Отчет по лабораторной работе №6

Дисциплина: Информационная безопасность

Выполнила Дяченко Злата Константиновна, НФИбд-03-18

Содержание

# Цель работы

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

# Задание

Выполнить все пункты указания к лабораторной работе, чтобы достигнуть цель.

# Выполнение лабораторной работы

## Шаг 1

Вошла в систему и убедилась, что SELinux работает в режиме enforcing политики targeted. Обратилась с помощью браузера к веб-серверу и запустила его. Нашла веб-сервер Apache в списке процессов, определила его контекст безопасности: system\_u:system\_r:httpd\_t:s0 . Использовала команду ps auxZ | grep httpd, результат представлен на Рисунке 1 (рис. 1)

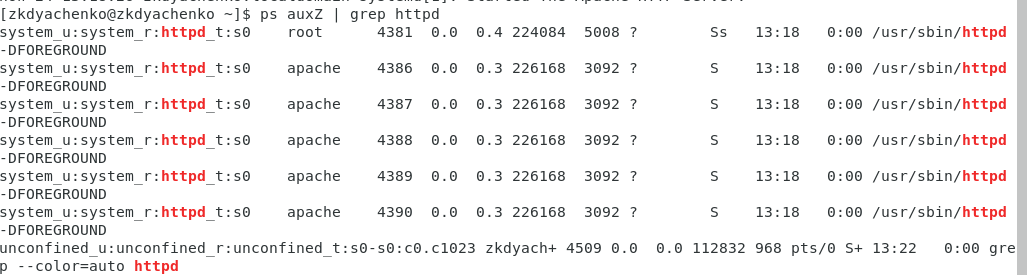


Figure 1: Список процессов

## Шаг 2

Посмотрела текущее состояние переключателей SELinux для Apache с помощью команды sestatus -bigrep httpd (рис. 2). Многие из них находятся в положении «off».

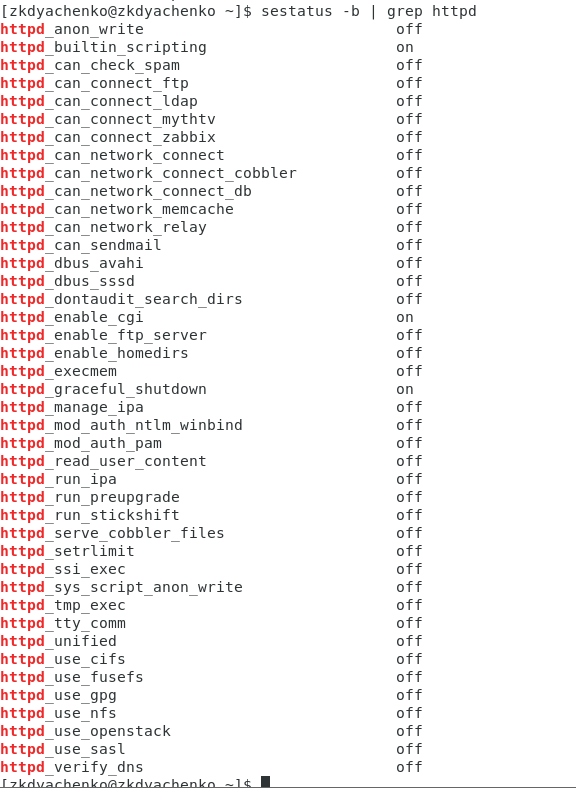


Figure 2: Состояния переключателей SELinux для Apache

## Шаг 3

Посмотрела статистику по политике с помощью команды seinfo (рис. 3).

