Отчёт по лабораторной работе 6

Основы информационной безопасности

Нджову Нелиа

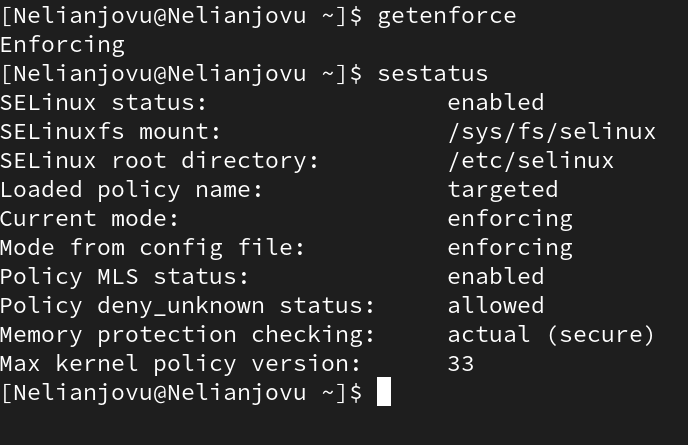
Содержание

# 1 Цель работы

Развить навыки администрирования ОС Linux. Получить первое практическое знакомство с технологией SELinux1. Проверить работу SELinux на практике совместно с веб-сервером Apache.

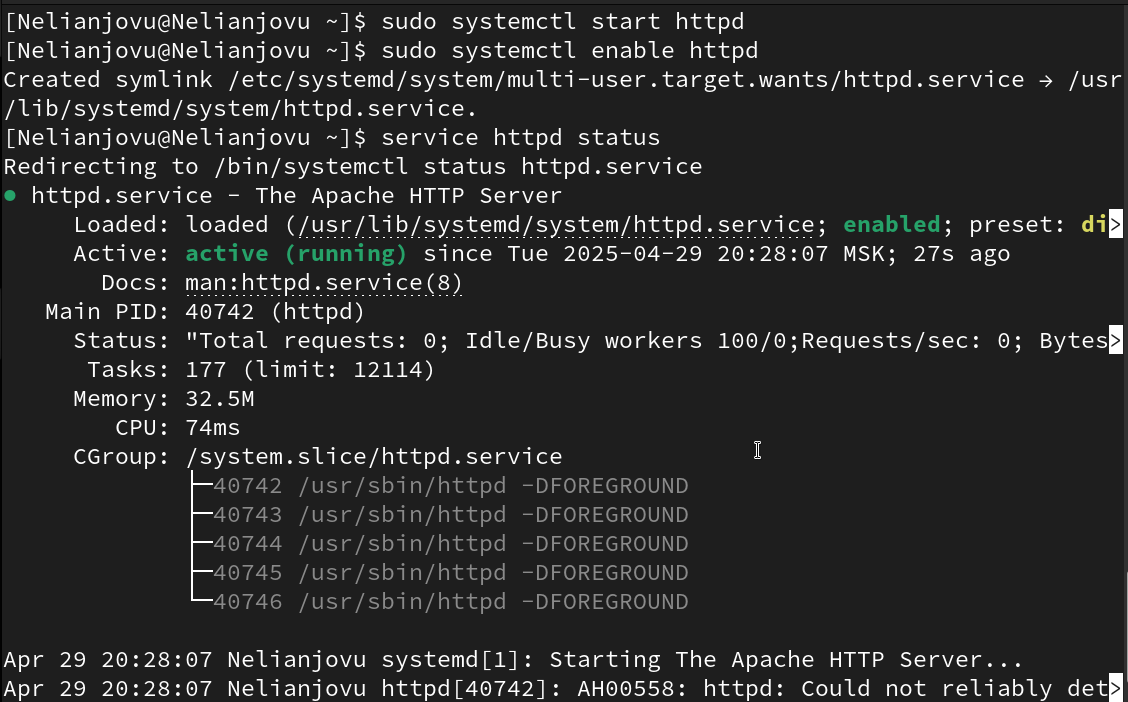
# 2 Выполнение лабораторной работы

Я вошла в свою учетную запись и убедилась, что SELinux работает в режиме принудительного применения целевой политики, используя команды getenforce и status(рис.1)



