Отчёт по лабораторной работе №6

Дисциплина: Основы информационной безопасности

Балакирева Дарья Сергеевна, НПМбд-01-19б

Содержание

# 1 Цель работы

Получить практические навыки адмирирования в OC Linux и ознакомиться с технологией SELinux совместно с веб-сервером Apache.

# 2 Теоретическое введение

SELinux, или Security Enhanced Linux, — это продвинутый механизм управления доступом, разработанный Агентством национальной безопасности (АНБ) США для предотвращения злонамеренных вторжений. Он реализует мандатную модель управления доступом (MAC — Mandatory Access control) в дополнение к уже существующей в Linux дискреционной модели (DAC — Discretionary Access Control), то есть разрешениям на чтение, запись, выполнение.

У SELinux есть три режима работы:

* Enforcing — ограничение доступа в соответствии с политикой. Запрещено все, что не разрешено в явном виде. Режим по умолчанию.
* Permissive — ведёт лог действий, нарушающих политику, которые в режиме enforcing были бы запрещены, но не запрещает сами действия.
* Disabled — полное отключение SELinux.

В основе структуры безопасности SELinux лежат политики. Политика — это набор правил, определяющих ограничения и права доступа для всего, что есть в системе. Под “всем” в данном случае понимаются пользователи, роли, процессы и файлы. Политика определяет связь этих категорий друг с другом. |

Более подробно см. в [1].

Apache — это свободное программное обеспечение, с помощью которого можно создать веб-сервер. Несмотря на то, что Apache чаще всего называют сервером (более того, его официальное название — Apache HTTP Server) — это всё-таки программа, которую устанавливают на сервер, чтобы добиться определённых результатов.

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 [**gnu-doc-1:bash?**].

# 3 Выполнение лабораторной работы

C помощью команды getenforce убеждаемся, что SELinux работает в режиме enforcing, а с помощью команды sestatus устанавливаем политику targeted (рис. 1).

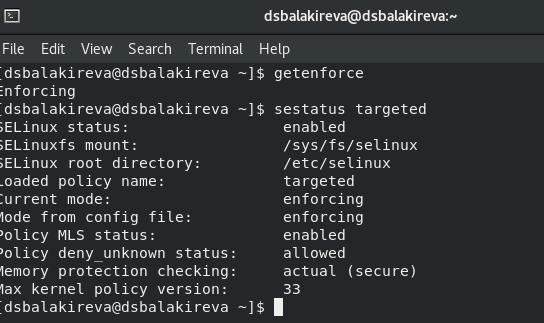


Рис. 1: getenforce и sestatus

Убеждаемся, что сервер работает с помощью команды service httpd status (рис. 2).

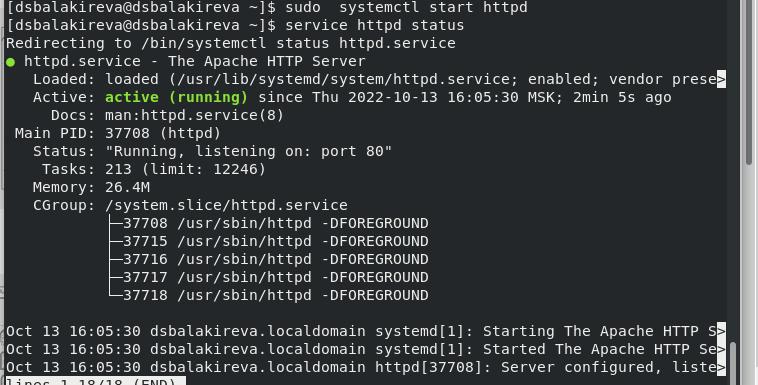


Рис. 2: Работающий сервер

С помощью команды ps -eZ находим, что контекст безопасности Apache — httpd\_t (рис. 3).

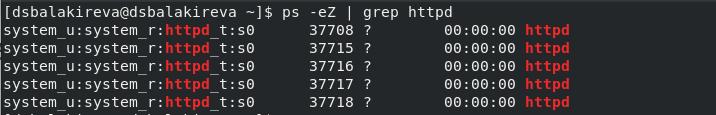


Рис. 3: Контекст безопасности Apache

Смотрим текущее состояние переключателей командой sestatus -b httpd (рис. 4).

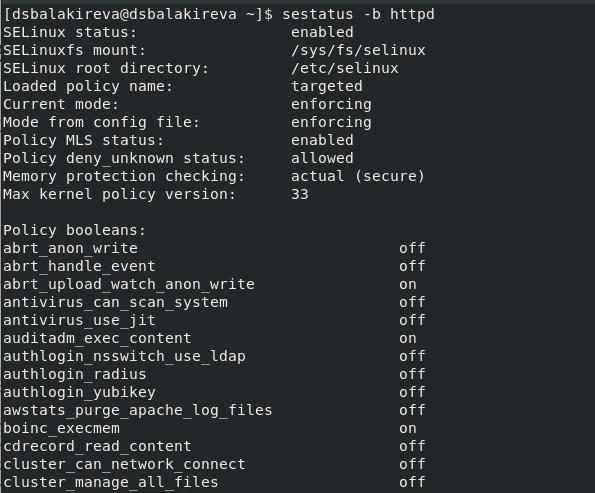


Рис. 4: Состояние переключателей

Смотрим статистику по политике командой seinfo. Узнаём, что множество пользователей — 8, ролей — 14, типов — 4982 (рис. 5).



