Лабораторная работа № 6

Мандатное разграничение прав в Linux

Абдуллаев Сайидазизхон Шухратович

Содержание

# Цель работы

Развить навыки администрирования ОС Linux.Проверить работу SELinx на практике совместно с веб-сервером Apache.

# Задание

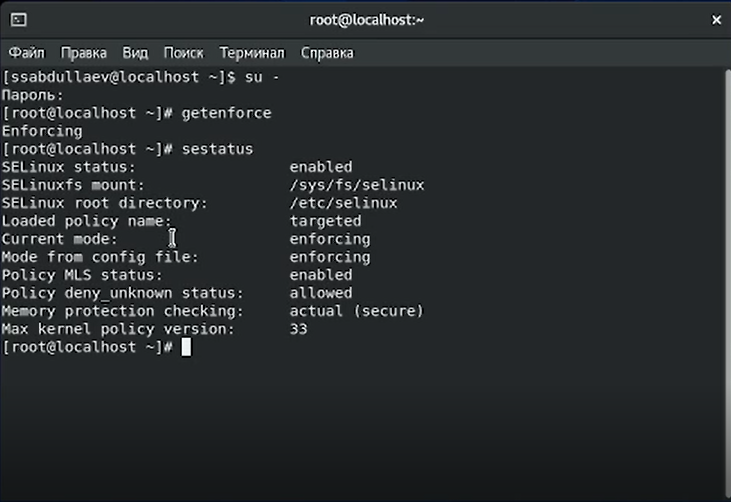
Получить первое практическое знакомство с технологией SELinux

# Теоретическое введение

Для разграничения доступа субъектов — программ к объектам — файлам дерева каталогов используют так называемый мандатный (от англ, mandatory — обязательный или принудительный) подход (MAC, mandotary access control), предполагающий следование обязательным правилам доступа к файлам, назначаемым администраторами системы. Правила доступа строятся на основе знания о внутреннем устройстве программ и представляют собой описание набора минимально необходимых условий их целевого функционирования. Таким образом, в мандатных правилах, ограничивающих доступ к SSH-ключам пользователя, только программе ssh должен быть разрешен доступ для непосредственного выполнения своих прямых функций, а программам firefox и skype в доступе к SSH-ключам должно быть отказано.

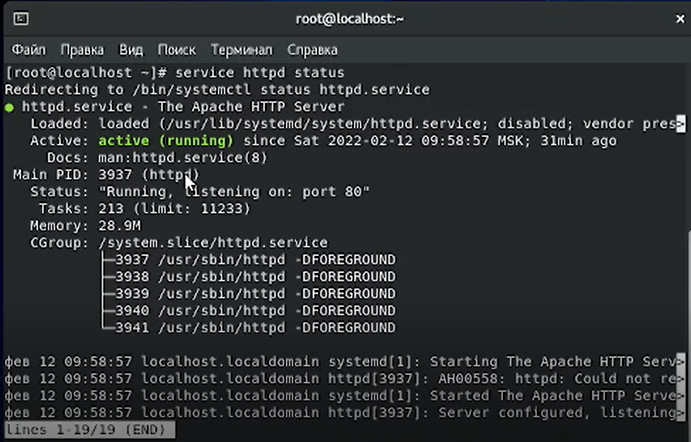
# Ход работы

1. Входим в систему и убеждаемся, что SELinux работает в режиме enforcing политики targeted. (Рис. [-@fig:001]).



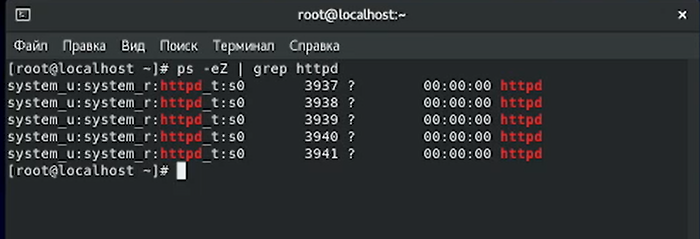
Проверка режима и политики SELinux

1. Проверяем работу веб-сервера. (Рис. [-@fig:002]).



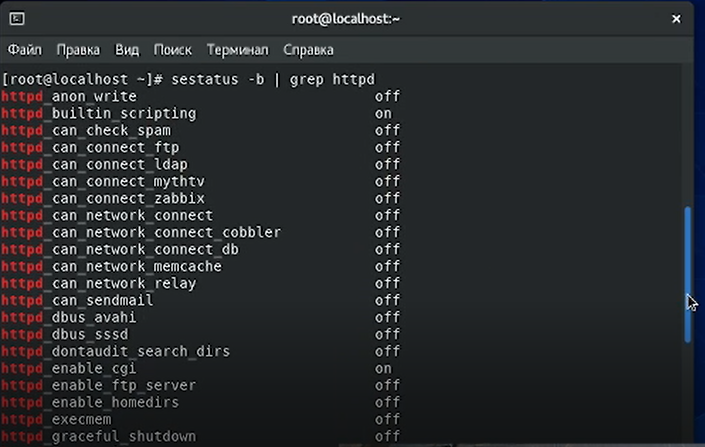
Проверка статуса веб-сервера

1. Определяем контекст безопасности веб-сервера.(Рис. [-@fig:003]).



