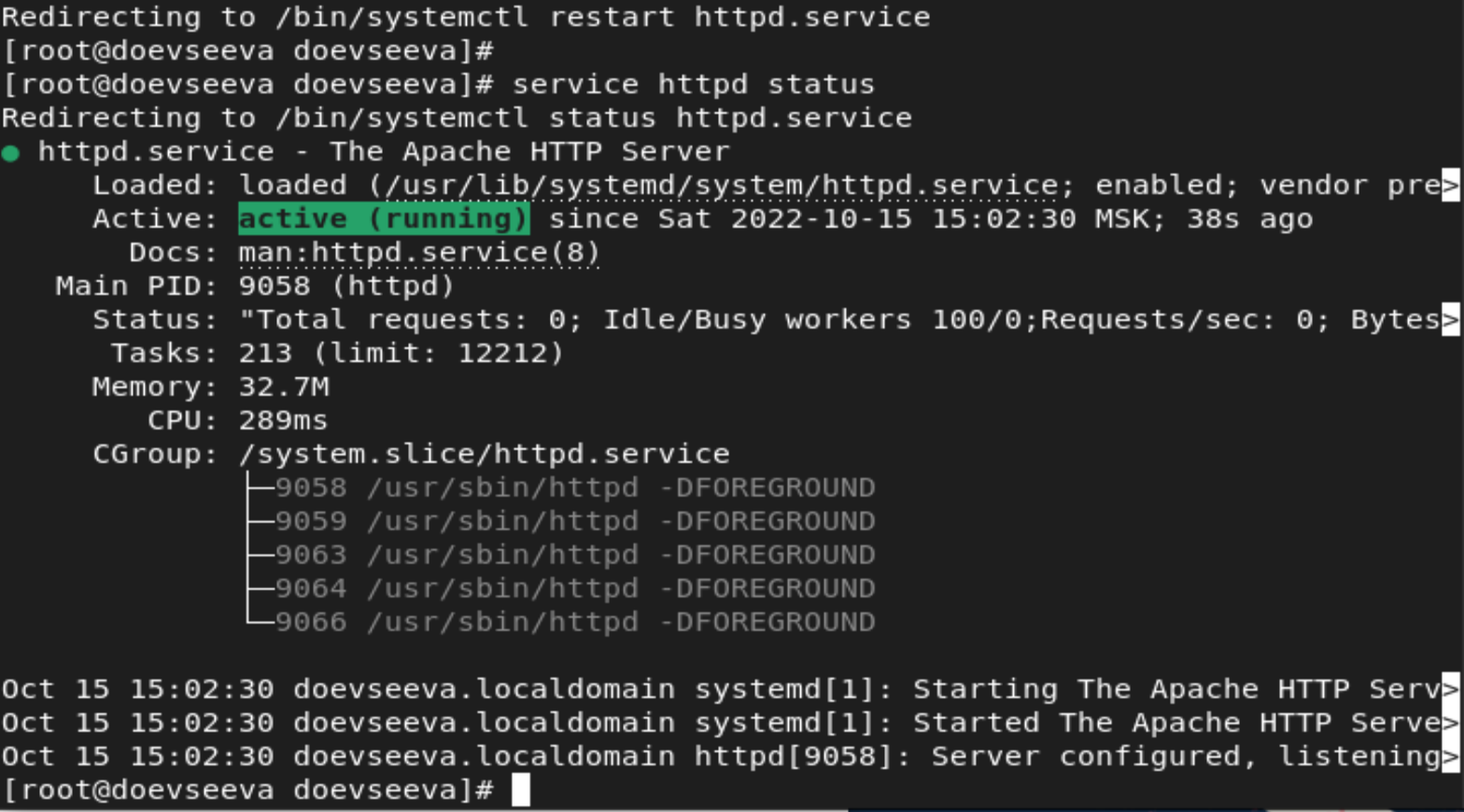
Выполним команду semanage port и проверим список портов.



Выполнение привязки и просмотр списка портов

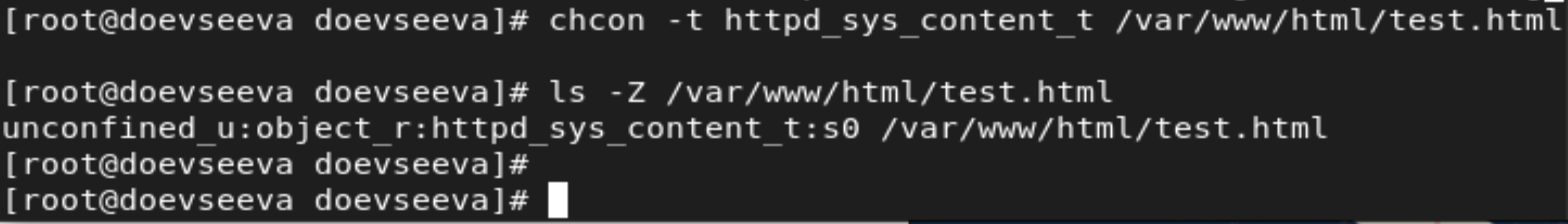
Мы видим, что в нашей системе порт 81 изначально находился в списке.

