Отчёт по лабораторной работе №6  
Информационная безопасность

Мандатное разграничение прав в Linux

Выполнил: Махорин Иван Сергеевич,  
НПИбд-02-21, 1032211221

Содержание

# 1 Цель работы

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

# 2 Теоретическое введение

1. **SELinux (Security-Enhanced Linux)** обеспечивает усиление защиты путем внесения изменений как на уровне ядра, так и на уровне пространства пользователя, что превращает ее в действительно «непробиваемую» операционную систему. Впервые эта система появилась в четвертой версии CentOS, а в 5 и 6 версии реализация была существенно дополнена и улучшена.

*SELinux имеет три основных режим работы:*

* 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.

Политика SELinux определяет доступ пользователей к ролям, доступ ролей к доменам и доступ доменов к типам. Контекст безопасности — все атрибуты SELinux — роли, типы и домены. Более подробно см. в [1].

1. **Apache** — это свободное программное обеспечение, с помощью которого можно создать веб-сервер. Данный продукт возник как доработанная версия другого HTTP-клиента от национального центра суперкомпьютерных приложений (NCSA).

*Для чего нужен Apache сервер:*

* чтобы открывать динамические PHP-страницы,
* для распределения поступающей на сервер нагрузки,
* для обеспечения отказоустойчивости сервера,
* чтобы потренироваться в настройке сервера и запуске PHP-скриптов.

Apache является кроссплатформенным ПО и поддерживает такие операционные системы, как Linux, BSD, MacOS, Microsoft, BeOS и другие.

Более подробно см. в [2].

# 3 Выполнение лабораторной работы

Войдём в систему и убедимся, что SELinux работает в режиме enforcing политики targeted с помощью команд getenforce и sestatus

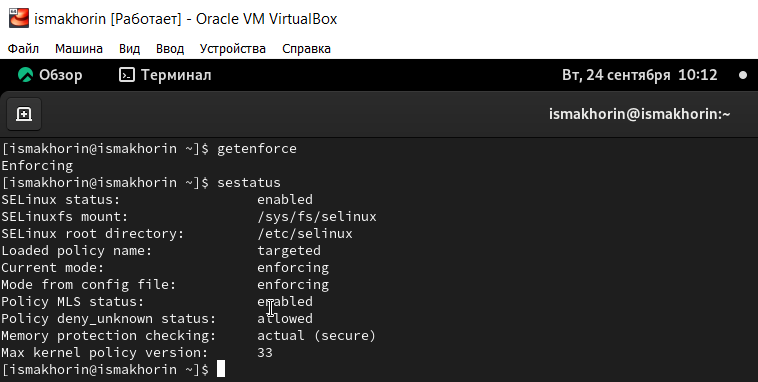


Рис. 1: Проверка работы в режиме enforcing политики targeted

Обратимся с помощью браузера к веб-серверу, запущенному на нашем компьютере, и убедимся, что последний работает

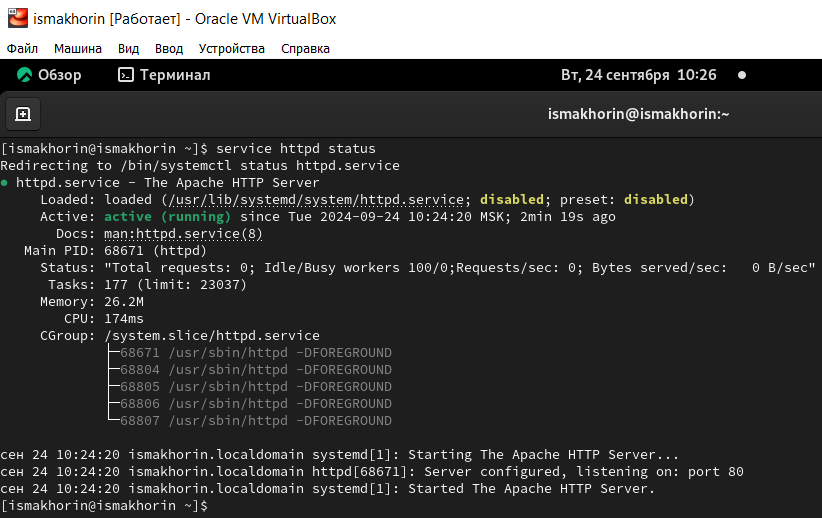


Рис. 2: Проверка работы веб-сервера

Затем найдём веб-сервер Apache в списке процессов и определим его контекст безопасности

