Лабораторная работа №6

Мандатное разграничение прав в Linux

Пономарева Лилия Михайловна

Содержание

# Цель работы

Развить навыки администрирования ОС Linux. Получить первое практическое знакомство с технологией SELinux1. Проверить работу SELinx на практике совместно с веб-сервером Apache. [[1]](#список-литературы)

# Теоретическое введение

Security Enhanced Linux, или SELinux – это усовершенствованный механизм контроля доступа, встроенный в большинство современных дистрибутивов Linux.

SELinux реализует так называемый MAC (Mandatory Access Control). Это разграничение контроля внедряется поверх того, что уже есть в каждом дистрибутиве Linux, DAC (Discretionary Access Control).

По сути, в традиционной модели избирательного управления доступом (DAC), хорошо реализованы только два уровня доступа — пользователь и суперпользователь. Нет простого метода, который позволил бы устанавливать для каждого пользователя необходимый минимум привилегий.

Основные термины, использующиеся в SELinux:

*Домен* — список действий, которые может выполнять процесс. Обычно в качестве домена определяется минимально-возможный набор действий, при помощи которых процесс способен функционировать. Таким образом, если процесс дискредитирован, злоумышленнику не удастся нанести большого вреда.

*Роль* — список доменов, которые могут быть применены. Если какого-то домена нет в списке доменов какой-то роли, то действия из этого домена не могут быть применены.

*Тип* — набор действий, которые допустимы по отношения к объекту. Тип отличается от домена тем, что он может применяться к пайпам, каталогам и файлам, в то время как домен применяется к процессам.

Контекст безопасности — все атрибуты SELinux — роли, типы и домены.

Система SELinux может работать в любом из трех доступных режимов: - Enforcing - Permissive - Disabled

В режиме *enforcing* SELinux применяет свою политику в системе Linux и следит за тем, чтобы все попытки несанкционированного доступа со стороны пользователей и процессов были запрещены. Отказы в доступе регистрируются в соответствующих логах.

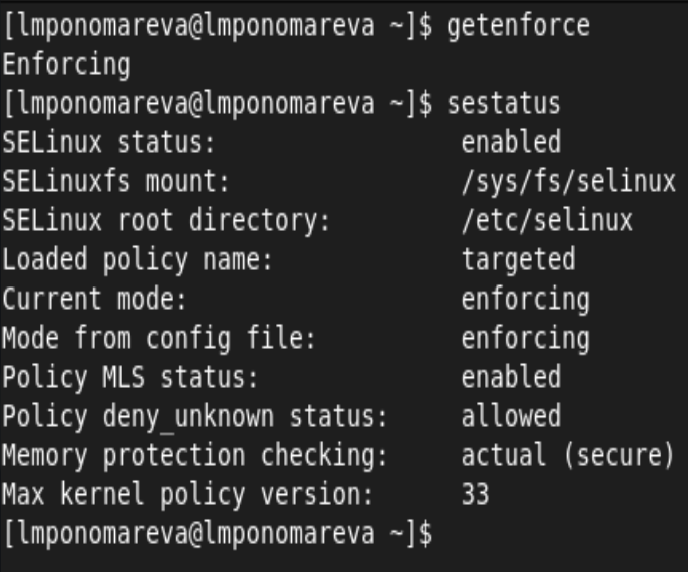
Режим *permissive* – это такое полуоткрытое состояние: в этом режиме SELinux не применяет свою политику, поэтому не блокирует доступ. Однако любое нарушение политики будет зарегистрировано в логах.

Режим *disabled* – система отключена.

SELinux по-умолчанию работает в режиме Enforcing, когда любые действия, кроме разрешенных, автоматически блокируются, каждая программа, пользователь или сервис обладают только теми привилегиями, которые необходимы им для функционирования, но не более того. Это довольно жесткая политика, которая обладает как плюсами — наибольший уровень информационной безопасности, так и минусами — конфигурирование системы в таком режиме сопряжено с большими трудозатратами системных администраторов, к тому же, велик риск того, что пользователи столкнутся с ограничением доступа, если захотят использовать систему хоть сколько-нибудь нетривиальным образом.