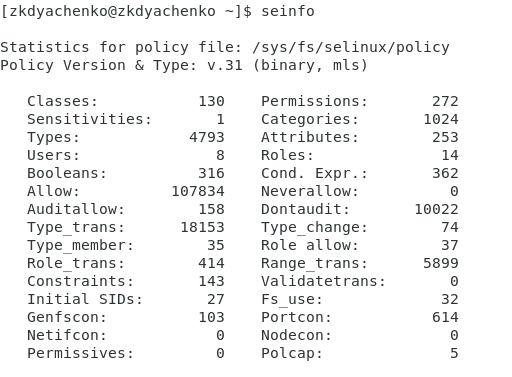


Figure 3: seinfo

## Шаг 4

Определите тип файлов и поддиректорий, находящихся в директории /var/www - там находятся только две поддиректории (рис. 4). Директория /var/www/html пуста. Создавать файлы в директории /var/www/html может только владелец директории - root.

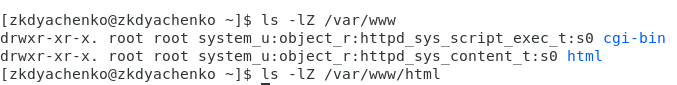


Figure 4: Информация о поддиректориях /var/www

## Шаг 5

Создала от имени суперпользователя html-файл /var/www/html/test.html следующего содержания (рис. 5).

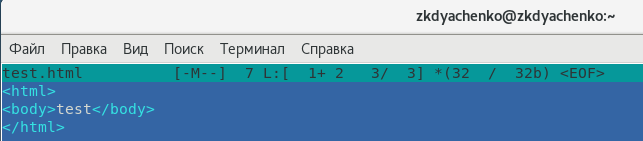


Figure 5: Содержание файла test.html

## Шаг 6

Проверила контекст созданного файла. По умолчанию это unconfined\_u:object\_r:httpd\_sys\_content\_t:s0 (рис. 6).

Figure 6: Контекст файла

Figure 6: Контекст файла

## Шаг 7

Обратилась к файлу через веб-сервер, введя в браузере адрес http://127.0.0.1/test.html. Файл был успешно отображён (рис. 7).

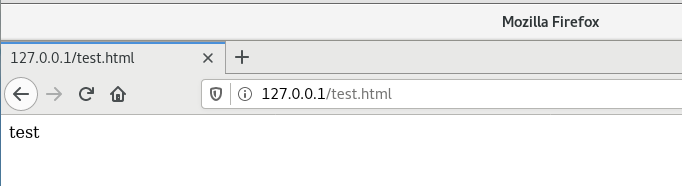


Figure 7: Отображение файла

## Шаг 8

Изучила справку man httpd\_selinux и выяснила, какие контексты файлов определены для httpd (рис. 8). Так как по умолчанию пользователи CentOS являются свободными от типа, созданному 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 возможно получить доступ к файлу при обращении к нему через браузер.