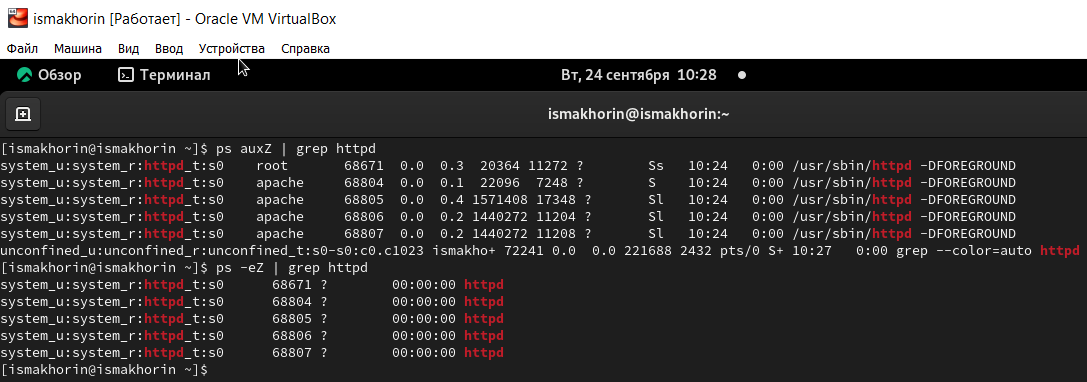


Рис. 3: Нахождение веб-сервера Apache в списке процессов и определение его контекста безопасности