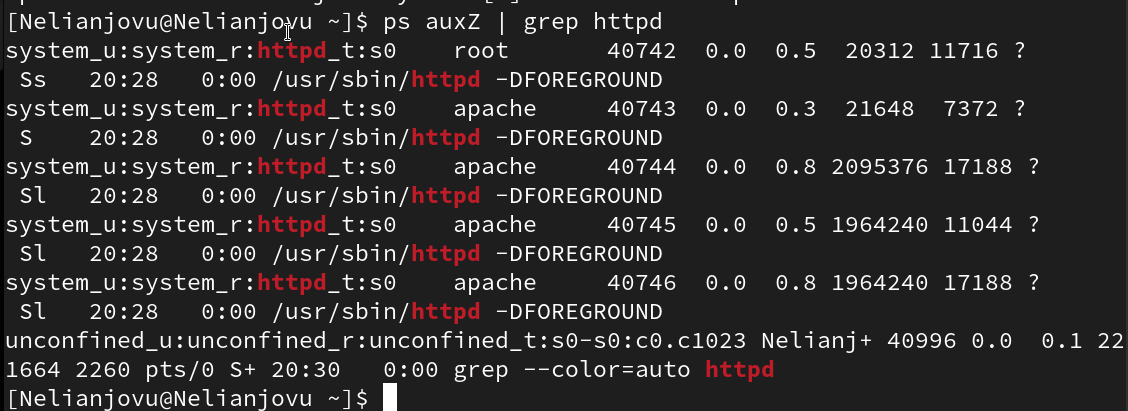
проверка режима работы SELinux

Я запускаю сервер apache, затем использую браузер для доступа к веб-серверу, запущенному на компьютере, он запущен, как видно из вывода команды service httpd status(рис.2)



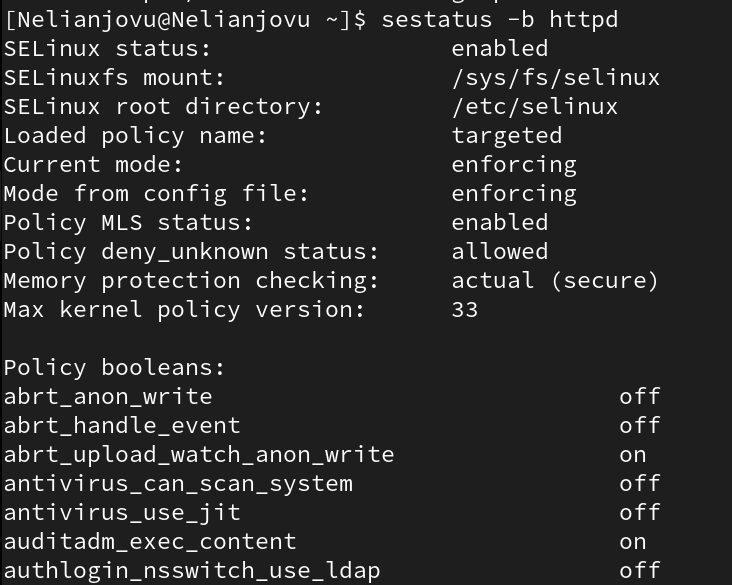
Проверка работы Apache

Я нашла веб-сервер Apache в списке процессов, используя команду ps aux | grep httpd. Его контекст безопасности - http\_t(рис.3)



