Отчёта по лабораторной работе № 6

Информационная безопасность

Адоле Фейт Эне

Содержание

## 0.1 Цель работы

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

## 0.2 Теоретическое введение

SELinux (Security-Enhanced Linux) обеспечивает усиление защиты путем внесения изменений как на уровне ядра, так и на уровне пространства пользователя, что превращает ее в действительно «непробиваемую» операционную систему. Впервые эта система появилась в четвертой версии CentOS, а в 5 и 6 версии реализация была существенно дополнена и улучшена. SELinux имеет три основных режим работы: • Enforcing: Режим по-умолчанию. При выборе этого режима все действия, которые каким-то образом нарушают текущую политику безопасности, будут блокироваться, а попытка нарушения будет зафиксирована в журнале. • Permissive: В случае использования этого режима, информация о всех действиях, которые нарушают текущую политику безопасности, будут зафиксированы в журнале, но сами действия не будут заблокированы. • Disabled: Полное отключение системы принудительного контроля доступа. Политика SELinux определяет доступ пользователей к ролям, доступ ролей к доменам и доступ доменов к типам. Контекст безопасности — все атрибуты SELinux — роли, типы и домены. Более подробно см. в [1]. Apache — это свободное программное обеспечение, с помощью которого можно создать веб-сервер. Данный продукт возник как доработанная версия другого HTTP-клиента от национального центра суперкомпьютерных приложений (NCSA).

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 [2].

## 0.3 Выполнение лабораторной работы

Вошла в систему под своей учетной записью и убедилась, что SELinux работает в режиме enforcing политики targeted с помощью команд “getenforce” и “sestatus” (рис. 6.1).



Рис. 6.1: Проверка режима enforcing политики targeted

братилась с помощью браузера к веб-серверу, запущенному на моем компьютере, и убедилась, что последний работает с помощью команды “service httpd status” (рис. 6.2).

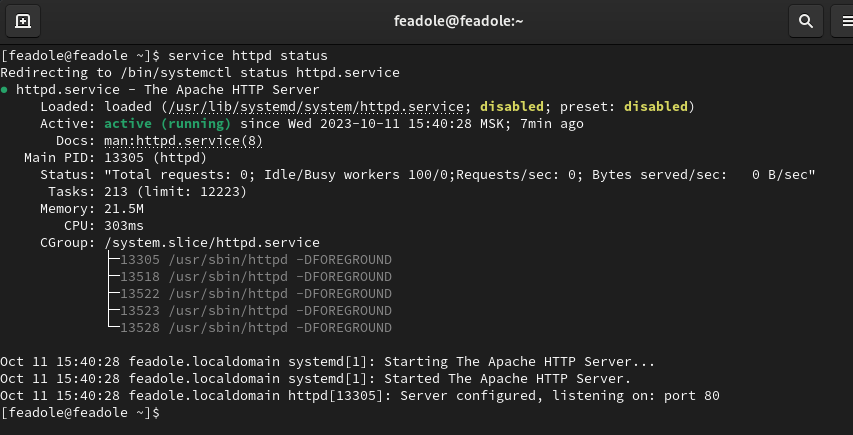


Рис. 6.2: Проверка работы веб-сервера

С помощью команды “ps auxZ | grep httpd” определила контекст безопасности веб-сервера Apache - httpd\_t (рис. 6.3).

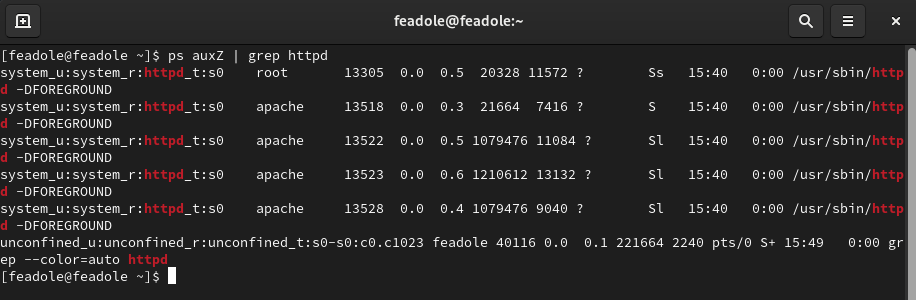


Рис. 6.3: Контекст безопасности веб-сервера Apache

Посмотрела текущее состояние переключателей SELinux для Apache с помощью команды “sestatus -bigrep httpd”, многие из переключателей находятся в положении “off” (рис. 6.4).

