Отчет по лабораторной работе №6

Информационная безопасность

Чекалова Лилия Руслановна

Содержание

# 1 Цель работы

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

# 2 Задание

* Поиск информации про веб-сервер
* Работа с Html-файлами
* Просмотр лог-файлов

# 3 Теоретическое введение

SELinux представляет собой систему маркировки, каждый процесс файл, каталог, пользователь, устройство, порт и так далее имеет метку. SELinux определяет правила доступа процесса к объектам с определенными метками. Это называется политикой.

Владелец файла не имеет полной свободы действий над атрибутами безопасности. Стандартные атрибуты контроля доступа, такие как группа и владелец ничего не значат для SELinux. Полностью все управляется метками. Значения атрибутов могут быть установлены и без прав root, но на это нужно иметь специальные полномочия SELinux.

SELinux может работать в трех режимах — отключен, система полностью отключена и не работает, режим ограничений Enforcing — программа активирована и блокирует все не соответствующие политикам действия и третий режим Permissive — только фиксировать нарушения.

Политики SELinux бывают тоже нескольких типов. Политика targeted относится к типу Type Enforcment (TE) политик, в которых управление доступом к файлам осуществляется на основе ролей. Сюда же относится политика strict. Есть ещё политики Multi-Level Security (MLS), в которых добавлены дополнительные категории.

Более подробно о см. в [1,2].

# 4 Выполнение лабораторной работы

В качестве первого шага лабораторной работы мы проверили режим работы SELinux с помощью команд getenforce и sestatus (рис. [1](#fig:001)).

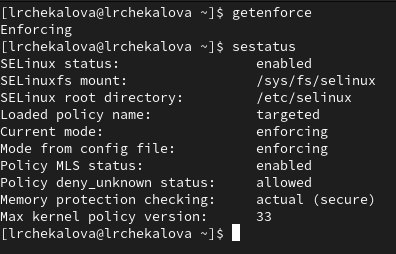


Figure 1: Проверка режима работы

Далее мы проверили, работает ли веб-сервер (рис. [2](#fig:002)), и запустили его, так как он не работал (рис. [3](#fig:003)).

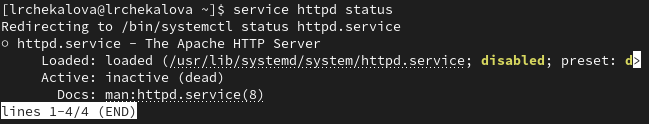


Figure 2: Проверка работы веб-сервера

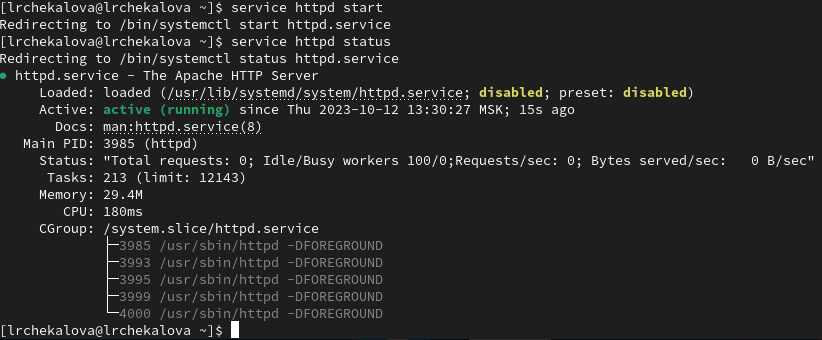


Figure 3: Запуск веб-сервера

Определили контекст безопасности процесса веб-сервера (рис. [4](#fig:004)). Главной информацией для нас стал тип процесса — httpd\_t.