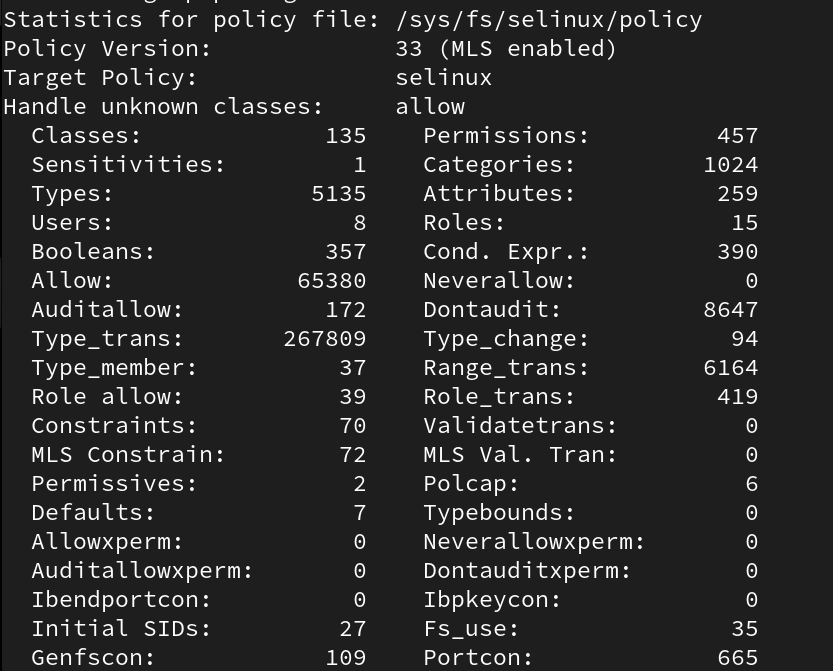
Контекст безопасности Apache

Я просмотрела текущее состояние коммутаторов SELinux для Apache, используя команду status -grep httpd(рис.4)



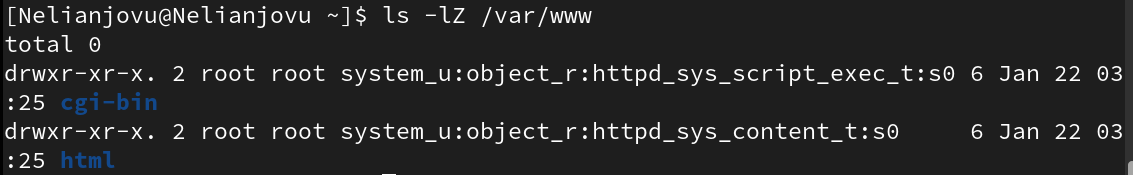
Состояние переключателей SELinux

Я просмотрела статистику по политике, используя команду setinfo. Всего 8 пользователей, 39 ролей и 5135 типов(рис.5)



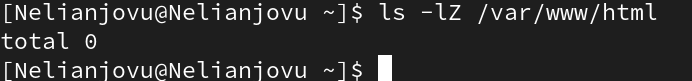
Cтатистика по политике

Типы подкаталогов, расположенных в каталоге /var/www с помощью команды ls - lZ /var/www, следующие: владельцем является root, только у владельца есть права на изменение. В каталоге нет файлов(рис.6)



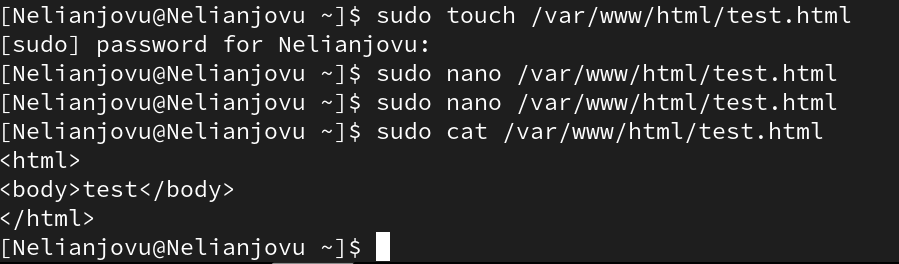
Типы поддиректорий

В директории /var/www/html нет файлов(рис.7)

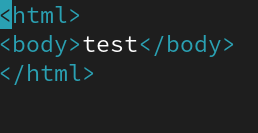


Типы файлов

Только суперпользователь может создать файл, поэтому я создала файл с помощью команды touch.html и ввела в него код(рис.8 и рис.9)