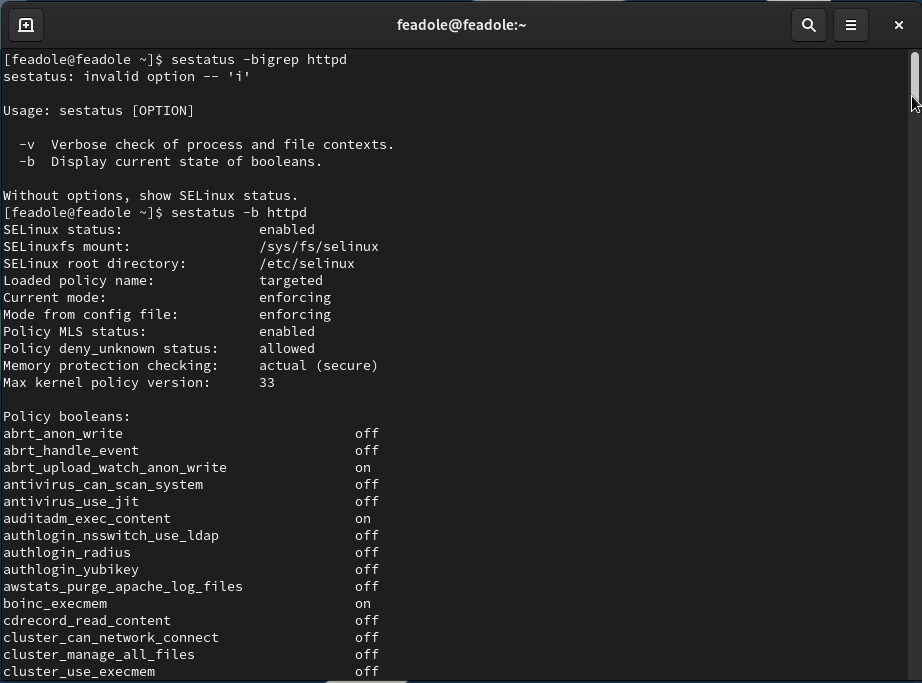


Рис. 6.4: Текущее состояние переключателей SELinux

Посмотрела статистику по политике с помощью команды “seinfo”. Множество пользователей - 8, ролей - 14, типов 4995 (рис. 6.5).

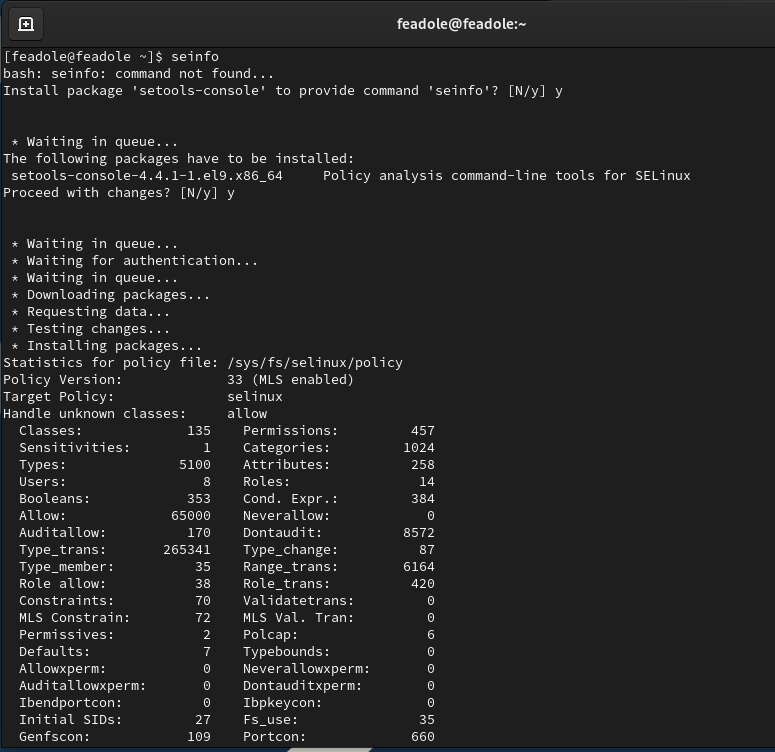


Рис. 6.5: Статистика по политике

С помощью команды “ls -lZ /var/www” посмотрела файлы и поддиректории, находящиеся в директории /var/www. Используя команду “ls -lZ /var/www/html”, определила, что в данной директории файлов нет. Только владелец/суперпользователь может создавать файлы в директории /var/www/html (рис. 6.6).

