Лабораторная работа №6

Мандатное разграничение прав в Linux

Калинина Кристина Сергеевна

Содержание

# Цель работы

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

# Теоретические сведения

Чтобы преодолеть ограничения и расширить механизмы безопасности, предоставляемые стандартными разрешениями ugo / rwx и списками контроля доступа, Агентство национальной безопасности США (NSA) разработало гибкий метод мандатного контроля доступа (MAC), известный как SELinux (сокращение от Security Enhanced). Linux) для того, чтобы, между прочим, ограничивать способность процессов получать доступ или выполнять другие операции над системными объектами (такими как файлы, каталоги, сетевые порты и т. д.) с наименьшими возможными правами, при этом допуская возможность последующих модификаций этой модели. [1]

Security Enhanced Linux может работать двумя различными способами:

* Enforcing: SELinux запрещает доступ на основе правил политики SELinux, набора руководящих принципов, которые управляют механизмом безопасности.
* Permissive:SELinux не запрещает доступ, но в журнале регистрируются отказы для действий, которые были бы запрещены при запуске в принудительном режиме. [1]

# Выполнение лабораторной работы

1. Вошла в систему с полученными учётными данными, провела подготовку к работе и убедилась, что SELinux работает в режиме enforcing политики targeted (fig. 1).

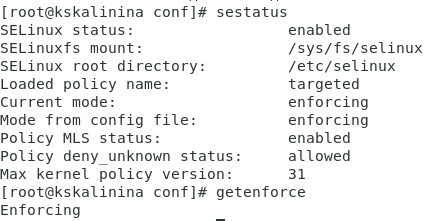


Figure 1: Проверка SELinux

1. Убедилась, что веб-сервис работает, а также посмотрела список процессов (fig. 2).

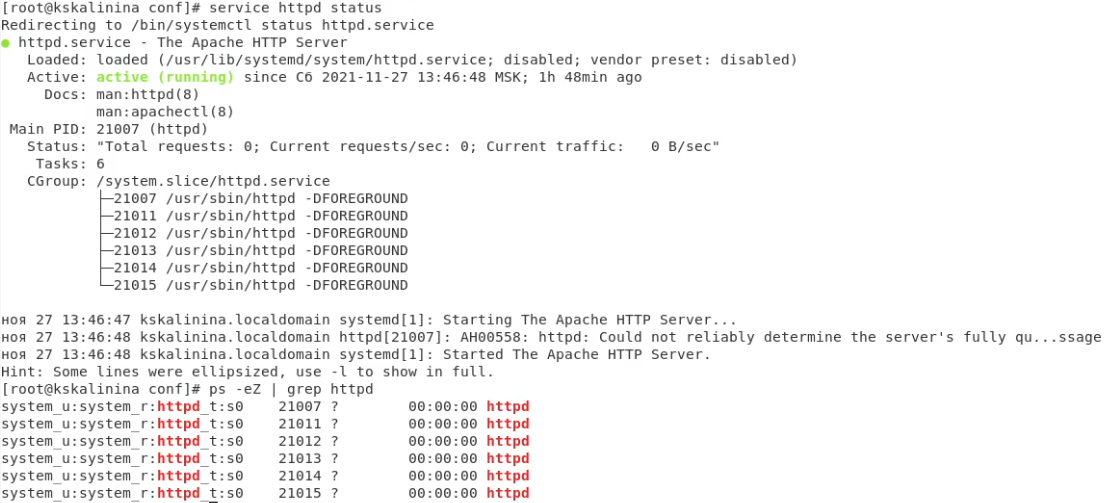


Figure 2: Проверка Apache

1. Посмотрела текущее состояние переключателей SELinux для Apache, многие из них находятся в положении «off» (fig. 3).

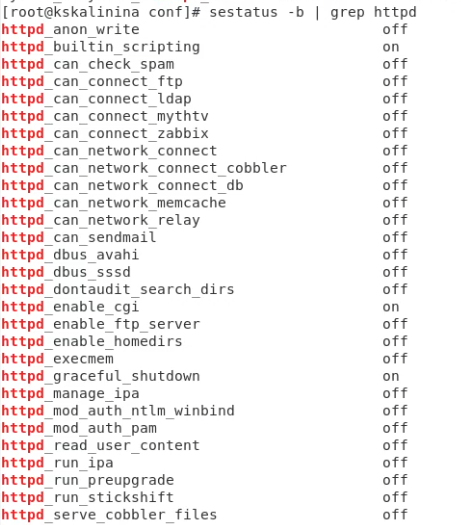


Figure 3: переключатели SELinux для Apache

1. Посмотрела статистику по политике с помощью команды seinfo, также определила множество пользователей, ролей, типов (fig. 4, fig. 5).