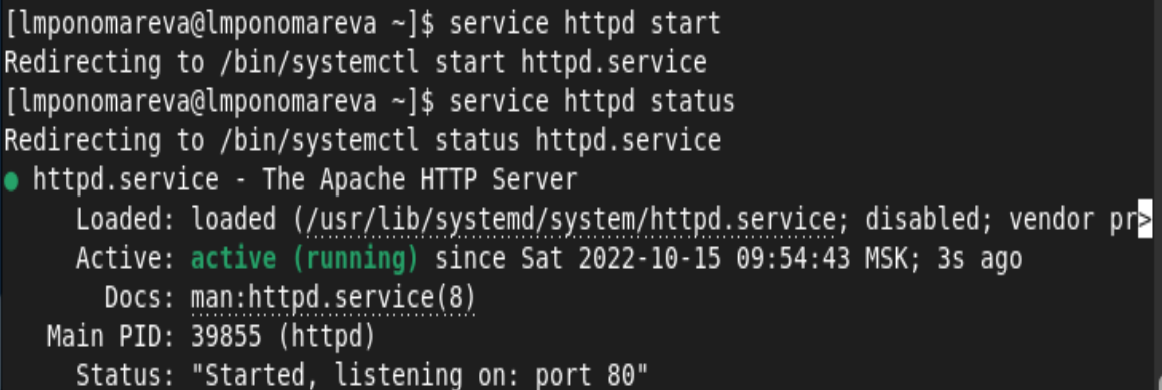
# Выполнение лабораторной работы

1. Вошла в систему с полученными учётными данными и убедилась, что SELinux работает в режиме enforcing политики targeted с помощью команд getenforce и sestatus. ([рис. 1](../../image/1.png))



Режим SELinux

1. Запустила веб-сервер. Обратилась с помощью браузера к веб-серверу, запущенному на компьютере, и убедилась, что последний работает: service httpd status. ([рис. 2](../../image/2.png)).



Проверка работы веб-сервера

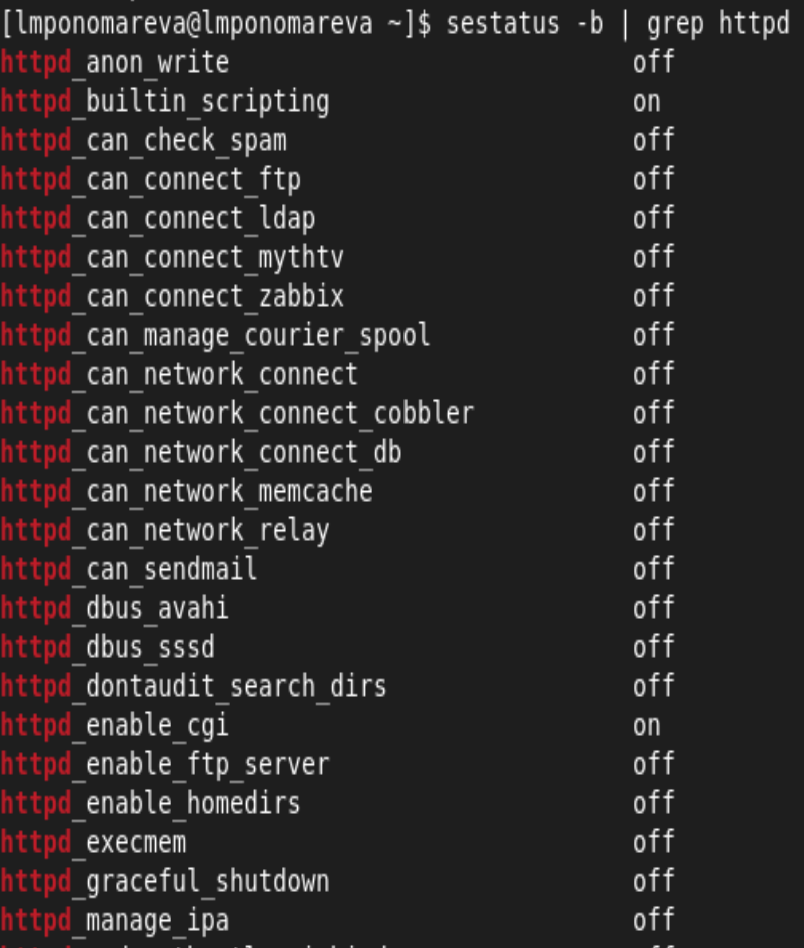
1. Нашла веб-сервер Apache в списке процессов. ([рис. 3](../../image/3.png)).