Посмотрим текущее состояние переключателей SELinux для Apache

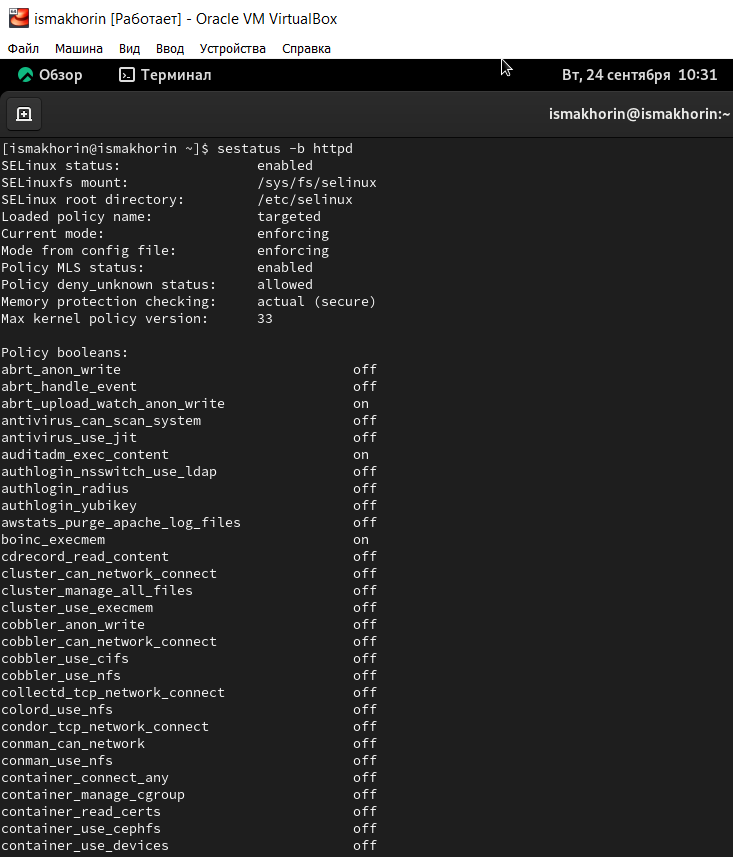


Рис. 4: Просмотр текущего состояния переключателей SELinux

Далее посмотрим статистику по политике

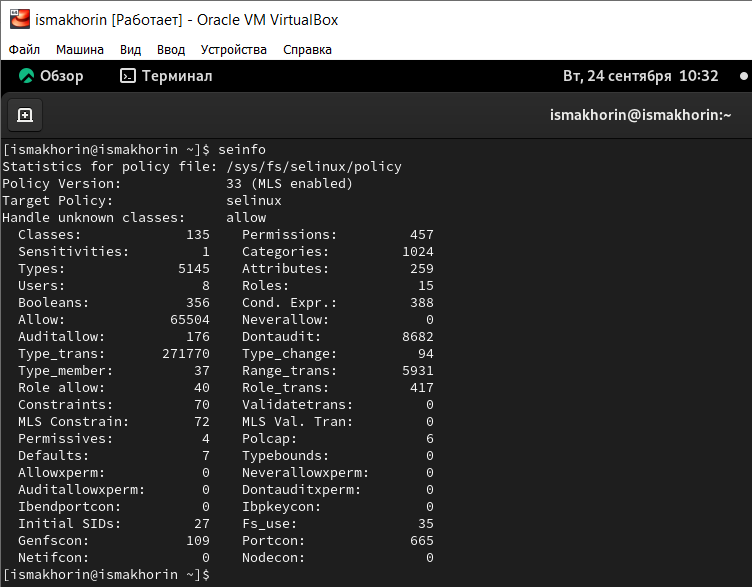


Рис. 5: Просмотр статистики по политике

Определим тип файлов и поддиректорий, находящихся в директории /var/www

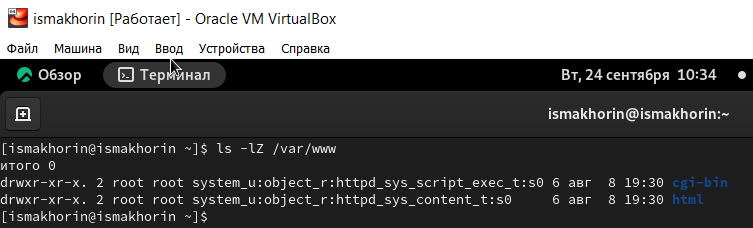


Рис. 6: Определение типов файлов и поддиректорий

Определим тип файлов, находящихся в директории /var/www/html

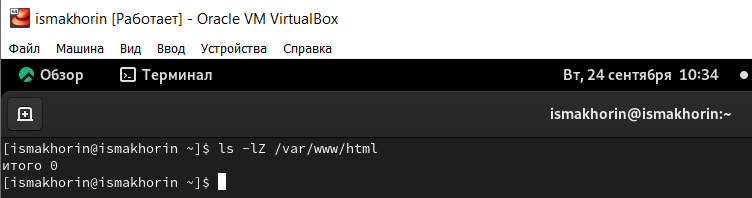


Рис. 7: Определение типов файлов и поддиректорий

Следующим шагом создадим от имени суперпользователя html-файл /var/www/html/test.html с содержанием “test”