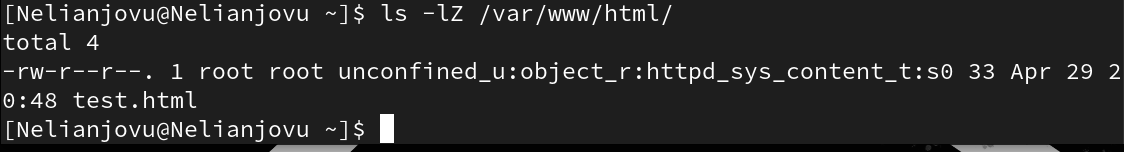


Создание файла



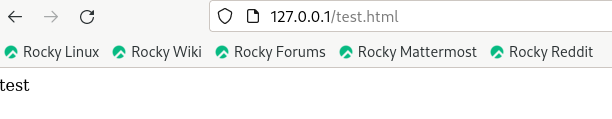
Содержание файла

Я проверяю контекст созданного файла. По умолчанию это httpd\_sys\_content\_type(рис.10)



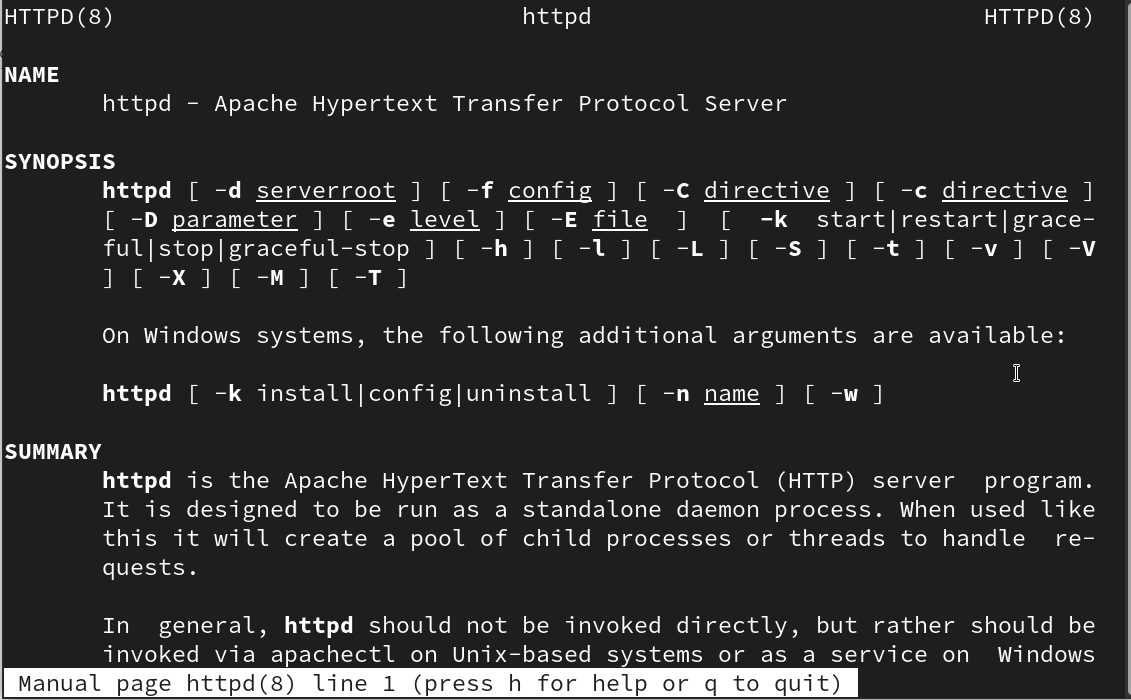
Контекст файла

Я получаю доступ к файлу через веб-сервер, вводя адрес в браузере http://127.0.0.1/test.html . Файл был успешно отображен(рис.11)