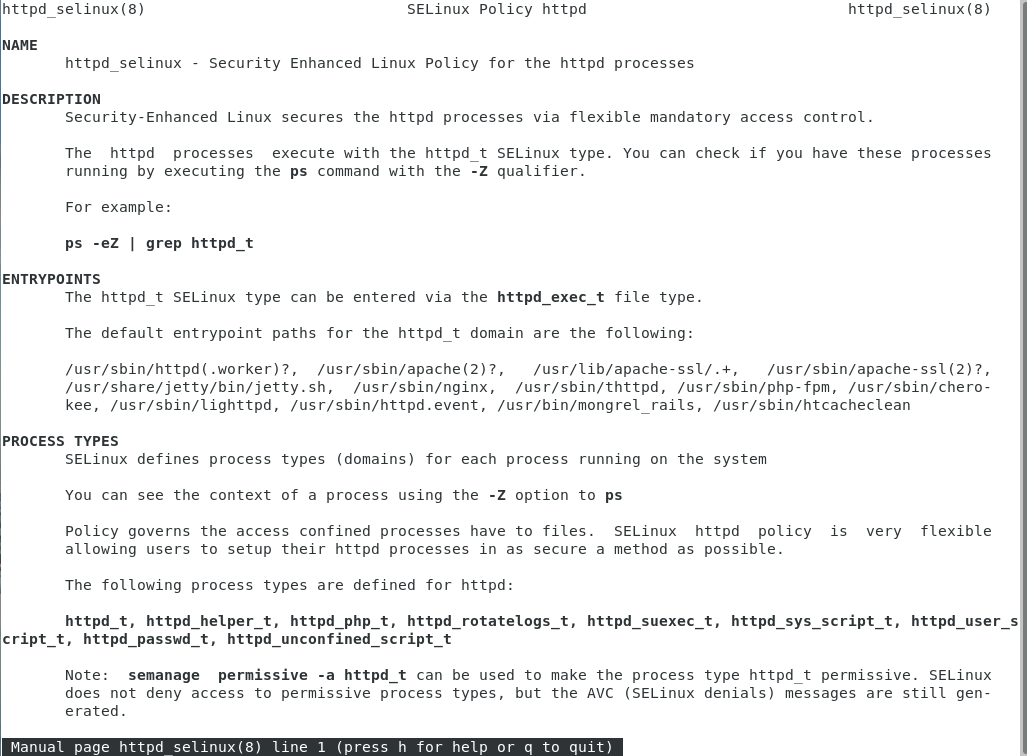


Figure 8: Справка man httpd\_selinux

## Шаг 9

Изменила контекст файла /var/www/html/test.html с httpd\_sys\_content\_t на на samba\_share\_t, к которому процесс httpd не имеет доступа. После этого проверила, что контекст поменялся (рис. 9).

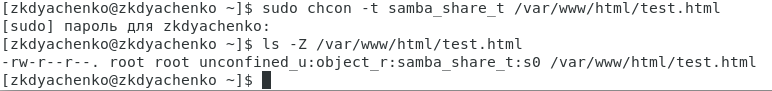


Figure 9: Изменение контекста файла

## Шаг 10

Попробовала ещё раз получить доступ к файлу через веб-сервер, но получила сообщение об ошибке (рис. 10).

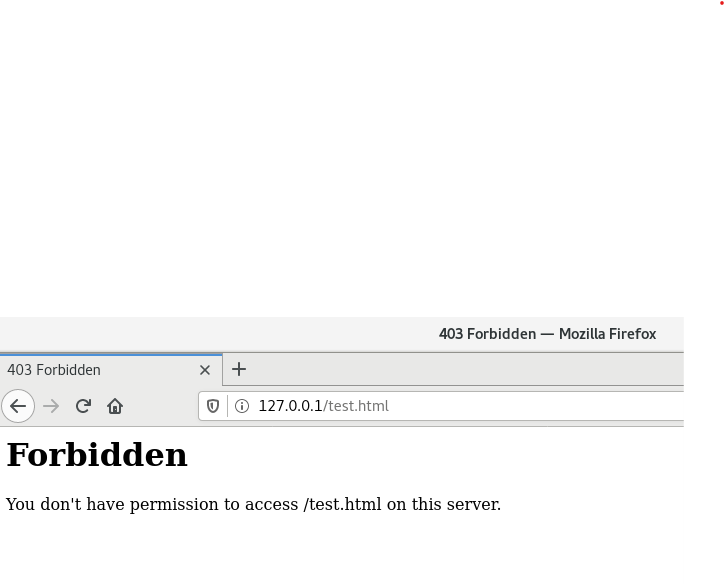


Figure 10: Ошибка

## Шаг 11

Просмотрела лог-файл tail /var/log/messages (рис. 11) и /var/log/audit/audit.log (рис. 12).

