Лабораторная работа № 6

Мандатное разграничение прав в Linux

Тарусов Артём Сергеевич

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.

# Задание

1. Настроить и запустить сервер Apache.
2. Исследовать влияние параметров сервера на его работу.

# Теоретическое введение

* Операционная система — это комплекс программ, предназначенных для управления ресурсами компьютера и организации взаимодействия с пользователем [1].
* Права доступа определяют, какие действия конкретный пользователь может или не может совершать с определенным файлами и каталогами. С помощью разрешений можно создать надежную среду — такую, в которой никто не может поменять содержимое ваших документов или повредить системные файлы. [2].

# Выполнение лабораторной работы

1. Войдем в систему с полученными учётными данными и убедимся, что SELinux работает в режиме enforcing политики targeted (fig. 1).

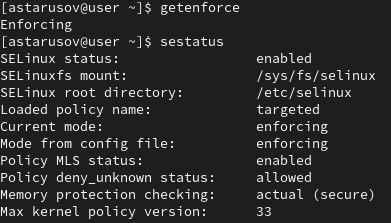


Рис. 1: Конфигурация SELinux

1. Обратимся с помощью браузера к веб-серверу, запущенному на нашем компьютере, и убедимся, что последний работает (fig. 2).

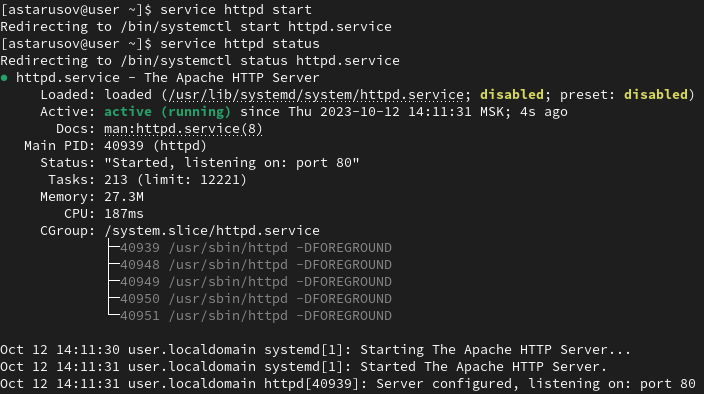


Рис. 2: Обращение к веб-серверу

1. Найдем веб-сервер Apache в списке процессов, определим его контекст безопасности (fig. 3).

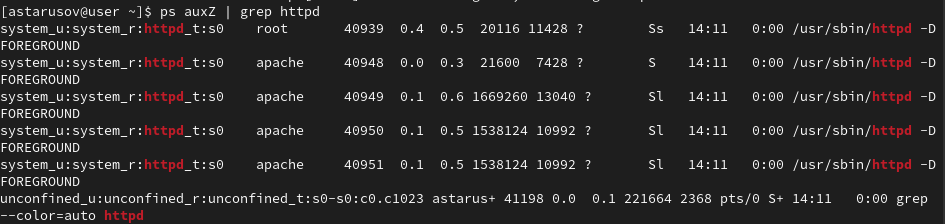


Рис. 3: Контекст безопасности веб-сервера Apache

1. Посмотрим текущее состояние переключателей SELinux для Apache (fig. 4).