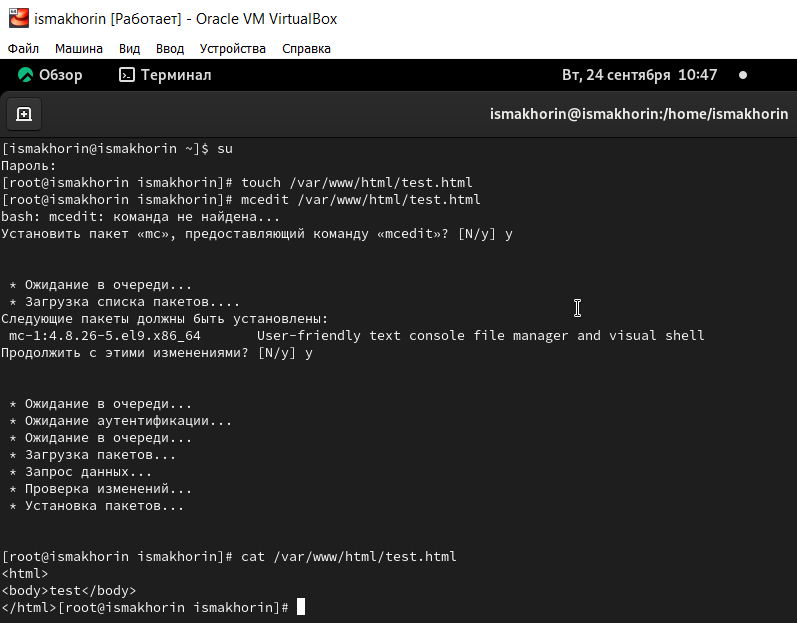


Рис. 8: Создание файла с содержанием

Проверим контекст созданного нами файла

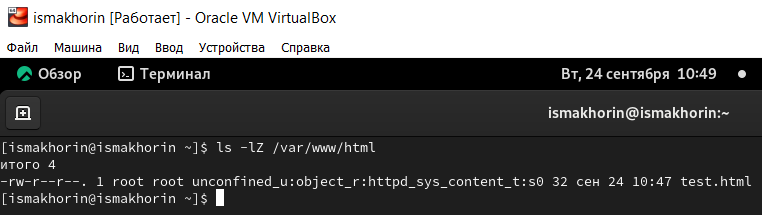


Рис. 9: Проверка контекста созданного файла

Изучим справку man httpd\_selinux и выясним, какие контексты файлов определены для httpd. Сопоставим их с типом файла test.html и проверим контекст файла

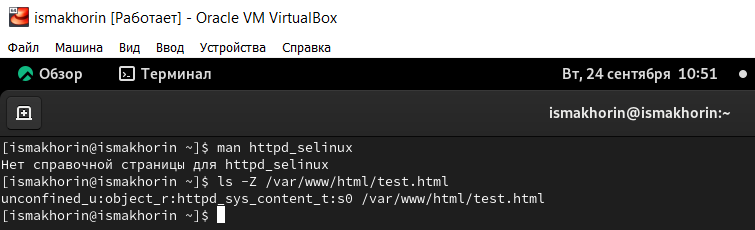


Рис. 10: Изучение справки и проверка контекста файла

Изменим контекст файла /var/www/html/test.html с httpd\_sys\_content\_t на samba\_share\_t

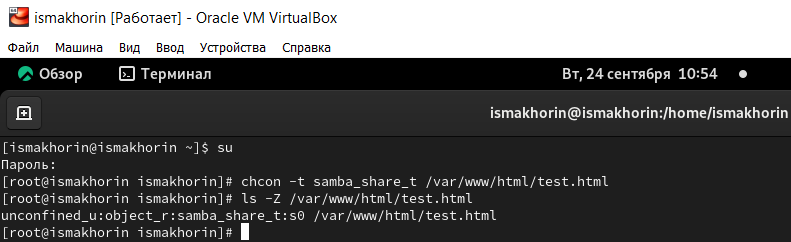


Рис. 11: Изменение контекста файла и проверка

Попробуем получить доступ к файлу через веб-сервер, введя в браузере адрес http://127.0.0.1/test.html