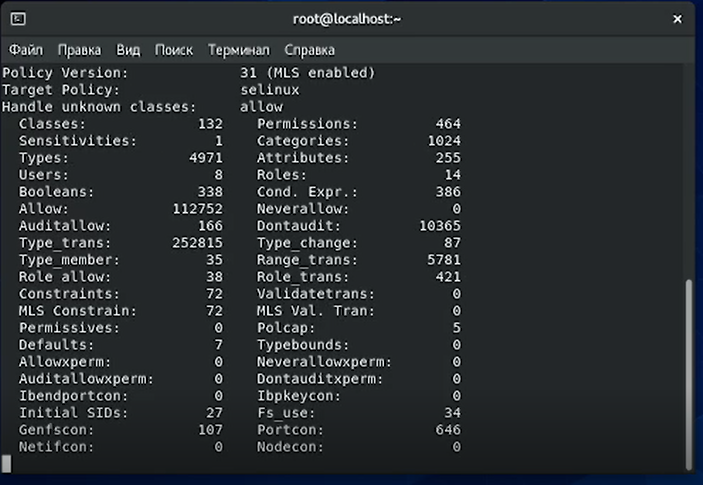
Определение контекста безопасности

1. Теперь посмотрим текущее состояние SELinux переключателей. Как можно заметить, практически все переключатели выключены. (Рис. [-@fig:004]).



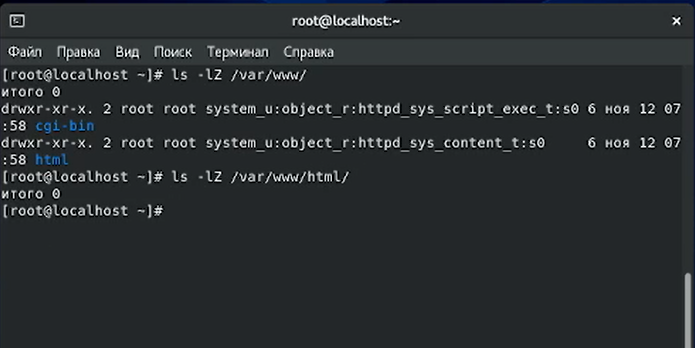
Состояние переключателей

1. Посмотрели статистику по политике. Кроме того, определили множество польхователей, ролей, типов (Рис. [-@fig:005]).



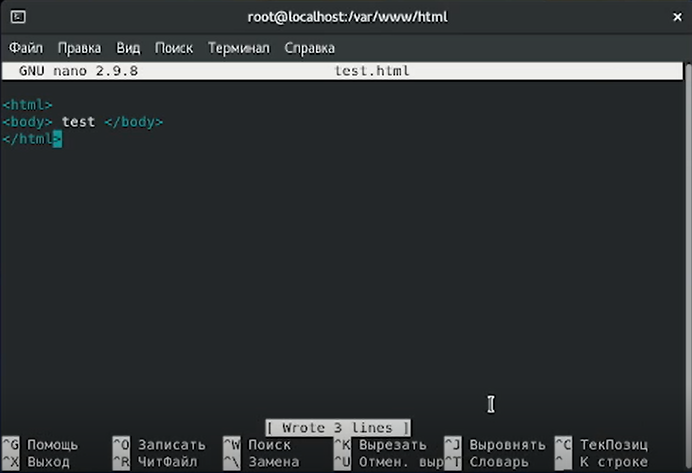
Статистика по политике

6.Определили тип файлов и поддиректорий, находящихся в директории /var/www. Тип файлов, находящихся в директории /var/www/html, определить не удалось, т.к. директория не сожержит файлов. Кроме того, определили круг пользователей, которым разрешено создание файлов в данной директории. Оказалось, что только суперпользователь имеет такое право. (Рис. [-@fig:006]).



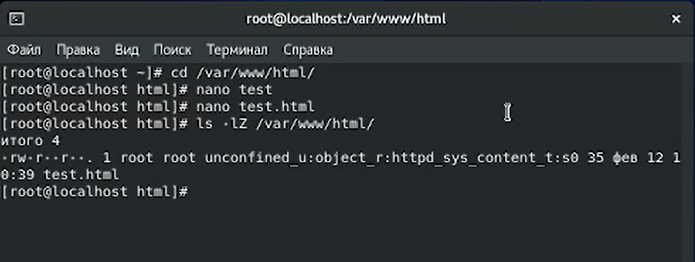
Определяем типы

1. Создали от имени суперпользователя html-файл. (Рис. [-@fig:007]).



Создание html-файла

1. Проверили контекст созданного файла и обратились к файлу через веб-сервер (Рис. [-@fig:008], -@fig:009]).