Список процессов

Контекст безопасности: system\_u:system\_r:httpd\_t:s0

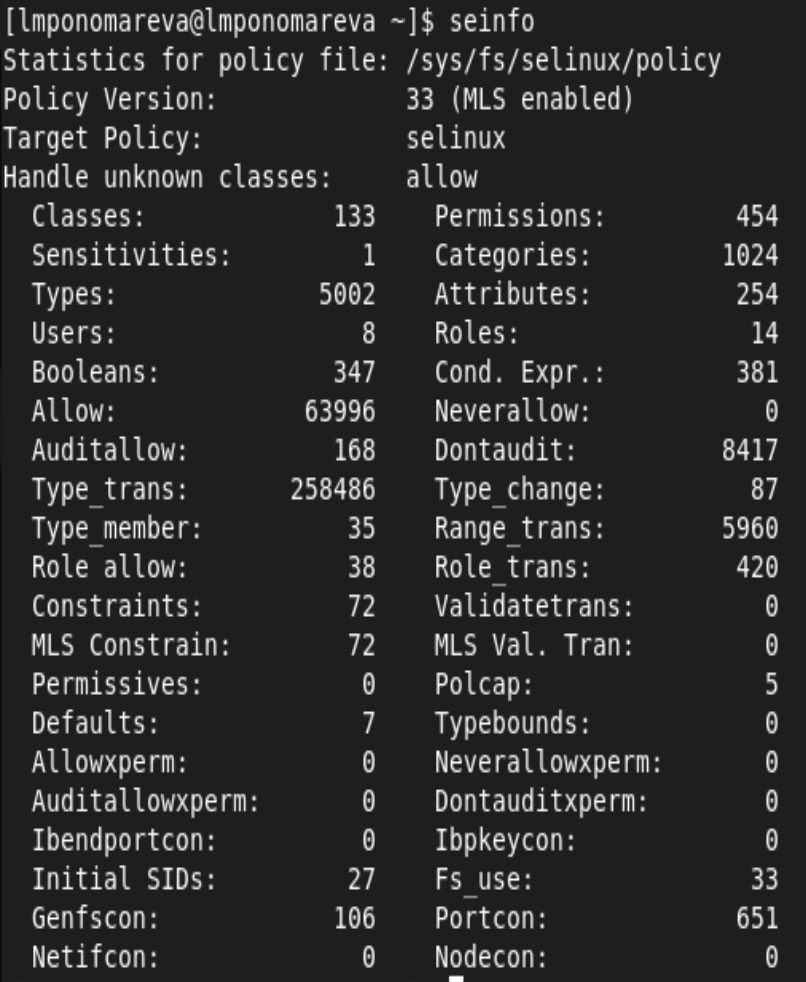
1. Посмотрела текущее состояние переключателей SELinux для Apache с помощью команды sestatus -b | grep httpd. ([рис. 4](../../image/4.png)).



Состояние переключателей SELinux

Многие из них находятся в положении «off»

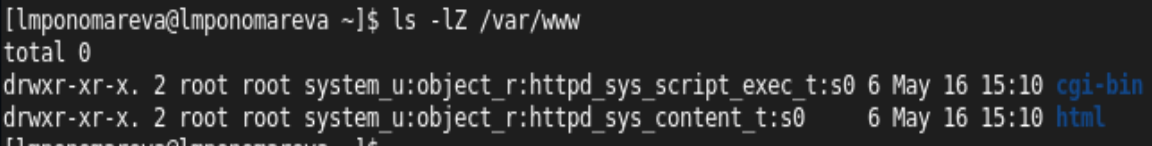
1. Посмотрела статистику по политике с помощью команды seinfo. ([рис. 5](../../image/5.png)).



Статистика SELinux

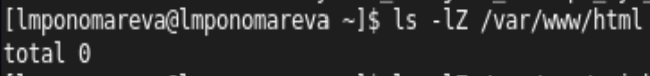
Множество пользователей - 8  
Ролей - 14 Типов - 5002

1. Определила тип файлов и поддиректорий, находящихся в директории /var/www, с помощью команды ls -lZ /var/www ([рис. 6](../../image/6.png)).



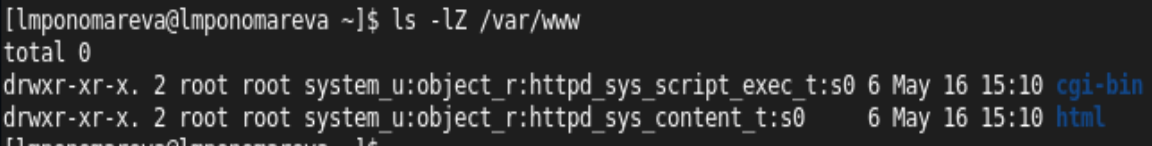
Тип поддиректорий в директории /var/www

1. Определила тип файлов, находящихся в директории /var/www/html: ls -lZ /var/www/html ([рис. 7](../../image/7.png)).



Директория /var/www/html

1. Определила круг пользователей, которым разрешено создание файлов в директории /var/www/html. ([рис. 8](../../image/6.png)).



Право на создание файлов

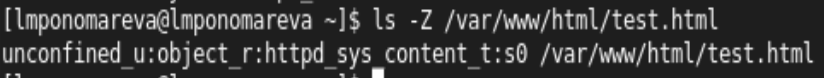
Создавать файлы в директории может только её владелец.

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. 9](../../image/8.png)).