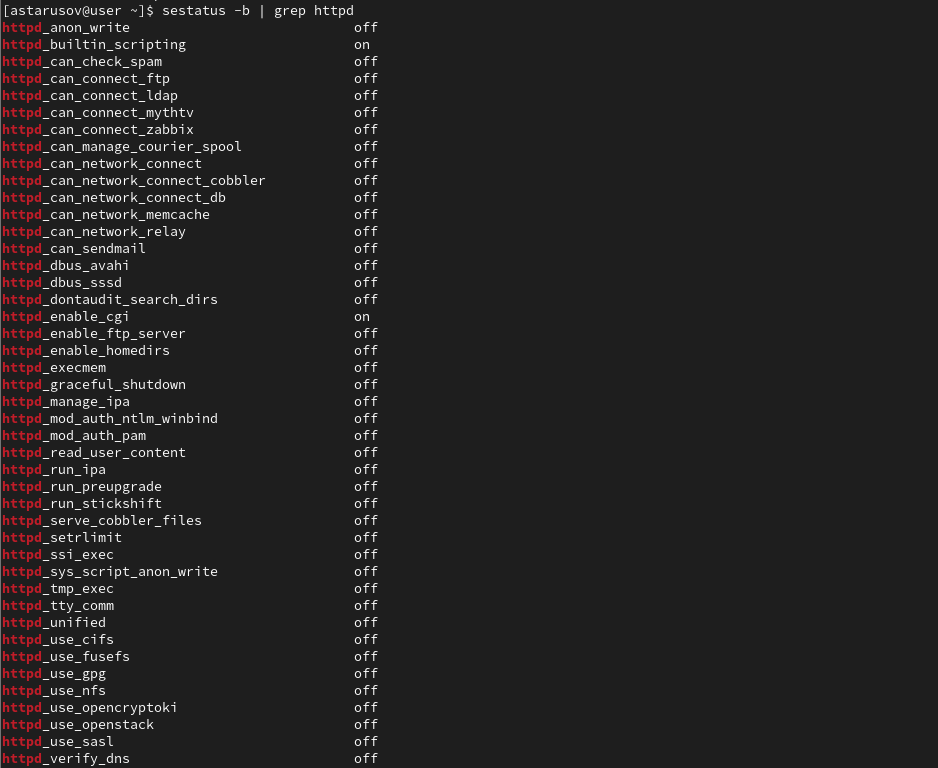


Рис. 4: Текущее состояние переключателей SELinux для Apache

1. Посмотрим статистику по политике с помощью команды seinfo (fig. 5).



Рис. 5: Статистика по политике

1. Определим тип файлов и поддиректорий, находящихся в директории /var/www (fig. 6).

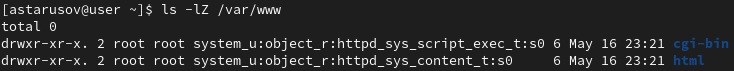


Рис. 6: Тип файлов и поддиректорий, находящихся в директории /var/www

1. Определим тип файлов, находящихся в директории /var/www/html (fig. 7).

Тип файлов, находящихся в директории /var/www/html

Рис. 7: Тип файлов, находящихся в директории /var/www/html

1. Определим круг пользователей, которым разрешено создание файлов в директории /var/www/html (fig. 8).

Круг пользователей, которым разрешено создание файлов в директории /var/www/html

Рис. 8: Круг пользователей, которым разрешено создание файлов в директории /var/www/html

1. Создадим от имени суперпользователя html-файл /var/www/html/test.html (fig. 9).

Создание файла /var/www/html/test.html

Рис. 9: Создание файла /var/www/html/test.html

Заполним его следующим содержимым:

<html>  
<body>test</body>  
</html>

1. Проверим контекст созданного нами файла (fig. 10).

Работа с параметрами readfile

Рис. 10: Работа с параметрами readfile

Как видим по умолчанию присваивается контекст unconfined\_u:object\_r:httpd\_sys\_content\_t:s0

1. Обратимся к файлу через веб-сервер, введя в браузере адрес http://127.0.0.1/test.html. Убедимся, что файл был успешно отображён (fig. 11).

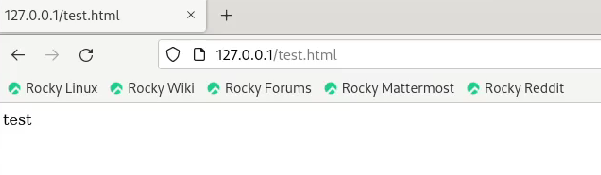


Рис. 11: Файл test.html в браузере

1. Изучим справку man httpd\_selinux и выясним, какие контексты файлов определены для httpd. Сопоставим их с типом файла test.html (fig. 12).

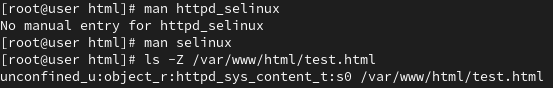


Рис. 12: Вызов справки и тип файла test.html

1. Изменим контекст файла /var/www/html/test.html с httpd\_sys\_content\_t на samba\_share\_t (fig. 13).