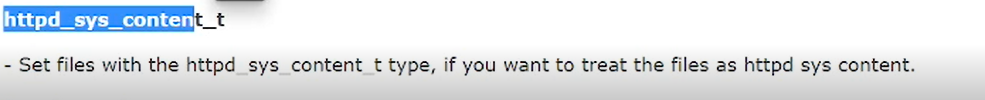


Проверка контекста файла



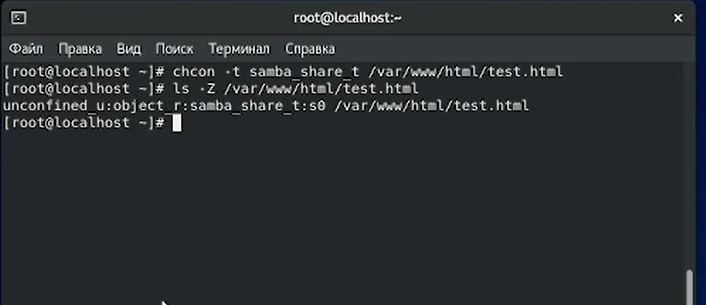
Обращение через веб-сервер

1. Изучили справку man httpd\_selinux и сопоставили их с типом файла test. (Рис. [-@fig:010]).

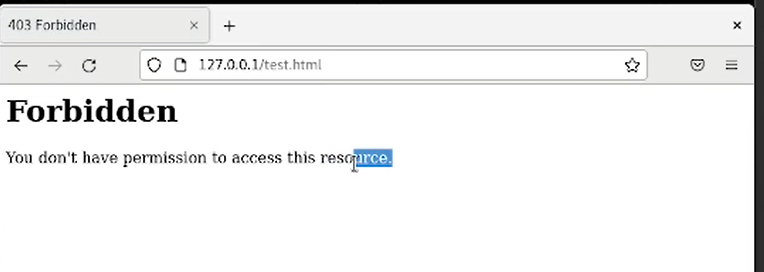


Тип файла

1. Изменили контекст файла на samba\_share\_t и проверили, что контекст поменялся. После этого в браузере получили сообщение об ошибке. Это произошло, поскольку SELinux запретил доступ к файлу (Рис. [-@fig:011], [-@fig:012]).

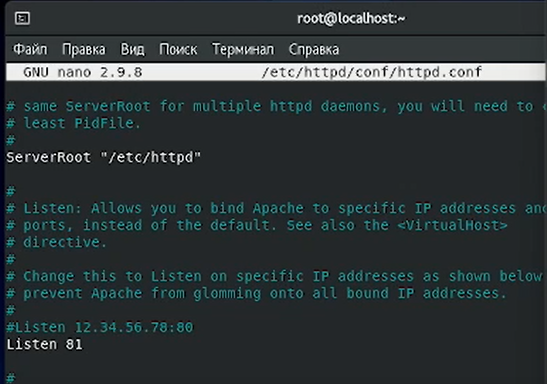


Изменение контекста

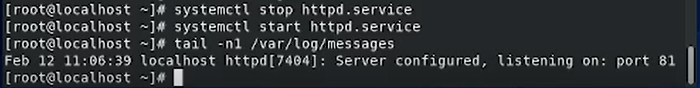


Повторный просмотр файла в браузере

1. Запустили веб-сервер Apache на прослушивание ТСР-порта 81. После чего перезапустили веб-сервер и проанализировали log-файлы. Также проверили список портов (Рис. [-@fig:013], [-@fig:014]).



Запуск веб-сервера

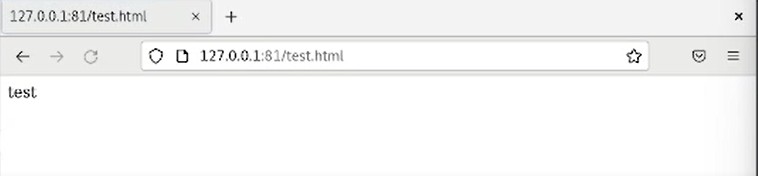


Анализ лог-файлов

1. Снова запустили веб-сервер Apache и вернули контекст httpd\_sys\_cоntent\_\_t к файлу, а затем попробовали получить доступ к файлу через браузер. В результате увидели содержимое файла (Рис. [-@fig:015], [-@fig:016]).

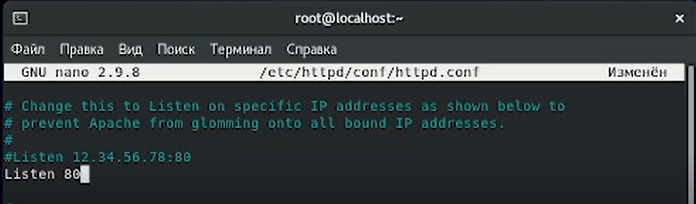


Запуск веб-сервера



Доступ к файлу

1. Исправили обратно конфигурационный файл apache, вернув Listen 80 и попытались удалите привязку http\_port\_t к 81 порту. Вылезла ошибка, поскольку порт 81 определен на уровне политики. После этого удилили html-файл (Рис. [-@fig:017], [-@fig:018]).



Исправление конфигурационного файла

Удаление привезки и файла html

Удаление привезки и файла html

# Выводы

В результате выполнения данной работы была изучена технология SELinux, а также проверена работа SELinux с веб-сервером Apache.