Еще раз попробуем перезапустить веб-сервер Apache.

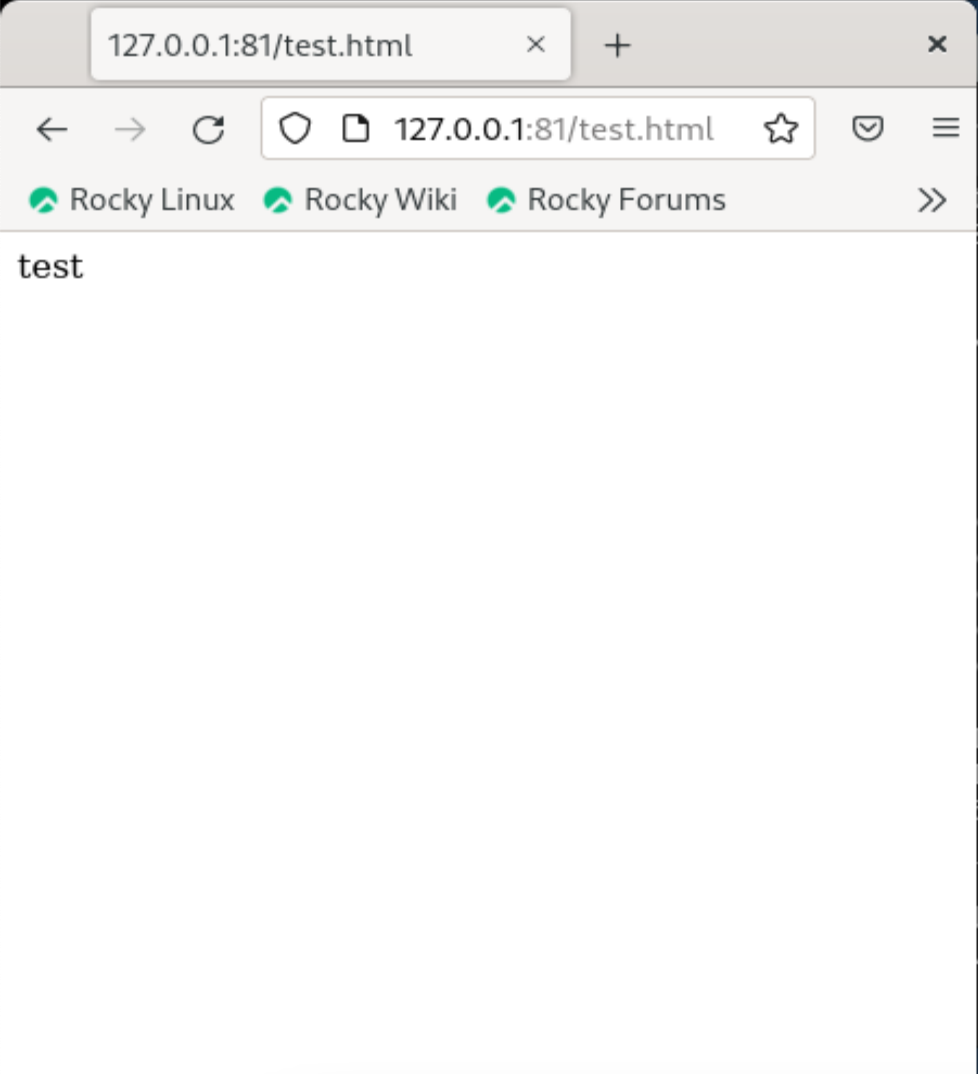


Перезапуск веб-сервера Apache

Вернем изначальный контекст созданному файлу, и попробуем получить к нему доступ через веб-сервер.