Рис. 5: Статистика seinfo

Определяем тип файлов и круг пользователей с правой на создание и поддиректорий в директориях /var/www и /var/www/html командой ls -lZ (рис. 6).

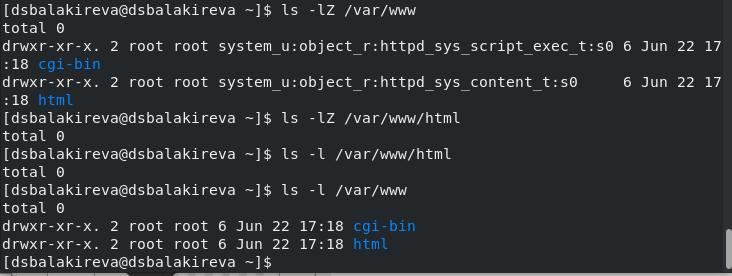


Рис. 6: Данные директорий /var/www и /var/www/html

От имени суперпользователя создаём файл /var/www/html/test.html (рис. 7).

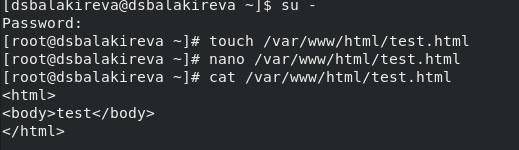


Рис. 7: Файл test.html

Командой matchpathcon узнаём контекст файла test.html и директории /var/www/html — это httpd\_sys\_content\_t (риc. [-#fig:008]).

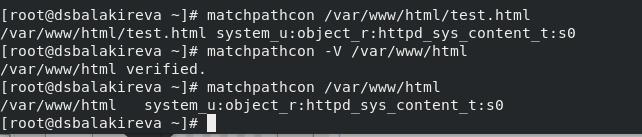


Рис. 8: Констекст файла test.html

Обращаемся к файлу через ссылку в веб-браузере. Контент отображён корректно (рис. 9).

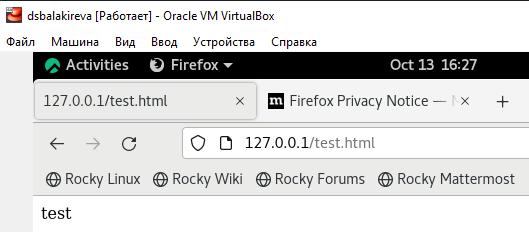


Рис. 9: Просмотр файла в веб-браузере

Изучая справку man httpd\_selunix узнаём, что для httpd определены следующие контексты: httpd\_sys\_content\_t, httpd\_sys\_script\_exec\_t, httpd\_sys\_script\_ro\_t, httpd\_sys\_script\_rw\_t, httpd\_sys\_script\_ra\_t, httpd\_unconfined\_script\_exec\_t. Меняем контекст файла test.html командой chcon -t (рис. 10).



Рис. 10: Смена контекста

При повторной попытке открыть файл через веб-браузер получаем ошибку доступа (рис. 11).

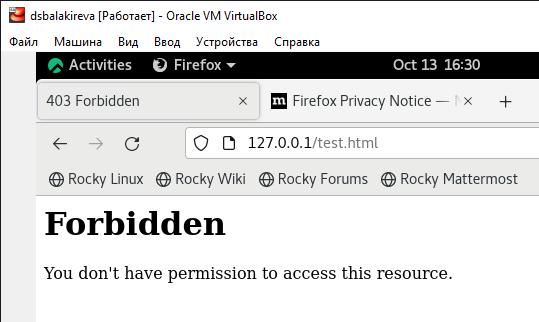


Рис. 11: Ошибка доступа

Убеждаемся, что файл доступен для чтения всем пользователям командой ls -l. Далее смотрим log-файлы веб-сервера Apache командой tail, где показаны ошибки (рис. 12).

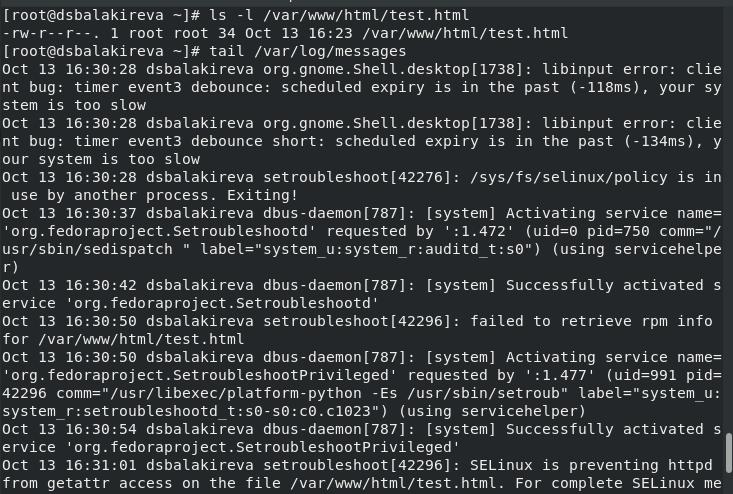


Рис. 12: Ошибки в log-файлах

Устанавливаем веб-сервер Apache на прослушивание TCP-порта 81, изменяя строку Listen в файле /etc/httpd/conf/httpd.conf (рис. 13).