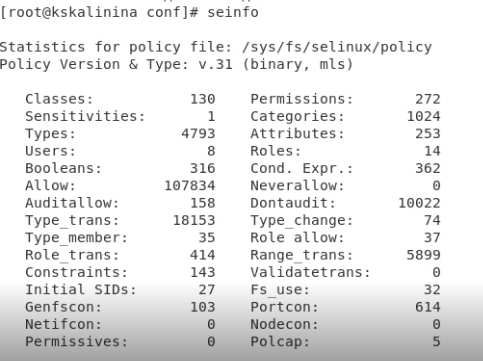


Figure 4: Статистика по политике



Figure 5: Множество пользователей, ролей и типов

1. Просмотрела файлы и поддиректории, находящиеся в директориях “/var/www” и “/var/www/html”. Увидела, что суперпользователь имеет разрешение на создание файлов в директории “/var/www/html” (fig. 6).



Figure 6: Просмотр информации

1. Создала от имени суперпользователя html-файл (fig. 7).

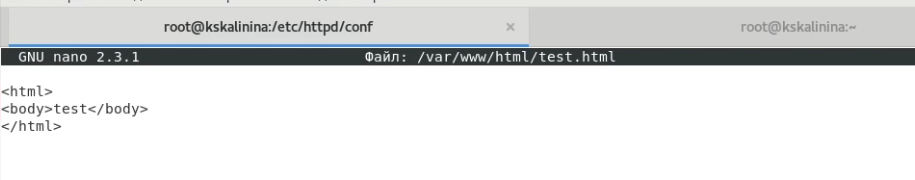


Figure 7: Создание html-файл

1. Проверила контекст созданного файла (fig. 8).

Figure 8: Контекст html-файла, заданный по умолчанию

Figure 8: Контекст html-файла, заданный по умолчанию

1. Обратилась к файлу через веб-сервер, убедилась, что файл был успешно отображён (fig. 9).

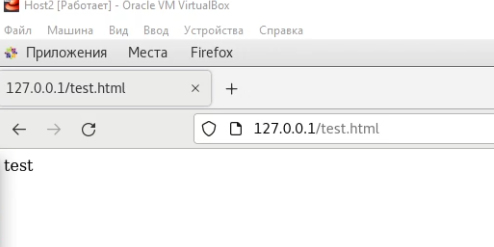


Figure 9: Успешное чтение файла через веб-сервер

1. Изменила контекст файла с httpd\_sys\_content\_t на samba\_share\_t и проверила это (fig. 10).

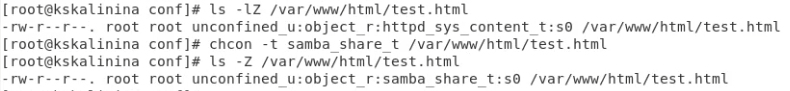


Figure 10: Изменение контекста html-файла

1. Попробовала ещё раз получить доступ к файлу через веб-сервер и получила сообщение об ошибке (fig. 11).

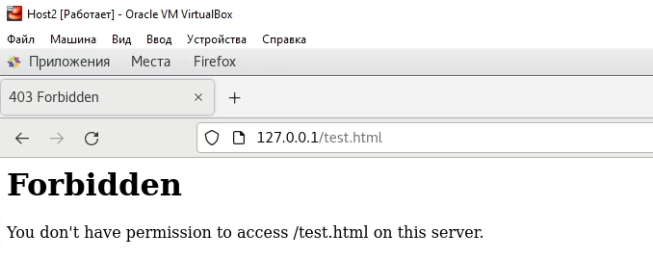


Figure 11: Ошибка доступа к файлу

1. Несмотря на то, что все пользователи могут читать файл, но новый контекст не дает домену доступ к файлу, потому и возникает сообщение об ошибке и невозможность посмотреть файл (fig. 12).



Figure 12: Просмотр прав файла и системного лог-файла

1. Попробовала запустить веб-сервер Apache на прослушивание ТСР-порта 81 (fig. 13).