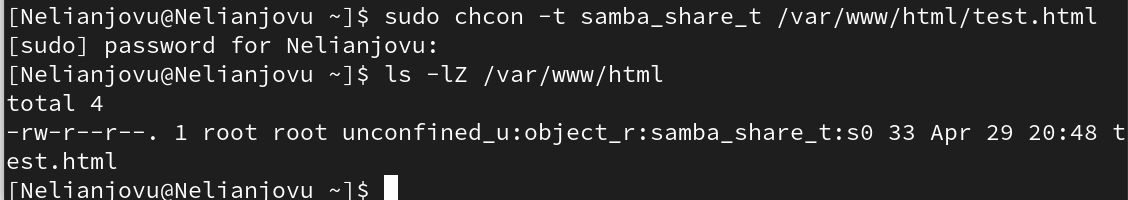
Отображение файла

Я изучил справку man по httpd-selinux. Давайте рассмотрим полученный контекст подробнее. Поскольку по умолчанию пользователи CentOS не имеют доступа к типам, созданный нами файл test.html SELinux был сопоставлен, пользователь unconfined-u. Это первая часть контекста. Кроме того, политика совместного использования ролей RBAC используется процессами, но не файлами, поэтому роли не имеют значения для файлов. Роль object-r используется по умолчанию для файлов на “постоянных” носителях и в 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.12)



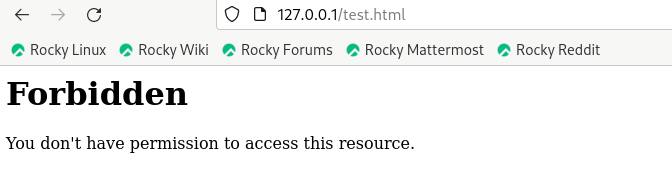
Изучение справки по команде

I change the context of the file /var/www/html/test.html from httpd\_sys\_content\_t to any other that the httpd process should not have access to, for example, to samba\_share\_t: chcon -t samba\_share\_t /var/www/html/test.html css/var/www/html/test.html The context has really changed(рис.13)



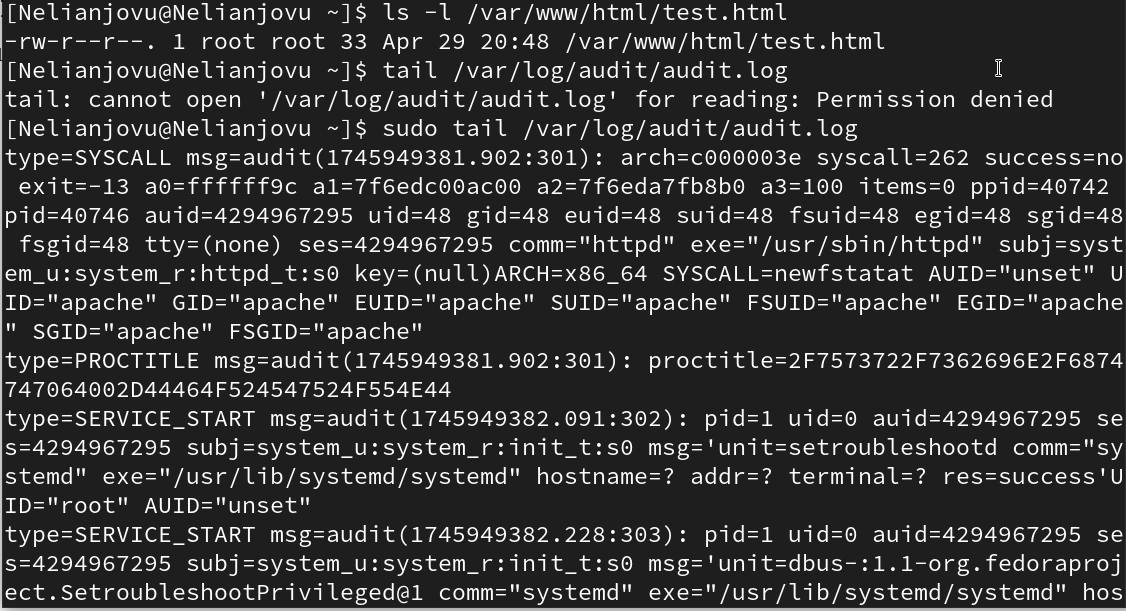
Изменение контекста

Когда я пытаюсь отобразить файл в браузере, мы получаем сообщение об ошибке(рис.14)