HTML-файл /var/www/html/test.html

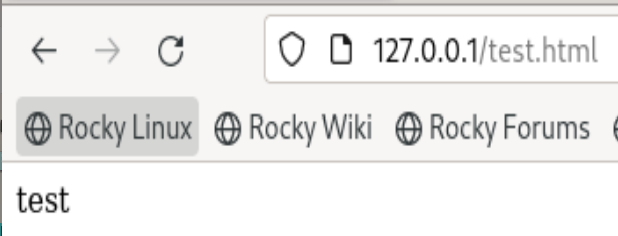
1. Проверила контекст созданного файла. ([рис. 10](../../image/9.png)).



Контекст файла

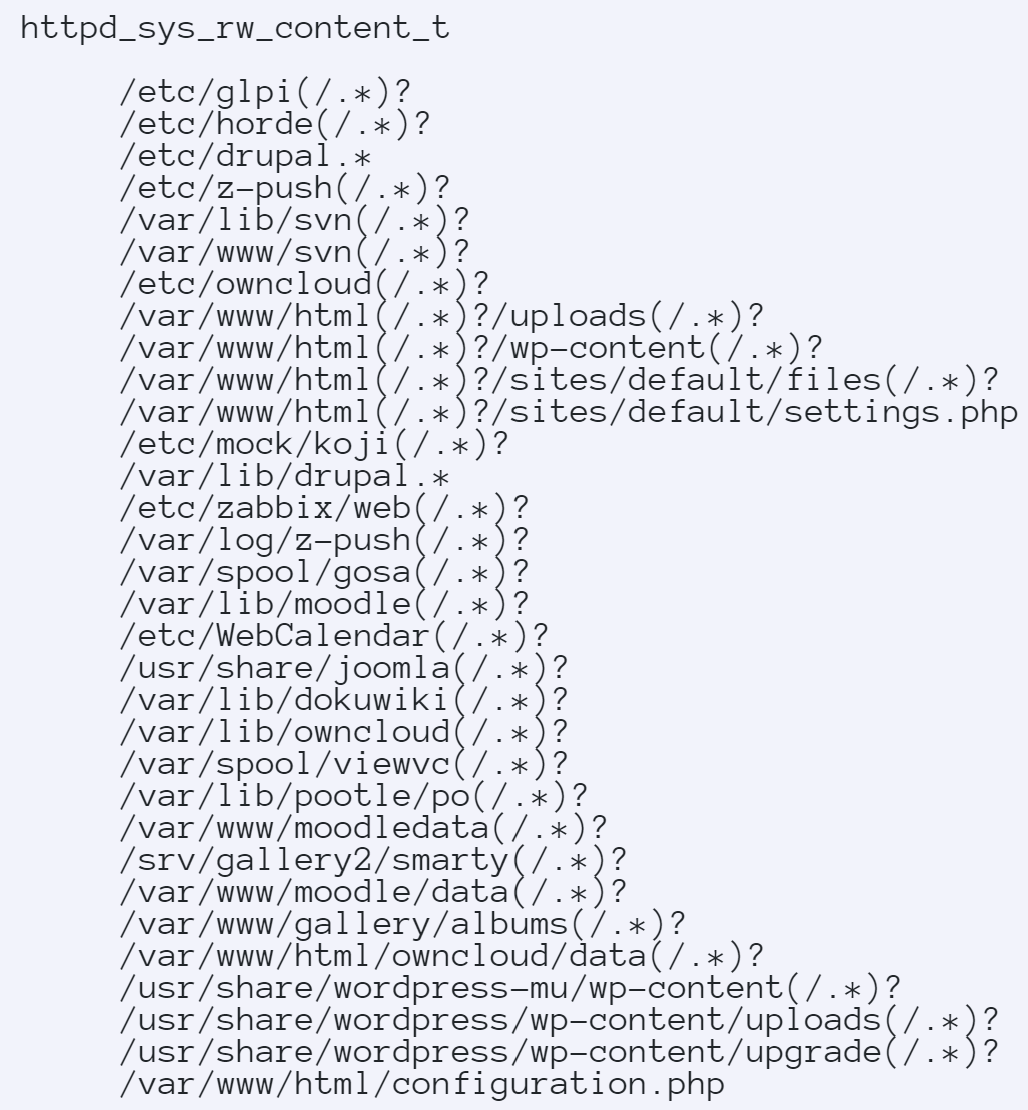
Контекст безопасности (по умолчания для новых файлов в директории): nconfined\_u:object\_r:httpd\_sys\_content\_t:s0

1. Обратилась к файлу через веб-сервер, введя в браузере адрес http://127.0.0.1/test.html. ([рис. 11](../../image/10.png)).