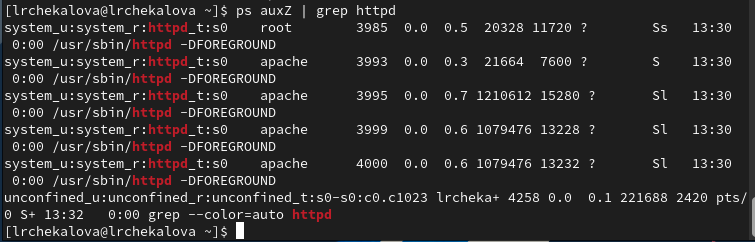


Figure 4: Определение контекста безопасности

Посмотрели текущее положение переключателей SELinux, большинство из них находятся в выключенном состоянии (рис. [5](#fig:005)).

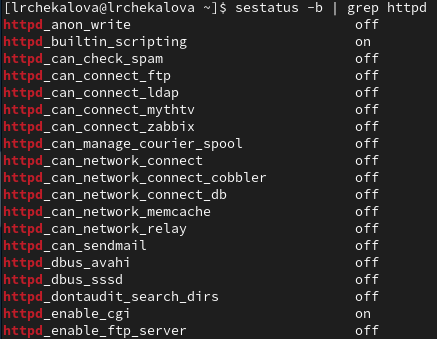


Figure 5: Просмотр переключателей

Посмотрели статистику по политике с помощью seinfo (рис. [6](#fig:006)). Определили, что множество пользователей имеет размер 8, множество ролей — 14, а множество типов — 5100.

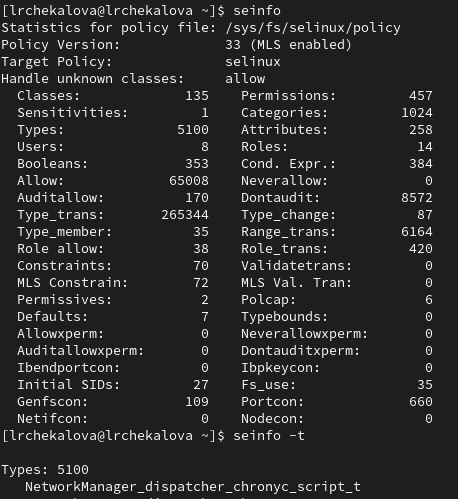


Figure 6: Статистика по политике

Определили тип файлов и поддиректорий директории /var/www (рис. [7](#fig:007)). Поддиректория cgi-bin имеет тип httpd\_sys\_script\_exec\_t, а html — httpd\_sys\_content\_t. Только пользователь-владелец имеет право создавать файлы в папке html.

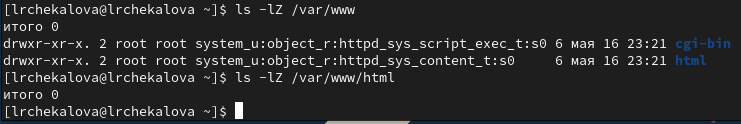


Figure 7: Определение типов поддиректорий

Создали файл test.html в папке html от лица суперпользователя (рис. [8](#fig:008)).

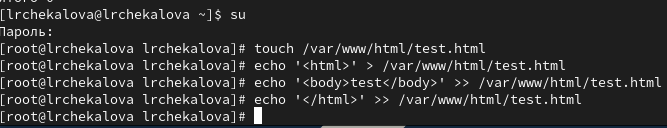


Figure 8: Создание файла test.html

Проверили его контекст (рис. [9](#fig:009)). Вновь созданным файлам в папке html по умолчанию присваивается тип httpd\_sys\_content\_t.

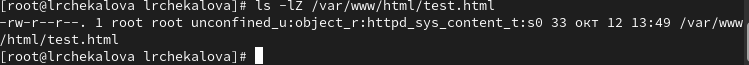


Figure 9: Проверка контекста test.html

Обратились к файлу через веб-сервер и увидели его содержимое (рис. [10](#fig:010)).