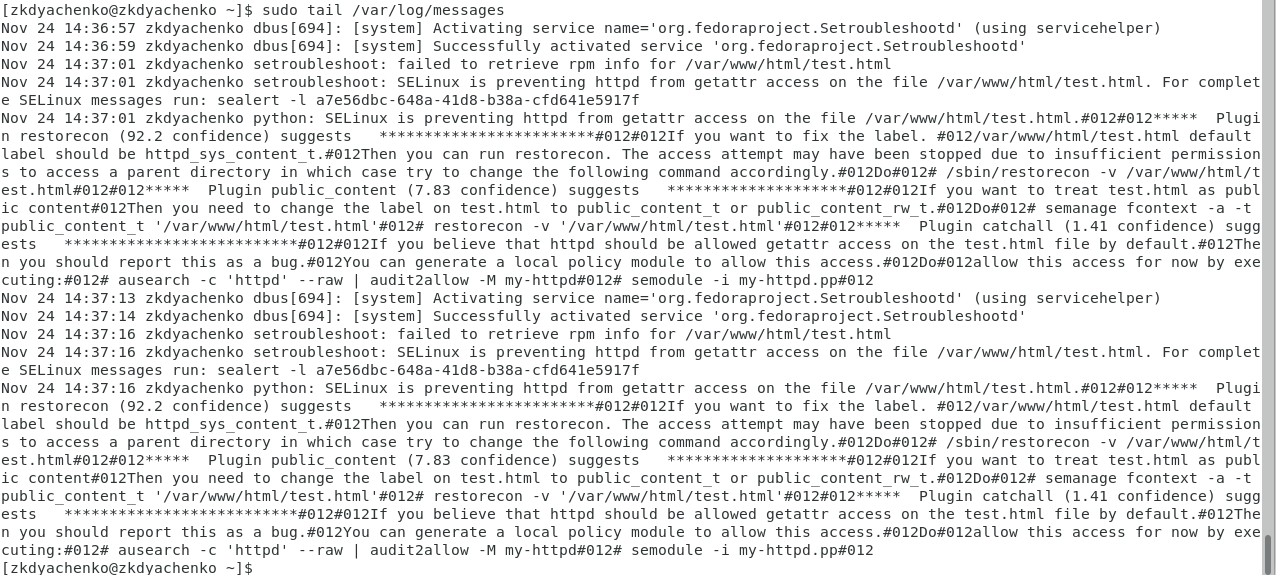


Figure 11: Файл /var/log/messages



Figure 12: Файл /var/log/audit/audit.log

## Шаг 12

Попробую запустить веб-сервер Apache на прослушивание ТСР-порта 81. Для этого в файле /etc/httpd/httpd.conf нашла строчку Listen 80 и заменила её на Listen 81 (рис. 13).

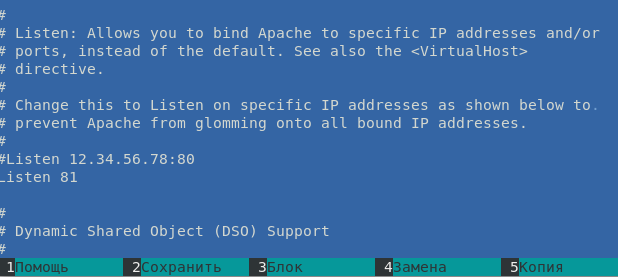


Figure 13: Замена строчки в файле

## Шаг 13

Выполнила перезапуск веб-сервера Apache. Сбоя не произошло (рис. 14). Следуя указаниям к лабораторной, посмотрела файлы tail -nl /var/log/messages, /var/log/http/error\_log, /var/log/http/access\_log и /var/log/audit/audit.log (рис. 15)

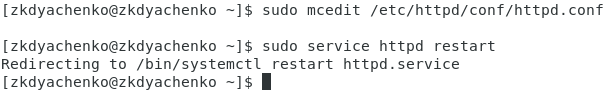


Figure 14: Перезапуск веб-сервера Apache



Figure 15: Просмотр файлов

## Шаг 14

Выполнила команду semanage port -a -t http\_port\_t -р tcp 81, после этого проверила список портов. Порт 81 есть в списке (рис. 16).

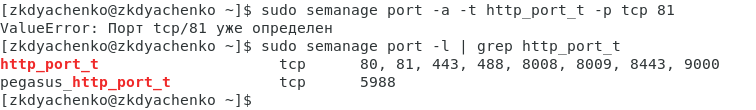


Figure 16: Список портов

## Шаг 15

Попробовала запустить веб-сервер Apache ещё раз. Он запустился (рис. 17).