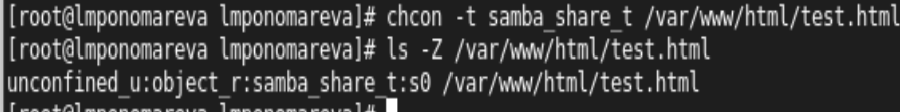
Отображение файла test.html

1. Изучила справку man httpd\_selinux ([рис. 12](../../image/12.png)).



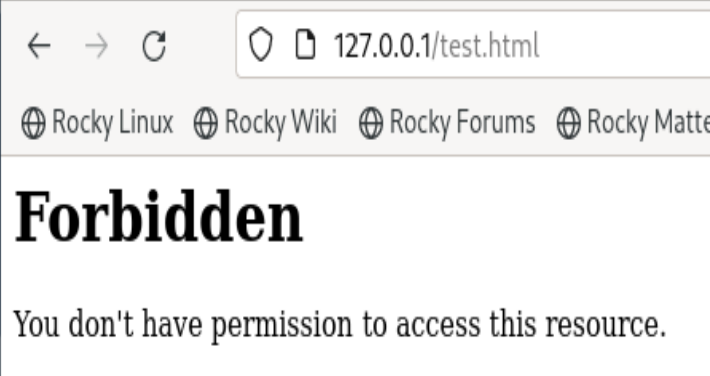
Справка man httpd\_selinux

1. Изменила контекст файла /var/www/html/test.html с httpd\_sys\_content\_t на тот, к которому процесс httpd не имеет доступ (samba\_share\_t) ([рис. 13](../../image/13.png)).



Изменение контекста файла

1. Попробовала получить доступ к файлу через веб-сервер, введя в браузере адрес http://127.0.0.1/test.html. ([рис. 14](../../image/14.png)).



Доступ через веб-сервер

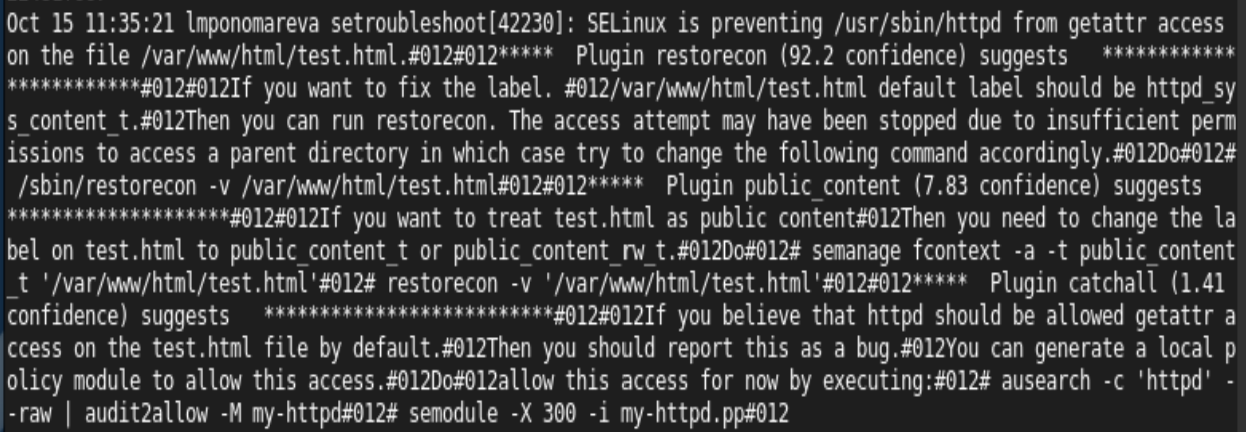
Файл не отображен, так как к заданному типу контекста httpd не имеет доступа.

1. Просмотрела log-файлы веб-сервера Apache. ([рис. 15](../../image/17.png))



Чтение файла readfile.c программой readfile

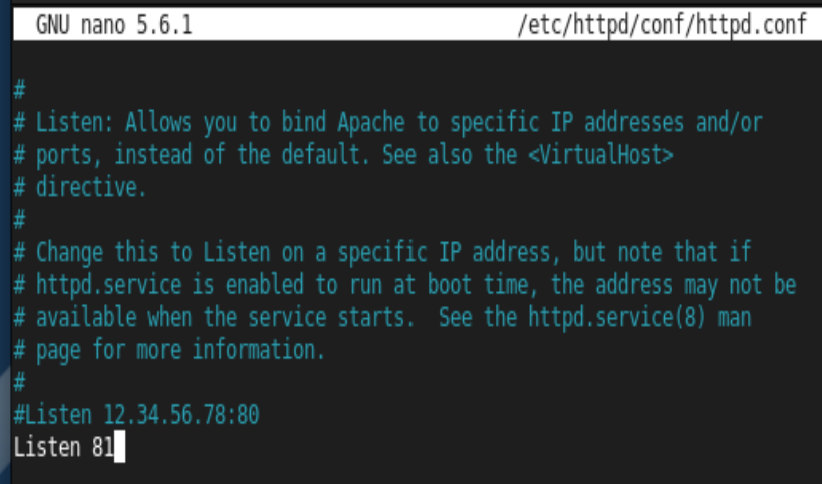
Посмотрела системный лог-файл ([рис. 16](../../image/16.png)).