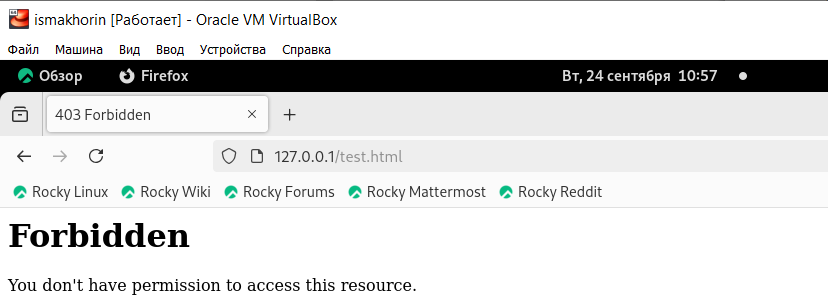


Рис. 12: Попытка получения доступа к файлу через веб-сервер

Просмотрим log-файлы веб-сервера Apache и системный лог-файл

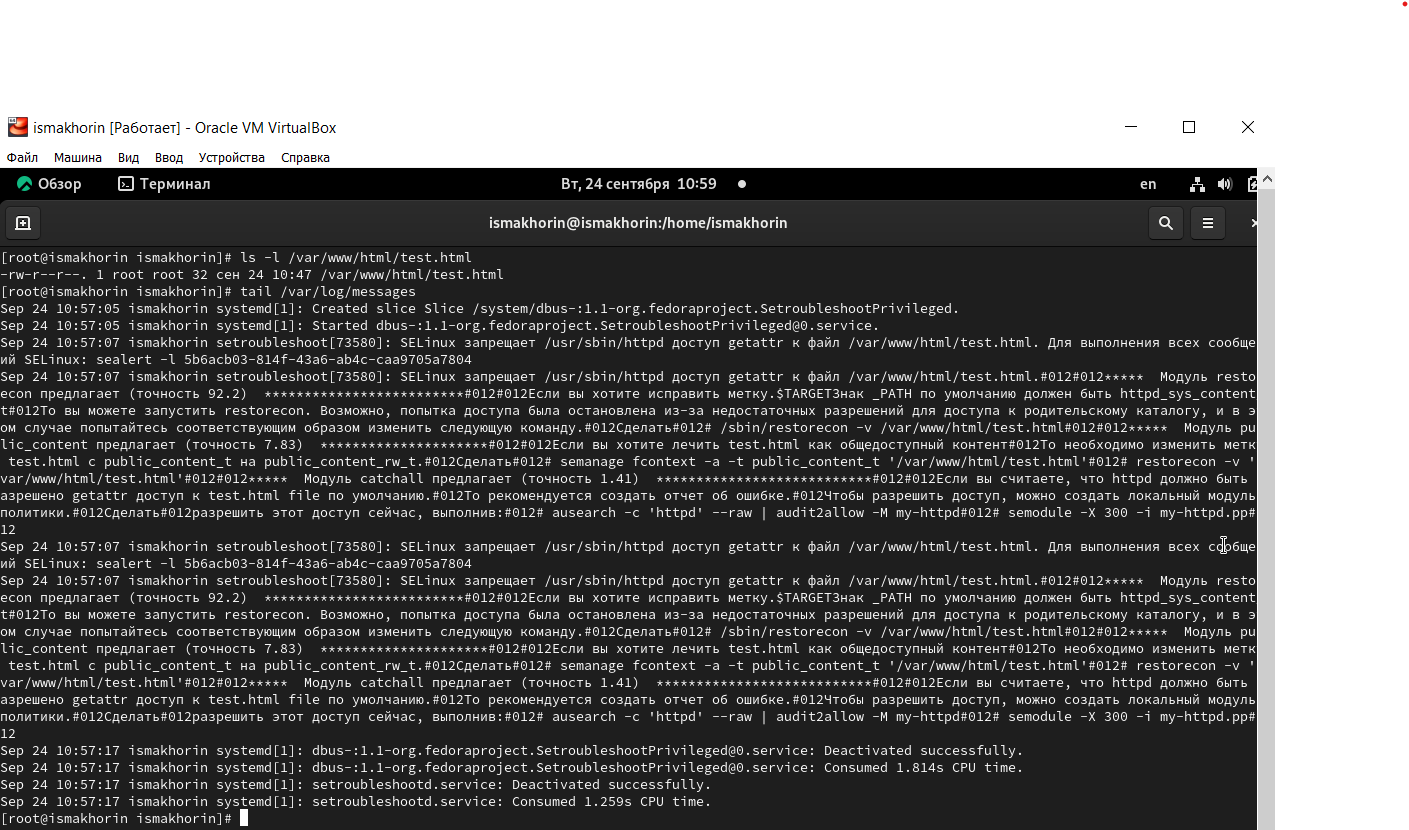


Рис. 13: Просмотр log-файлов веб-сервера Apache и системного лог-файла

Попробуем запустить веб-сервер Apache на прослушивание ТСР-порта 81 (а не 80, как рекомендует IANA и прописано в /etc/services)