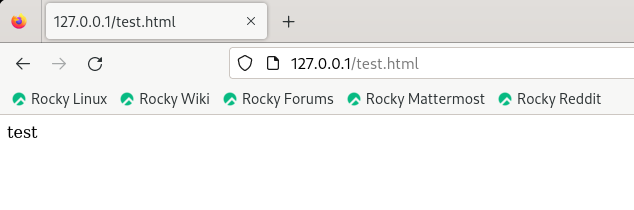


Figure 10: Просмотр файла в браузере

Снова проверили контекст файла и поменяли его на другой (рис. [11](#fig:011)). Новый контекст файла не позволяет процессу httpd получить доступ к файлу при обращении к нему через браузер.

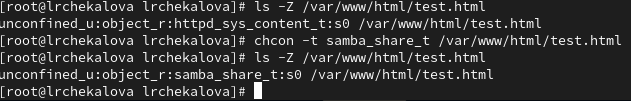


Figure 11: Смена контекста

Попробовали открыть файл в браузере (рис. [12](#fig:012)). Возникла ошибка из-за нового контекста.

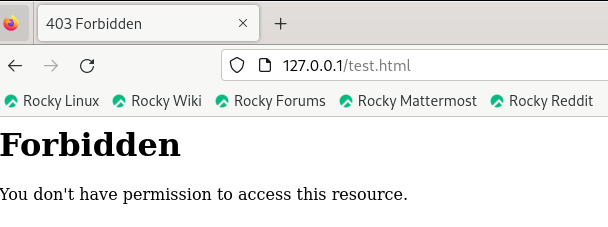


Figure 12: Просмотр файла в браузере

Посмотрели лог-файлы (рис. [13](#fig:013)). Увидели запись о неудачной попытке браузера получить доступ к файлу (ошибка 403).

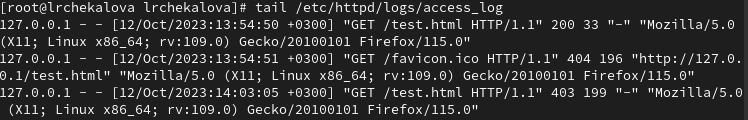


Figure 13: Чтение лог-файлов

Изменили порт в конфигурационном файле httpd.conf с 80 на 81 (рис. [14](#fig:014)).

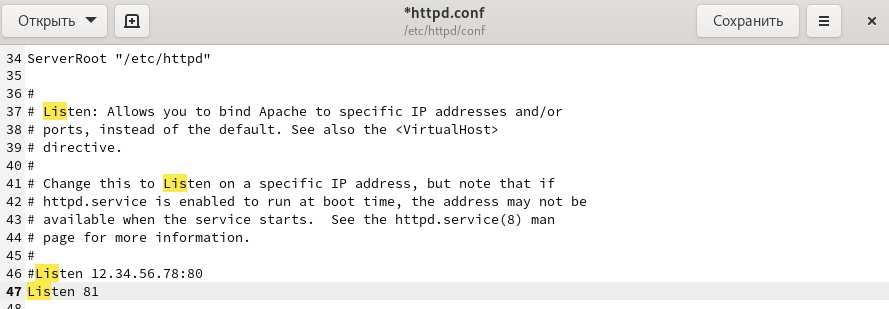


Figure 14: Смена порта

Перезапустили веб-сервер, получили сообщение о том, что он запущен на прослушивание 81 порта (рис. [15](#fig:015)).