Изменение контекста

Рис. 13: Изменение контекста

14.Попробуем ещё раз получить доступ к файлу через веб-сервер (fig. 14).

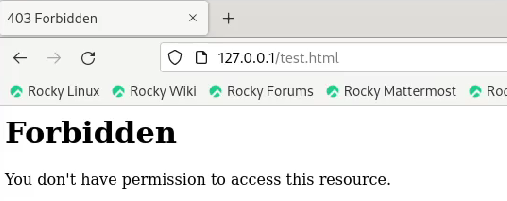


Рис. 14: Файл test.html в браузере после изменения контекста

1. Просмотрим log-файлы веб-сервера Apache и системный лог-файл (fig. 15).

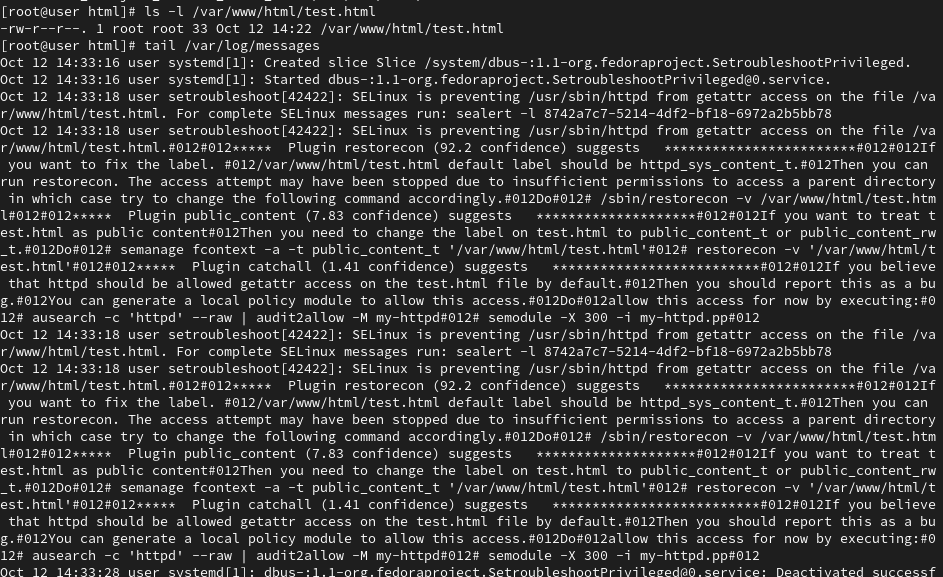


Рис. 15: Содержимое логов

Как видим, нам не удалось получить доступ к файлу как раз из-за измененного контекста.

1. Попробуем запустить веб-сервер Apache на прослушивание ТСР-порта 81 (fig. 16).

Изменение содержимого файла /etc/httpd/httpd.conf

Рис. 16: Изменение содержимого файла /etc/httpd/httpd.conf

1. Выполним перезапуск веб-сервера. Сбоя не произошло (fig. 17).

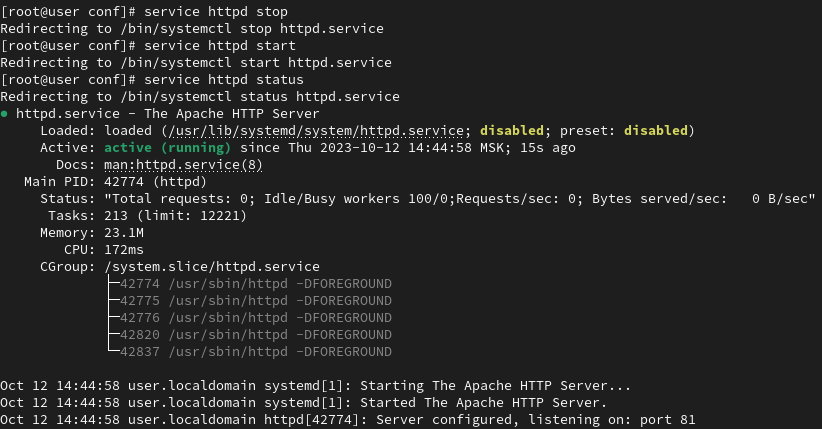


Рис. 17: Перезапуск веб-сервера

1. Проанализируем лог-файлы (fig. 18).