Чтение файла /etc/shadow программой из readfile

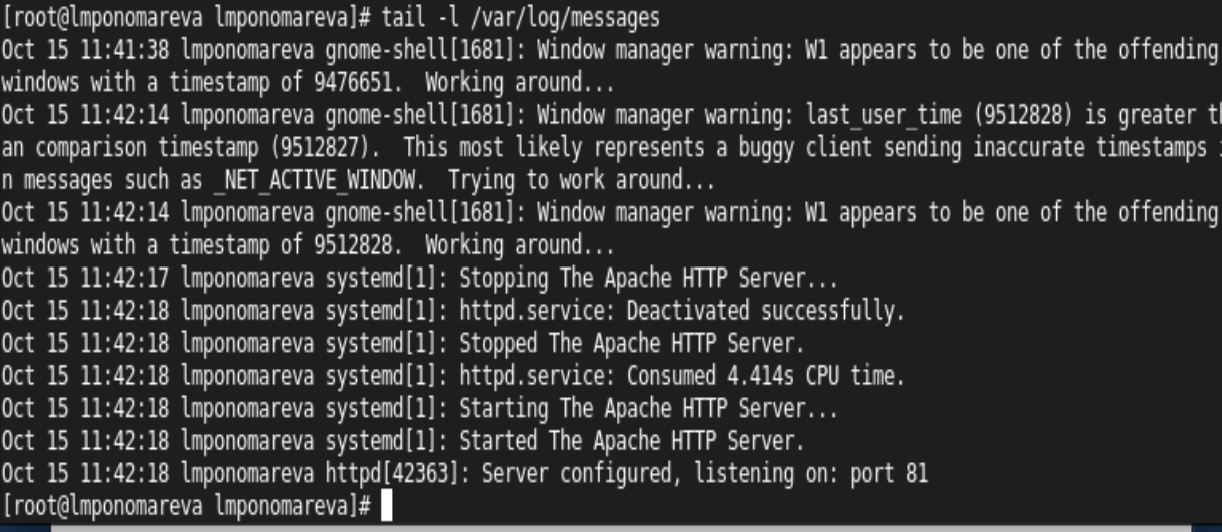
Можем видеть как отображаются ошибки.

1. Попробовала запустить веб-сервер Apache на прослушивание ТСР-порта 81 ([рис. 17](../../image/18.png)).