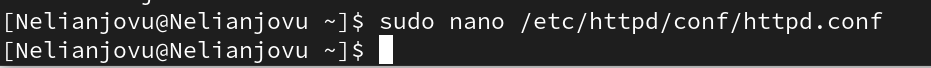
Отображение файла

Файл не был отображен, хотя права доступа позволяют любому пользователю читать этот файл, поскольку был задан контекст, к которому процесс httpd не должен иметь доступа. Я просматриваю файлы журнала веб-сервера Apache и файл системного журнала: tail /var/log/messages. Если в системе запущены процессы setroubleshootd и audit, вы также можете увидеть ошибки, аналогичные перечисленным выше, в файле /var/log/audit/audit.log(рис.15)



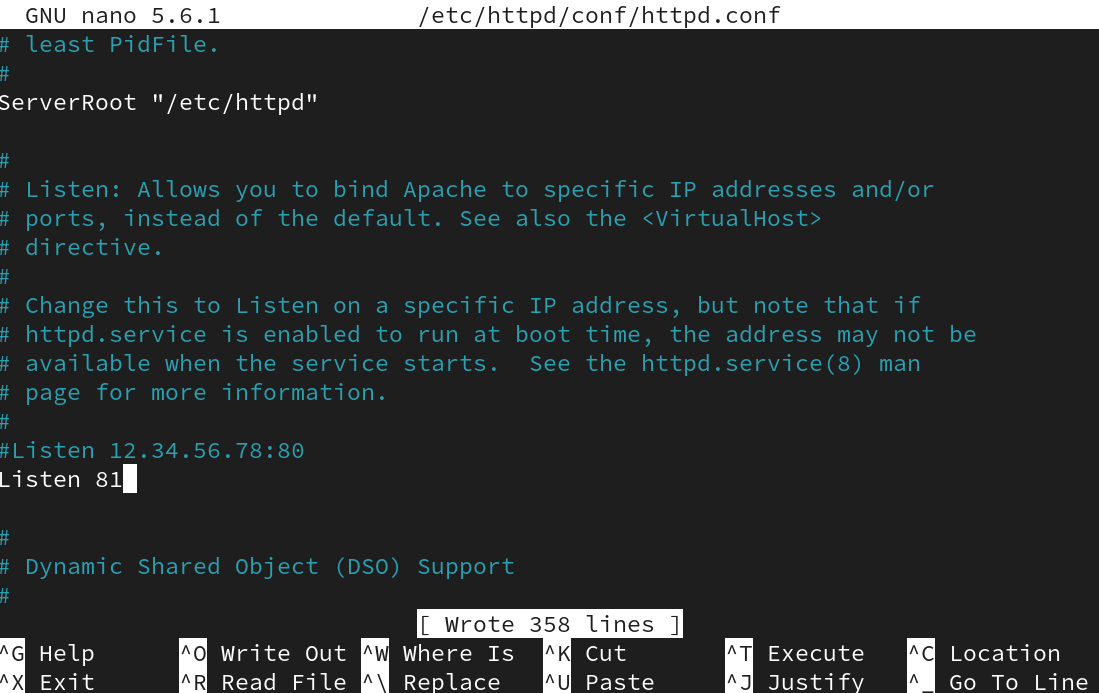
Попытка прочесть лог-файл

Чтобы запустить веб-сервер Apache, прослушивающий TCP-порт 81, я открываю файл /etc/httpd/httpd.conf для внесения изменений(рис.16)