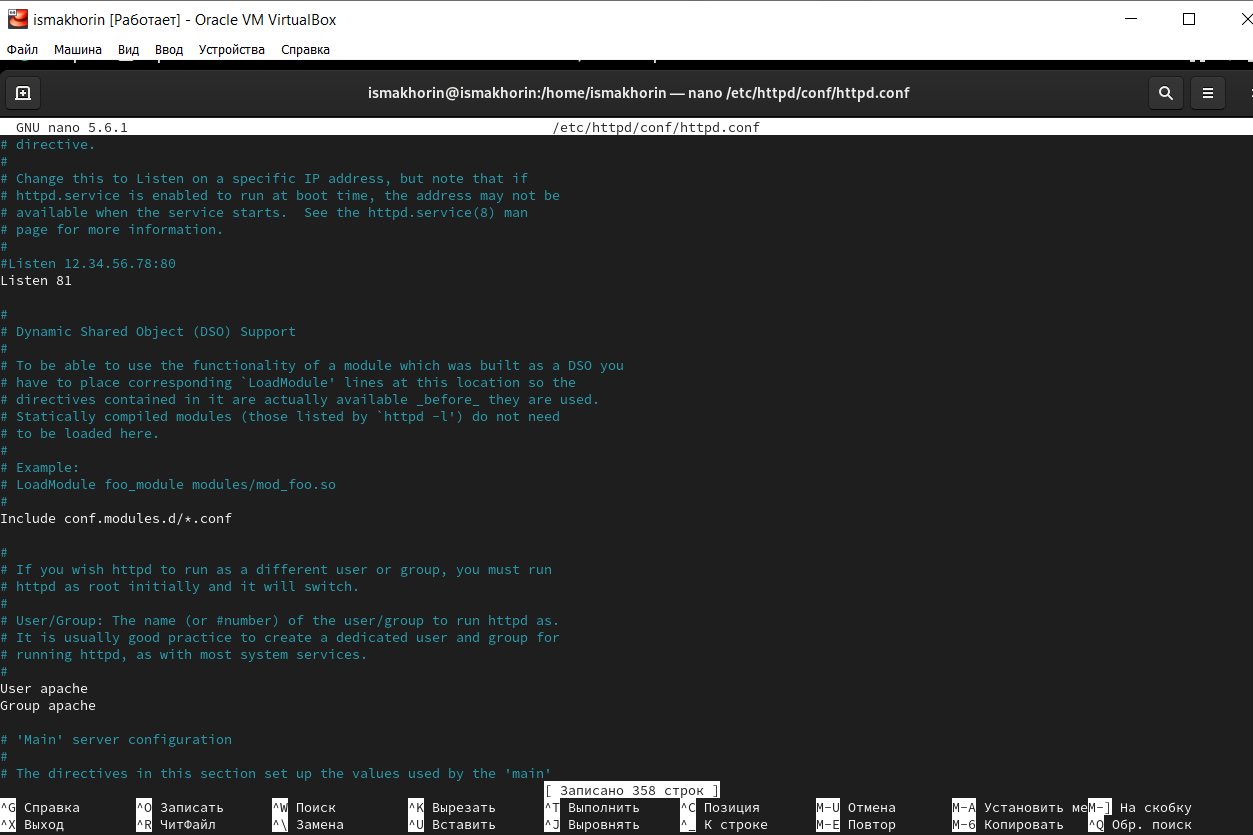


Рис. 14: Попытка запуска веб-сервера Apache на прослушивание ТСР-порта 81

Выполним перезапуск веб-сервера Apache

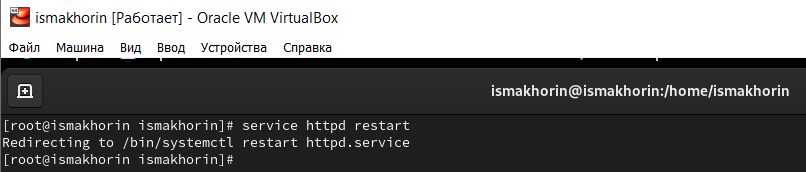


Рис. 15: Перезапуск веб-сервера Apache

Проанализируем лог-файлы: tail -nl /var/log/messages, /var/log/http/error\_log, /var/log/http/access\_log и /var/log/audit/audit.log