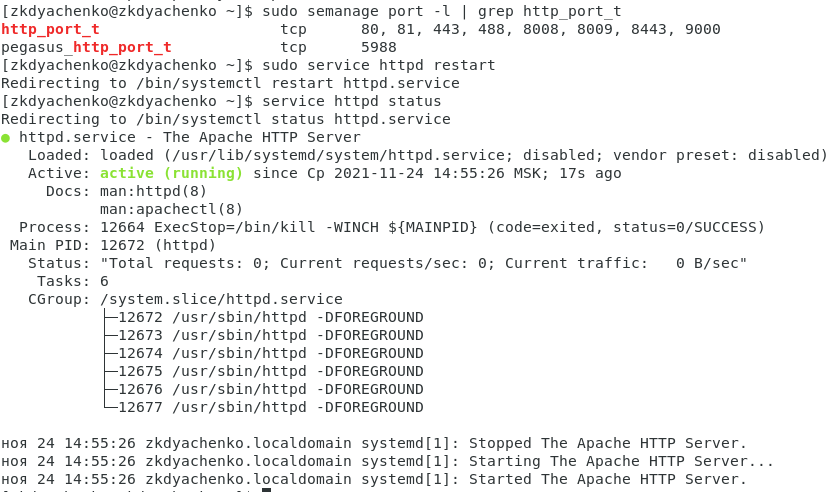


Figure 17: Перезапуск веб-сервера Apache

## Шаг 16

Вернула контекст httpd\_sys\_cоntent\_\_t к файлу /var/www/html/ test.html. После этого попробовала получить доступ к файлу через веб-сервер и увидела содержимое файла — слово «test». (рис. 18)

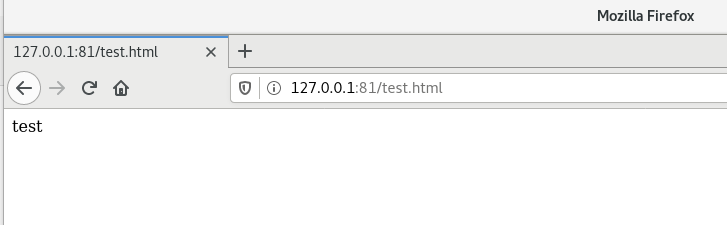


Figure 18: Получение доступа к файлу через веб-браузер

## Шаг 17

Исправила обратно конфигурационный файл apache, вернув Listen 80 (рис. 19).

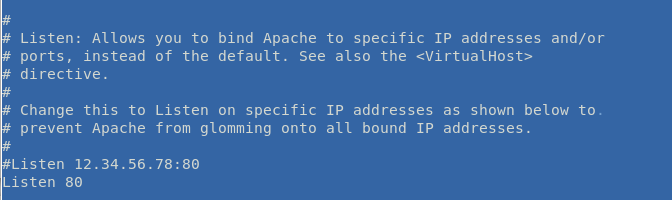


Figure 19: Исправление файла

## Шаг 18

Попыталась удалить привязку http\_port\_t к 81 порту, но сделать это не удалось. Затем удалила файл /var/www/html/test.html (рис. 20).

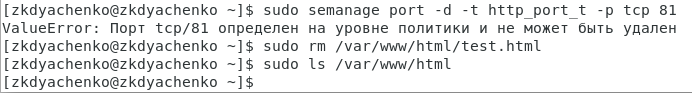


Figure 20: Попытка удалить привязку http\_port\_t к 81 порту и удаление файла

# Выводы

В результате работы я развила навыки администрирования ОС Linux, получила первое практическое знакомство с технологией SELinux. Также я проверила работу SELinux на практике совместно с веб-сервером Apache. Результаты работы находятся в [репозитории на GitHub](https://github.com/ZlataDyachenko/workD), а также есть [скринкаст выполнения лабораторной работы](https://www.youtube.com/watch?v=fbgxymqFADg).