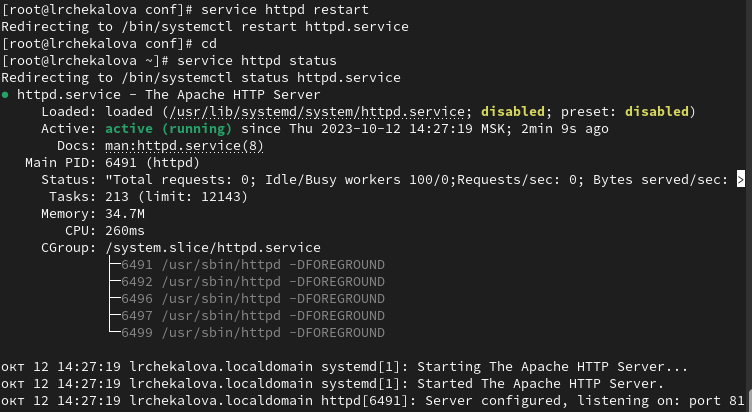


Figure 15: Перезапуск веб-сервера

Проверили лог-файлы и нашли информацию о переключении веб-сервера на прослушивание нового порта (рис. [16](#fig:016)).

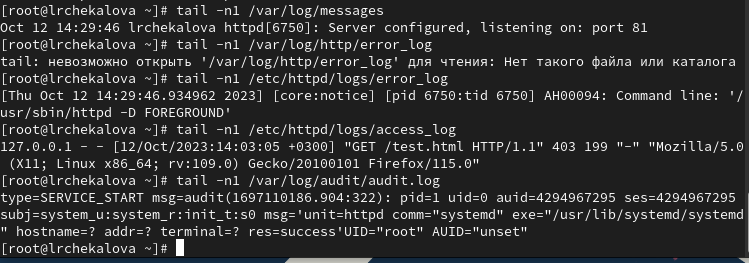


Figure 16: Проверка лог-файлов

Посмотрели список портов веб-сервера, нашли там указанный нами порт (рис. [17](#fig:017)).

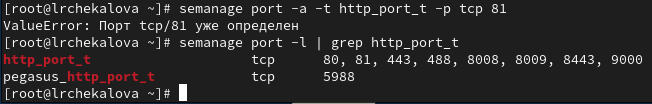


Figure 17: Просмотр списка портов

Вернули файлу test.html старый контекст (рис. [18](#fig:018)).

Figure 18: Смена контекста

Figure 18: Смена контекста

Получили доступ к файлу через веб-сервер в браузере (рис. [19](#fig:019)).

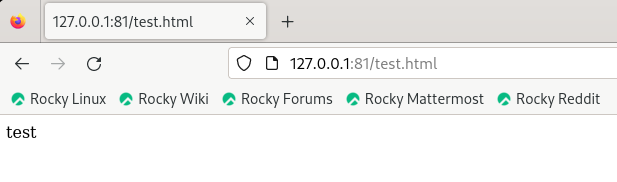


Figure 19: Просмотр файла

Вернули порт 80 в конфигурационном файле (рис. [20](#fig:020)).

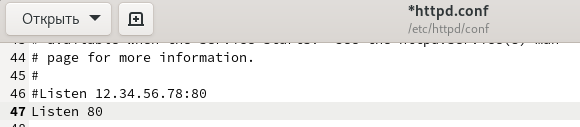


Figure 20: Изменение порта

Попытались удалить 81 порт, но столкнулись с ошибкой, что он определен на уровне политики и не может быть удален (рис. [21](#fig:021)). После этого удалили файл test.html.

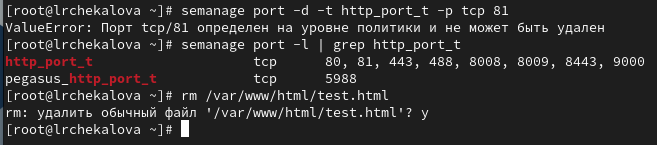


Figure 21: Попытка удаления порта

# 5 Выводы

В результате лабораторной работы я получила базовые навыки администрирования ОС Linux, познакомилась с технологией SELinux и проверила ее работу на практике совместно с веб-сервером Apache.

# Список литературы

1. Мандатное разграничение прав в Linux [Электронный ресурс]. URL: <https://esystem.rudn.ru/pluginfile.php/2090282/mod_resource/content/2/006-lab_selinux.pdf>.

2. Настройка SELinux [Электронный ресурс]. URL: <https://losst.pro/nastrojka-selinux>.