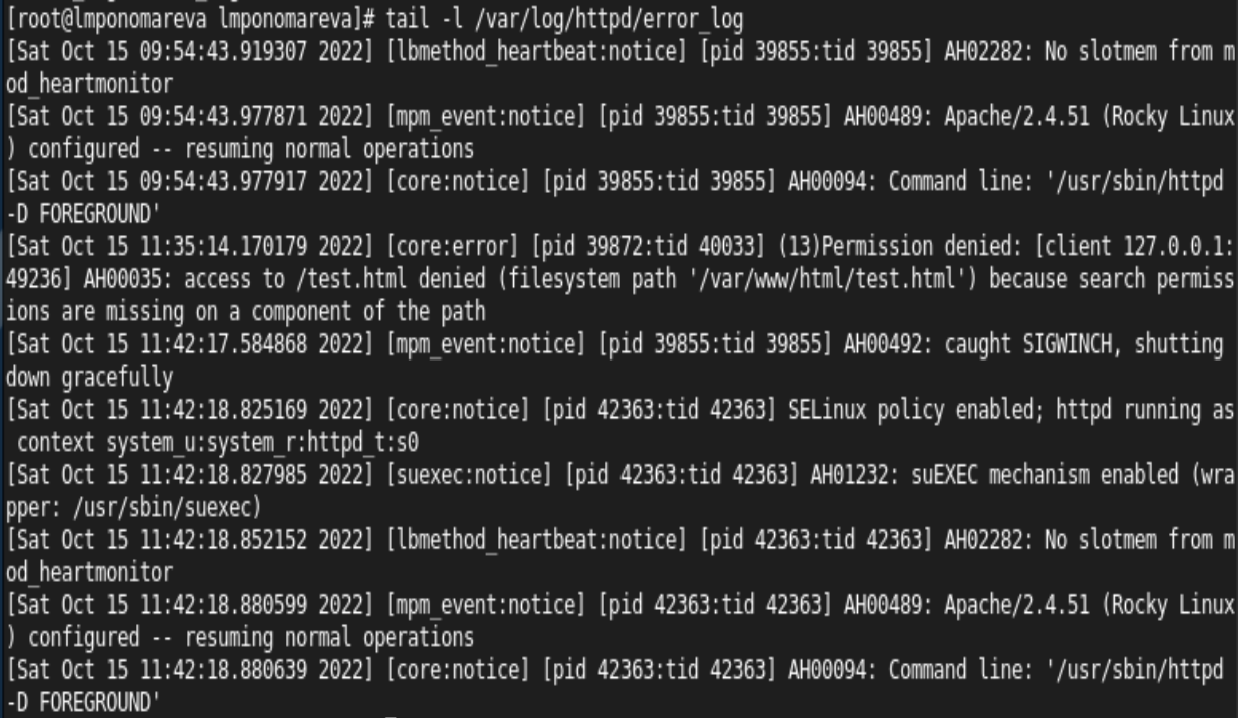


Включение прослушивания 81 порта

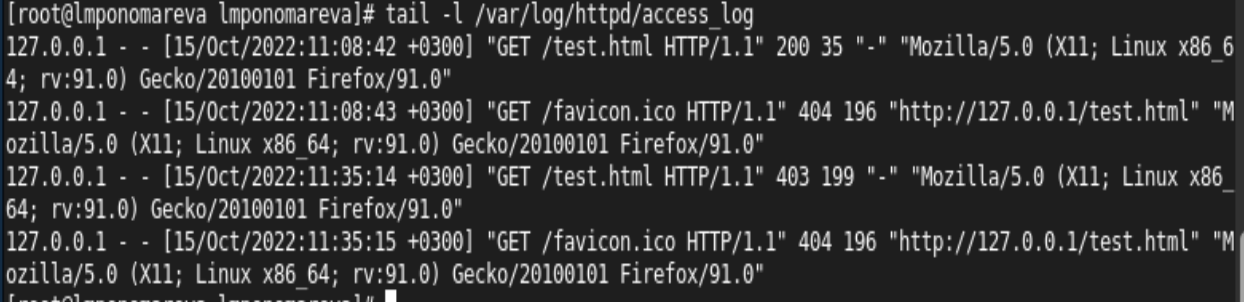
1. Выполнила перезапуск веб-сервера Apache и посмотрела лог-файлы ([рис. 18](../../image/19.png)-[рис. 21](../../image/24.png)).