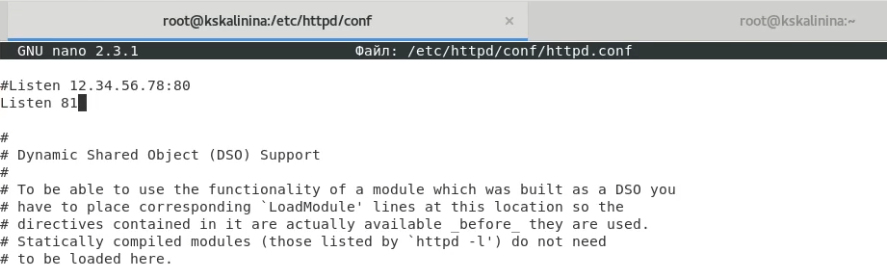


Figure 13: Смена TCP-порта

1. Выполнила перезапуск веб-сервера Apache, всё прошло успешно,т.к. 81 порт по умолчанию был в списке портов (fig. 14, fig. 15).



Figure 14: Перезапуск Apache



Figure 15: Просмотр списка портов

1. Вернула контекст httpd\_sys\_cоntent\_\_t файлу и попробовала получить доступ к файлу через 81 порт (fig. 16, fig. 17).

Figure 16: Смена контекста файла

Figure 16: Смена контекста файла

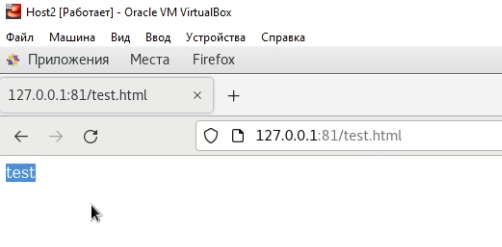


Figure 17: Просмотр файла через 81 порт

1. Исправила обратно конфигурационный файл apache (fig. 18).

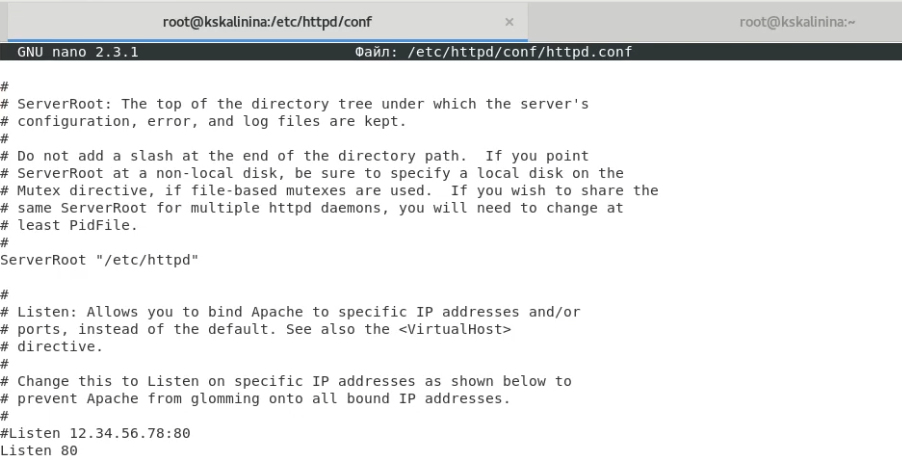


Figure 18: Восстановление файла

1. Попыталась удалить привязку http\_port\_t к 81 порту и удалила созданный файл (fig. 19).

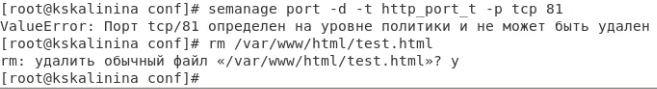


Figure 19: Попытка удаления привязки http\_port\_t к 81 порту и удаление файла

# Выводы

Таким образом я успешно познакомилась с технологией SELinux и проверила работу SELinx на практике совместно с веб-сервером Apache.

# Список литературы

1. Реализация мандатного контроля доступа. // itsecforu.ru 2019. URL: https://itsecforu.ru/2019/07/25/%F0%9F%9B%A1%EF%B8%8F-%D1%80%D0%B5%D0%B0%D0%BB%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F-%D0%BC%D0%B0%D0%BD%D0%B4%D0%B0%D1%82%D0%BD%D0%BE%D0%B3%D0%BE-%D0%BA%D0%BE%D0%BD%D1%82%D1%80%D0%BE%D0%BB%D1%8F-%D0%B4/ (дата обращения 27.11.2021).
2. Д. С. Кулябов, А. В. Королькова, М. Н. Геворкян. Информационная безопасность компьютерных сетей: лабораторные работы. // Факультет физико-математических и естественных наук. M.: РУДН, 2015. 64 с..