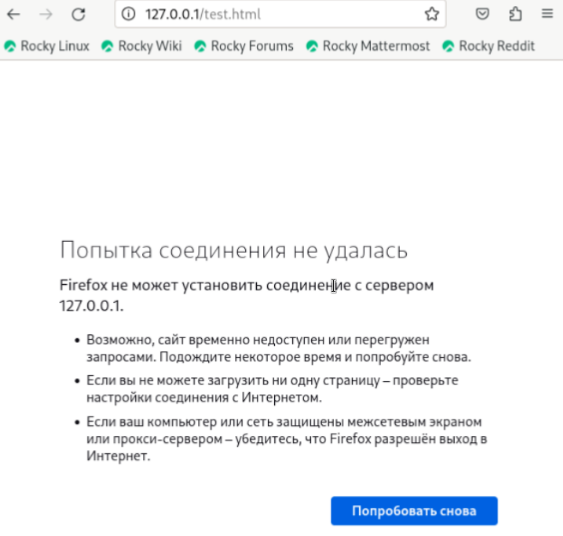
Изменение файла

Нахожу строчку Listen 80 и заменяю её на Listen 81(рис.17)



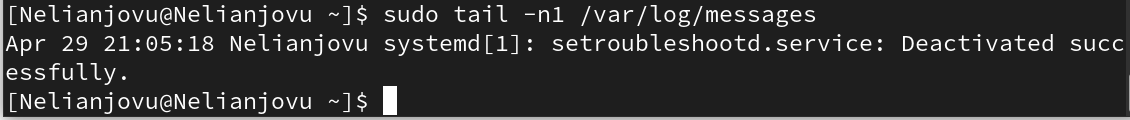
Изменение порта

Я перезапускаю веб-сервер Apache. Сбой произошел из-за того, что порт 80 предназначен для локальной сети, а порт 81 - нет(рис.18)



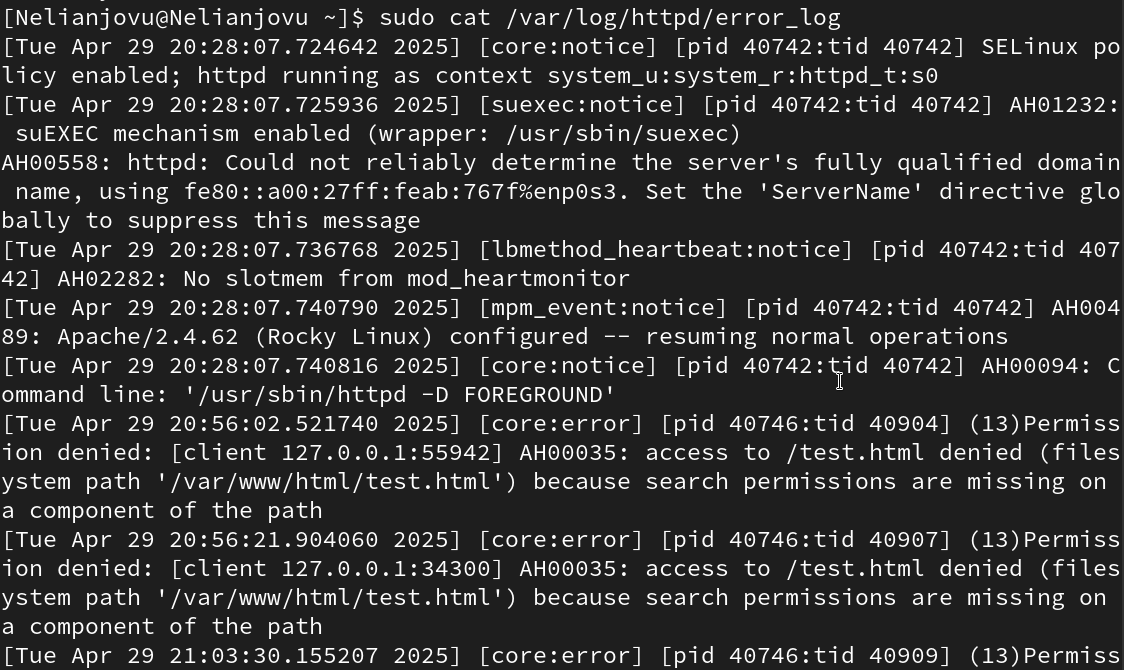
Попытка прослушивания другого порта

Проанализируйте лог-файлы: tail -nl /var/log/messages(рис.19)