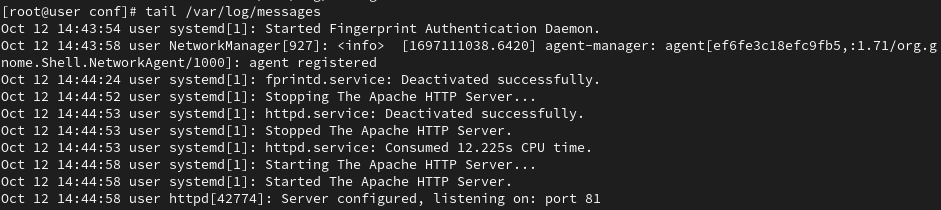


Рис. 18: Лог-файл tail -nl /var/log/messages

1. Выполним команду semanage port -a -t http\_port\_t -р tcp 81. После этого проверим список портов командой semanage port -l | grep http\_port\_t Убедимся, что порт 81 есть в списке. (fig. 19).

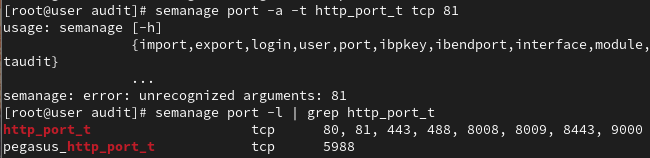


Рис. 19: Попытка добавления порта 81 в список и вывод списка допустимых портов

1. Попробуем запустить веб-сервер Apache ещё раз (fig. 20).

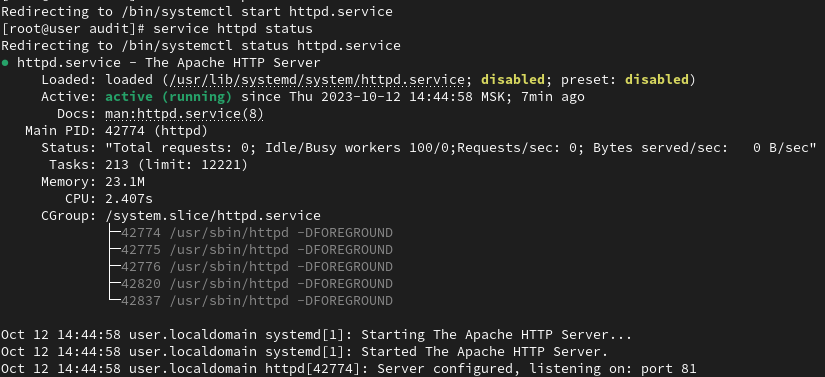


Рис. 20: Повторный запуск веб-сервера

1. Вернем контекст httpd\_sys\_cоntent\_\_t к файлу /var/www/html/ test.html. Попробуем получить доступ к файлу через веб-сервер (fig. 21).

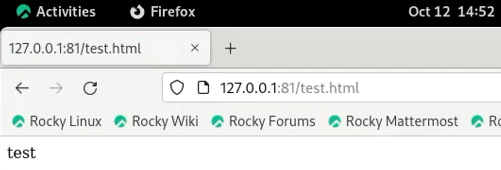


Рис. 21: Файл test.html в браузере после возвращения контекста

1. Исправим обратно конфигурационный файл apache, вернув Listen 80 (fig. 22).

Параметр Listen после возвращения значения

Рис. 22: Параметр Listen после возвращения значения

1. Попробуем удалить привязку http\_port\_t к 81. Удаление невозможно(fig. 23).

Попытка удаления привязки к порту 81

Рис. 23: Попытка удаления привязки к порту 81

1. Удалим файл /var/www/html/test.html(fig. 24).

Удаление файла /var/www/html/test.html

Рис. 24: Удаление файла /var/www/html/test.html

# Выводы

В рамках данной лабораторной работы были развиты навыки администрирования ОС Linux. Получено первое практическое знакомство с технологией SELinux1. Проверена работа SELinx на практике совместно с веб-сервером Apache.

# Список литературы

[1] https://blog.skillfactory.ru/glossary/operaczionnaya-sistema/

[2] https://codechick.io/tutorials/unix-linux/unix-linux-permissions