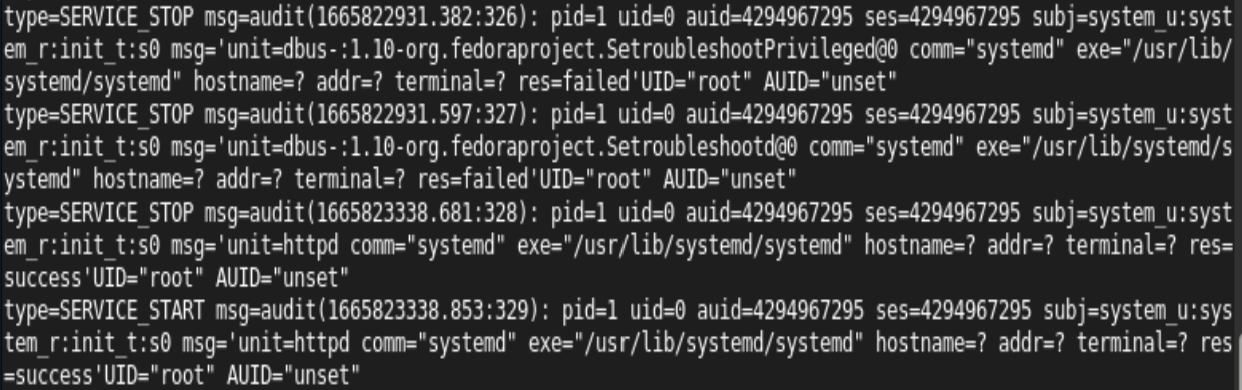
Лог-файл /var/log/messages



Лог-файл /var/log/http/error\_log

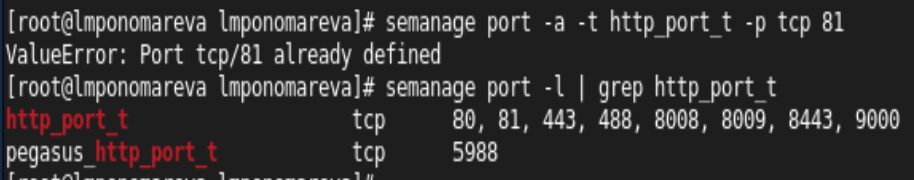


Лог-файл /var/log/http/access\_log



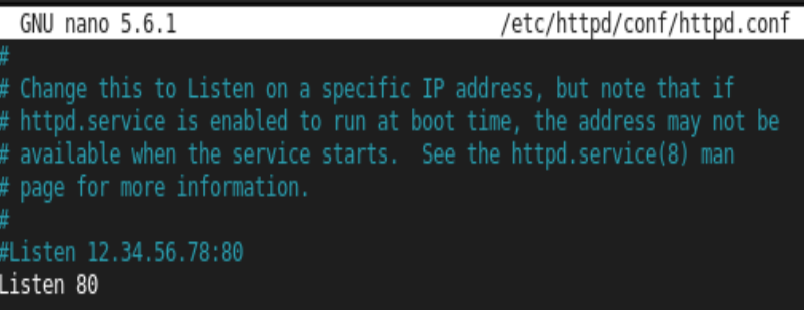
Лог-файл /var/log/audit/audit.log

1. Добавила порт 81 в список портов. ([рис. 22](../../image/23.png)).



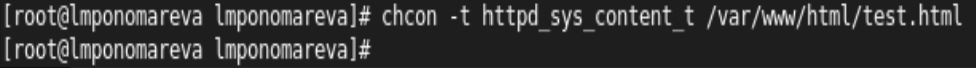
Список портов

1. Запустила веб-сервер Apache ещё раз. ([рис. 23](../../image/24.png)).



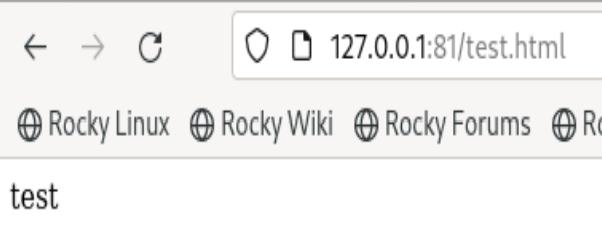
Перезапуск Apache

1. Вернула контекст httpd\_sys\_cоntent\_\_t к файлу /var/www/html/ test.html: ([рис. 24](../../image/26.png)).



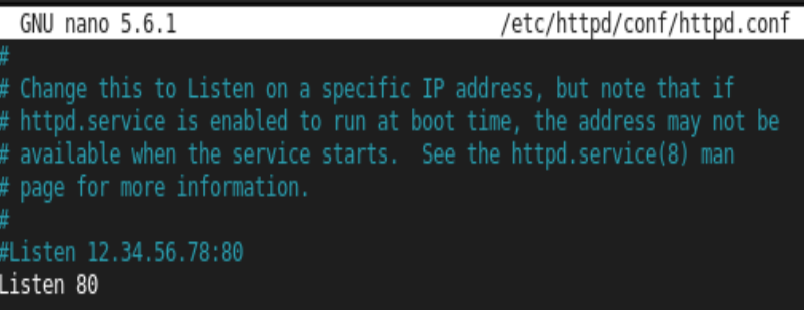
Контекст файла

Попробовала получить доступ к файлу через веб-сервер по 81 порту ([рис. 25](../../image/25.png)).



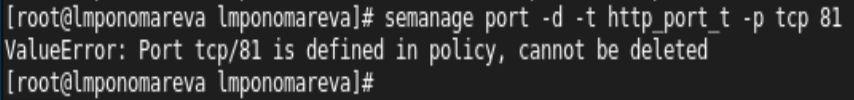
Доступ по другому порту

1. Исправила обратно конфигурационный файл apache ([рис. 26](../../image/27.png)).



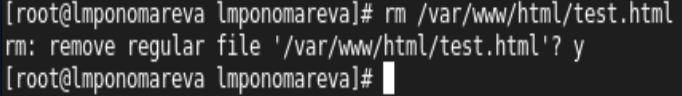
Конфигурационный файл Apache

1. Попробовала удалить привязку http\_port\_t к 81 порту ([рис. 27](../../image/28.png)).



Удаление порта из списка

1. Удалила файл /var/www/html/test.html ([рис. 28](../../image/29.png)).



Удаление файла

# Выводы

Получили практическое знакомство с технологией SELinux1. Проверили работу SELinx на практике совместно с веб-сервером Apache

# Список литературы

1. [Основы безопасности информационных систем : Учеб. пособие для студентов вузов, обучающихся по специальностям “Компьютер. безопасность” и “Комплекс. обеспечение информ. безопасности автоматизир. систем” / Д.А. Зегжда, А.М. Ивашко. - М. : Горячая линия - Телеком, 2000. - 449, [2] с. : ил., табл.; 21 см.; ISBN 5-93517-018-3](https://search.rsl.ru/ru/record/01000682756).