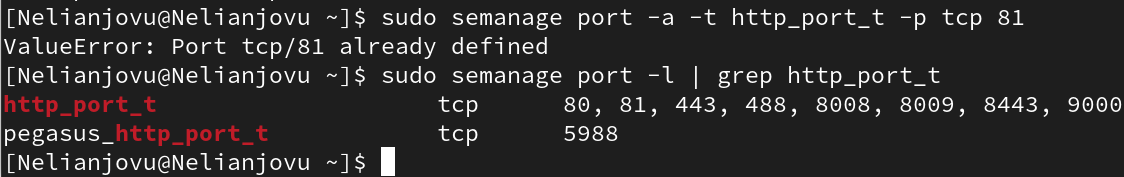
Проверка лог-файлов

Я просматриваю файлы /var/log/http/error\_log, /var/log/httpd access\_log и /var/log/audit/аудит.лог и выясняю, в каких файлах появились записи. Запись появилась в файле error\_log(рис.20)



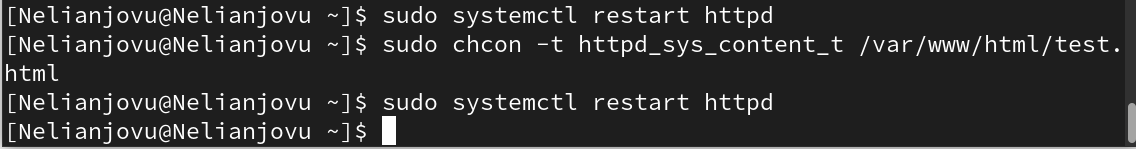
Проверка лог-файлов

Я запускаю команду semanage port -at http\_port\_to tcp 81, после чего проверяю список портов командой semanage port -l | grep http\_port\_t Порт 81 появился в списке(рис.21)



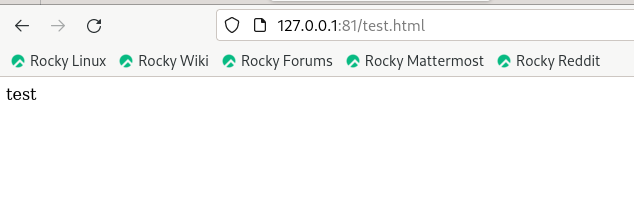
Проверка портов

Перезапускаю сервер Apache(рис.22)



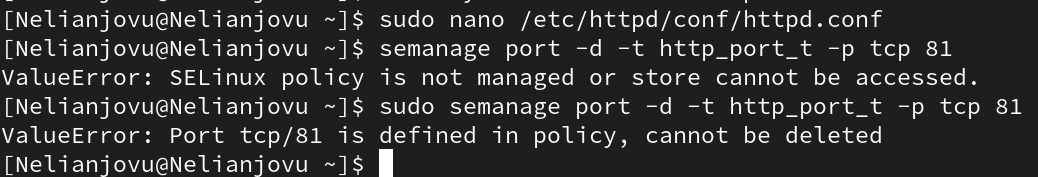
Перезапуск сервера

Теперь он работает, ведь мы внесли порт 81 в список портов httpd\_port\_t(рис.23)



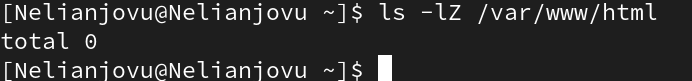
Проверка сервера

Возвращаю в файле /etc/httpd/httpd.conf порт 80, вместо 81. Проверяю, что порт 81 удален, это правда(рис.24)



Проверка порта 81

Далее удаляю файл test.html, проверяю, что он удален(рис.25)



Удаление файла

# 3 Выводы

В ходе выполнения данной лабораторной работы были развиты навыки администрирования ОС Linux, получено первое практическое знакомство с технологией SELinux и проверена работа SELinux на практике совместно с веб-сервером Apache

# Список литературы

006-lab\_selinux.pdf