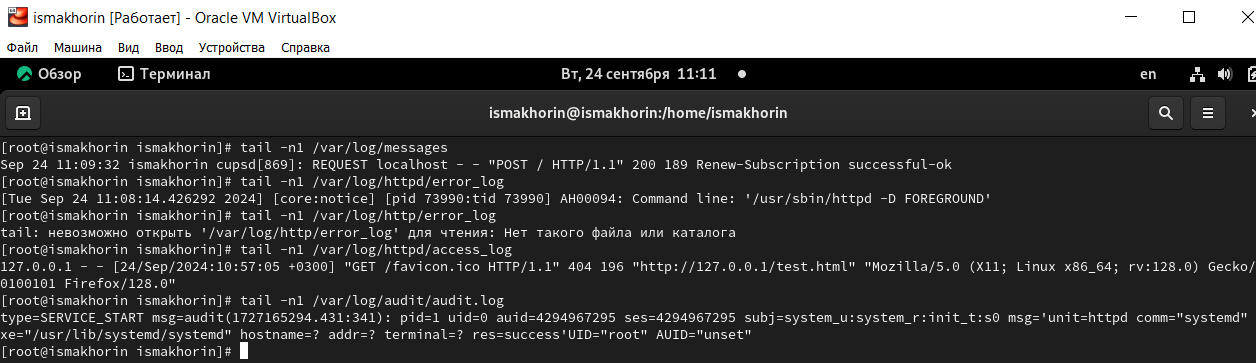


Рис. 16: Анализ лог-файлов

Выполним команду semanage port -a -t http\_port\_t -р tcp 81 и после этого проверим список портов

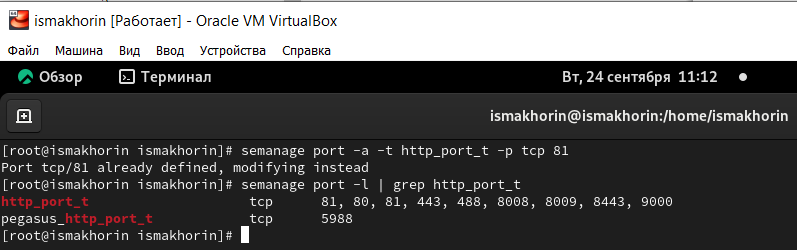


Рис. 17: Выполнение команды и проверка списка портов

Вернём контекст httpd\_sys\_cоntent\_\_t к файлу /var/www/html/ test.html. После этого попробуем получить доступ к файлу через веб-сервер, введя в браузере адрес http://127.0.0.1:81/test.html

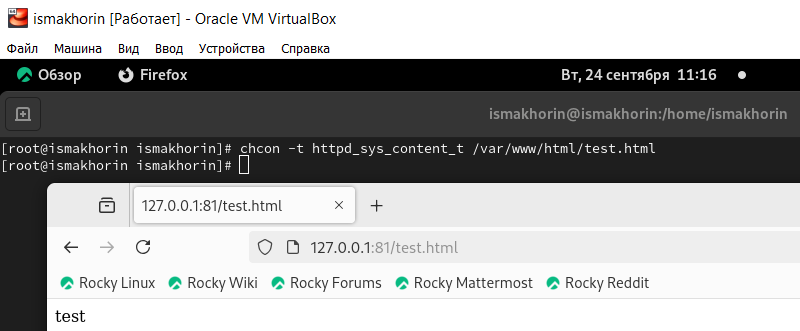


Рис. 18: Возвращение контекста httpd\_sys\_cоntent\_\_t к файлу /var/www/html/ test.html и попытка получения доступа к файлу через веб-сервис