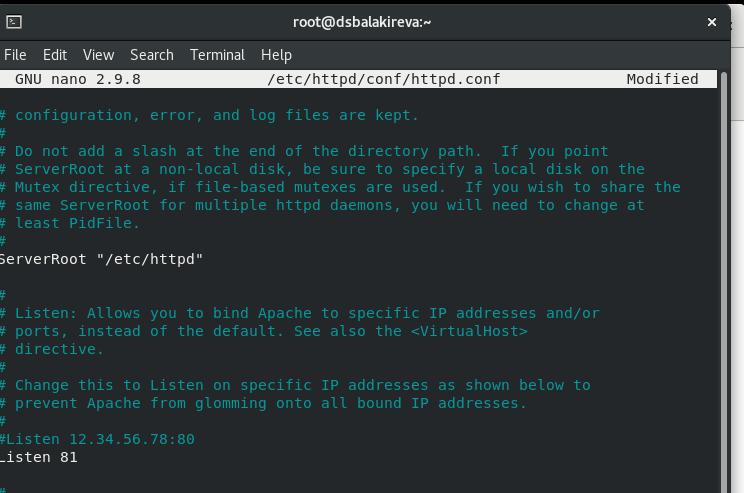


Рис. 13: Прослушивание 81 порта

Перезапускаем сервер и смотри данные log-файлов веб-сервера Apache (рис. 14).

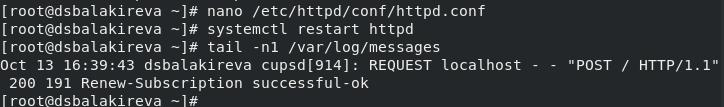


Рис. 14: Перезапуск сервера

Устанавлием для веб-сервера Apache порт TCP-81 и проверяем его наличие в списке портов командой semanage (рис. 15).

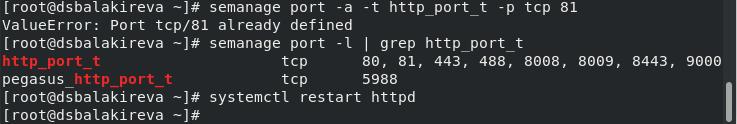


Рис. 15: Установка порта

Возвращаем файлу test.html контекст httpd\_sys\_content\_t и снова успешно просматриваем страницу в веб-браузере (рис. 16).

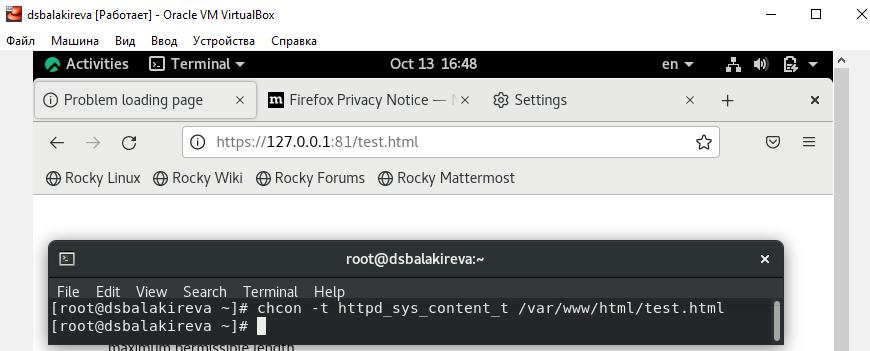


Рис. 16: Повторный просмотр файла в веб-браузере

Возвращаем в конфигурационный файл прослушивание порта 80 и удаляем порт 81 из списка портов (рис. 17).

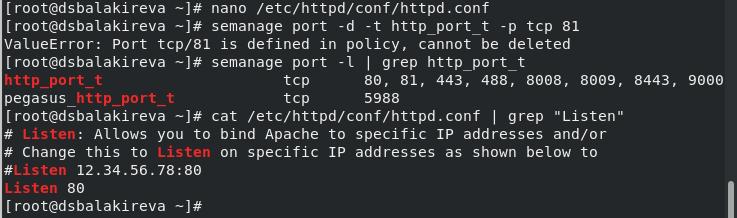


Рис. 17: Удаление порта

Удаляем файл test.html (рис. 18).



Рис. 18: Удаление файла

# 4 Выводы

Я получила основные навыки администрирования в OC Linux и проверила работу SELinux на практике совместно с веб-сервером Apache.

# Список литературы

1. GNU Bash Manual [Электронный ресурс]. Free Software Foundation, 2016. URL: <https://www.gnu.org/software/bash/manual/>.