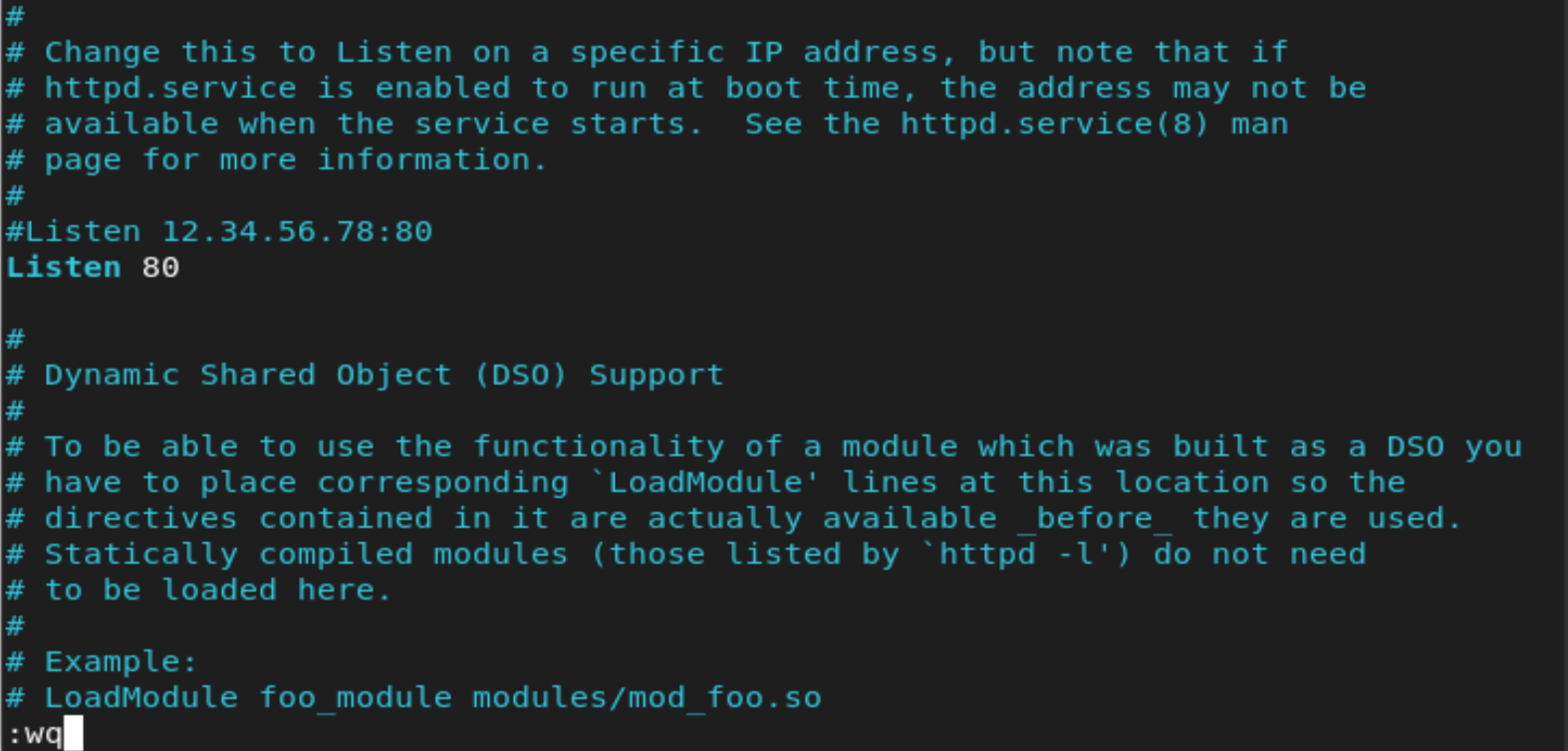
Возвращение изначального контекста



Получение доступа к файлу через веб-сервер

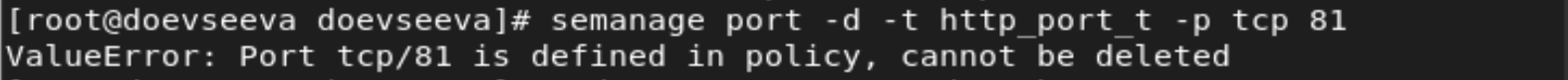
Мы видим, что содержимое файла успешно отобразилось.

Исправим конфигурационный файл Apache, вернув ‘Listen 80’.



Исправление конфигурационного файла

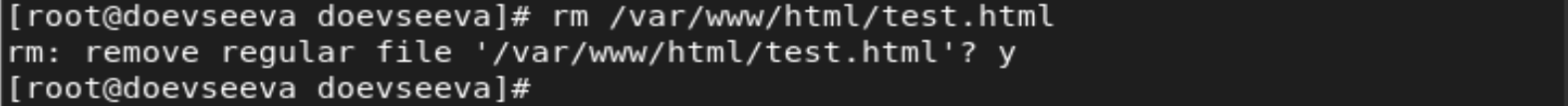
Удалим привязку к 81 порту.



Удаление привязки к порту

Как мы видим, в нашей системе мы не можем удалить привязку к данному порту.

Удалим созданный html-файл.



Удаление html-файла

# Выводы

В результате проделанной работы мы развили навыки администрирования ОС Linux, ознакомились с технологией SELinux, а также проверили работу SELinux на практике совместно с веб-сервером Apache.

# Список литературы

* Методические материалы к лабораторной работе, представленные на сайте “ТУИС РУДН” https://esystem.rudn.ru/