Исправим обратно конфигурационный файл apache, вернув Listen 80

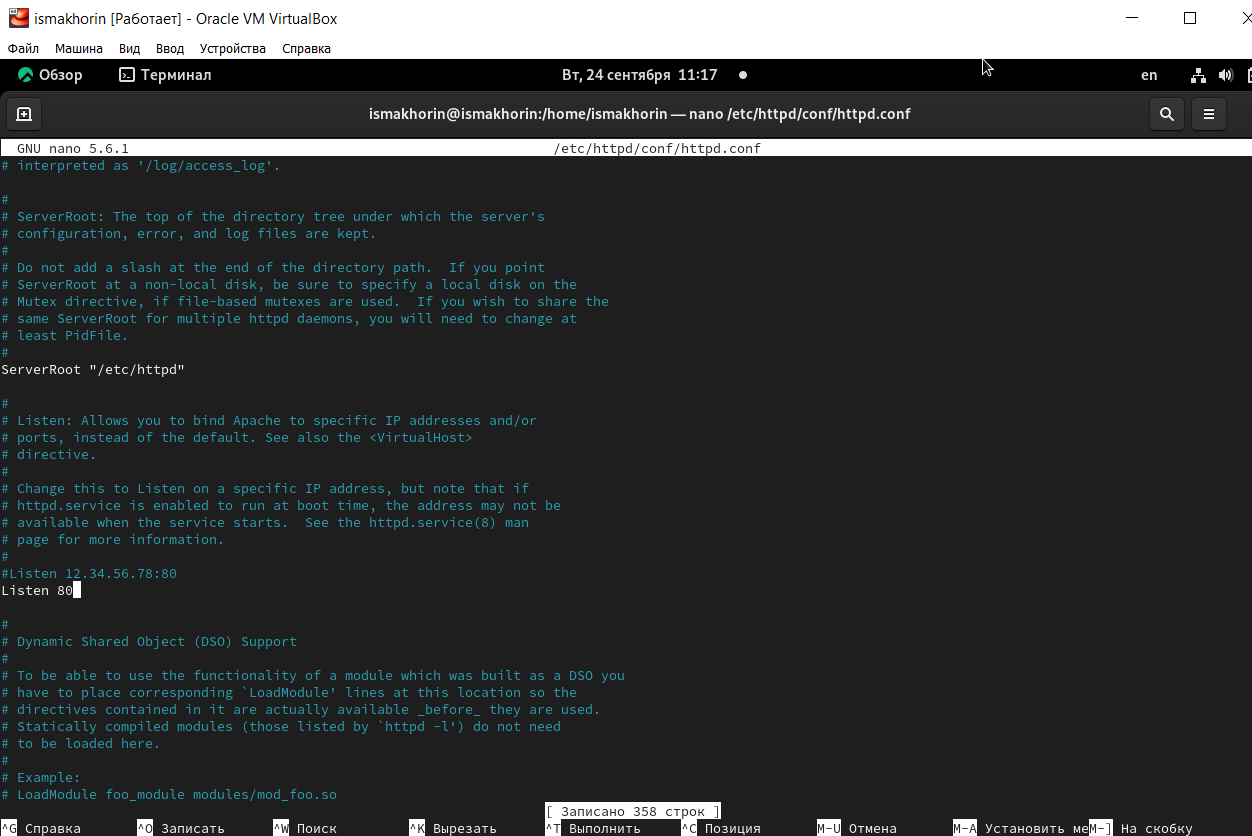


Рис. 19: Исправление конфигурационного файла apache

Удалим привязку http\_port\_t к 81 порту

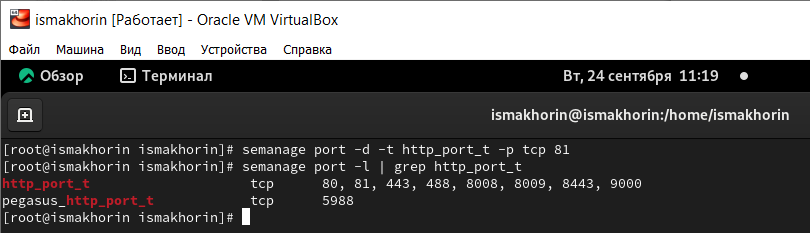


Рис. 20: Удаление привязки http\_port\_t к 81 порту и проверка

Удалим файл /var/www/html/test.html

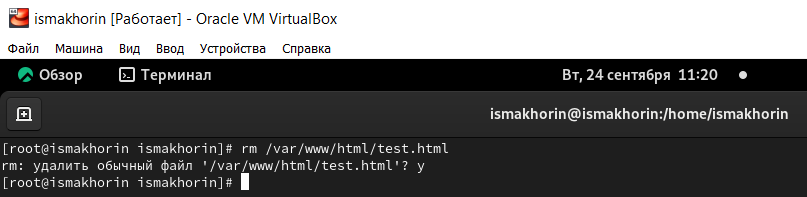


Рис. 21: Удаление файла /var/www/html/test.html

# 4 Вывод

В ходе выполнения 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.

# 5 Список литературы. Библиография

[1] SELinux: https://habr.com/ru/companies/kingservers/articles/209644/

[2] Apache: https://2domains.ru/support/vps-i-servery/shto-takoye-apache