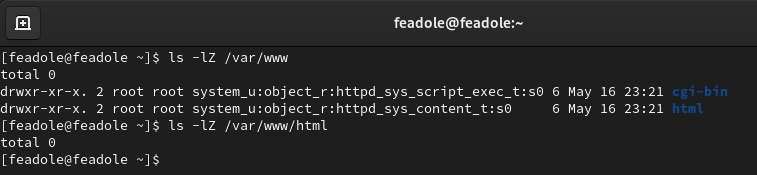


Рис. 6.6: : Просмотр файлов и поддиректориий в директории /var/www

От имени суперпользователя создала html-файл /var/www/html/test.html. Контекст созданного файла - httpd\_sys\_content\_t (рис. 6.7).

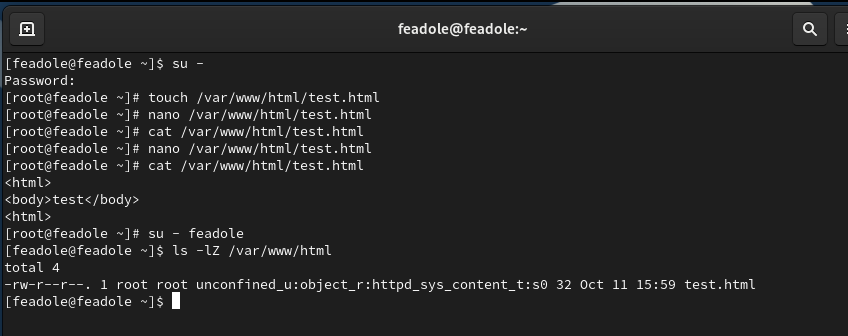


Рис. 6.7: Создание файла /var/www/html/test.html

Обратилась к файлу через веб-сервер, введя в браузере адрес <http://127.0.0.1/test.html>. Файл был успешно отображен (рис. 6.8).

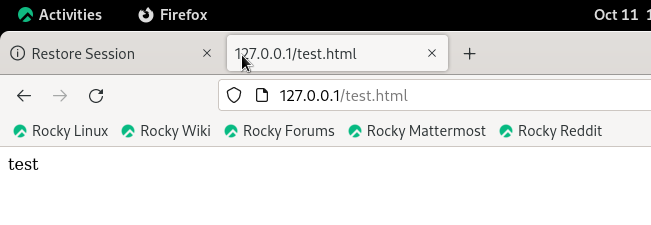


Рис. 6.8: Обращение к файлу через веб-сервер

Изучив справку man httpd\_selinux, выяснила, что для httpd определены следующие контексты файлов: 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. Контекст моего файла - httpd\_sys\_content\_t (в таком случае содержимое должно быть доступно для всех скриптов httpd и для самого демона). Изменила контекст файла на samba\_share\_t командой “sudo chcon -t samba\_share\_t /var/www/html/test.html” и проверила, что контекст поменялся (рис. 6.9).

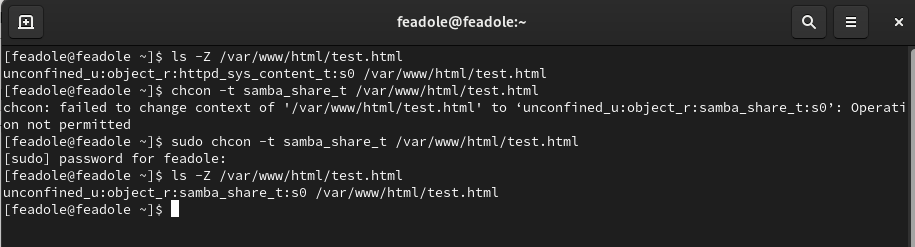


Рис. 6.9: Изменение контекста

Попробовала еще раз получить доступ к файлу через веб-сервер, введя в браузере адрес “http://127.0.0.1/test.html” и получила сообщение об ошибке (т.к. к установленному ранее контексту процесс httpd не имеет доступа) (рис. 6.10).

[Рис. 6.10: Обращение к файлу через веб-сервер](image/lab6.10.png)

Командой “ls -l /var/www/html/test.html” убедилась, что читать данный файл может любой пользователь. Просмотрела системный лог-файл веб-сервера Apache командой “sudo tail /var/log/messages”, отображающий ошибки (рис. 6.11).

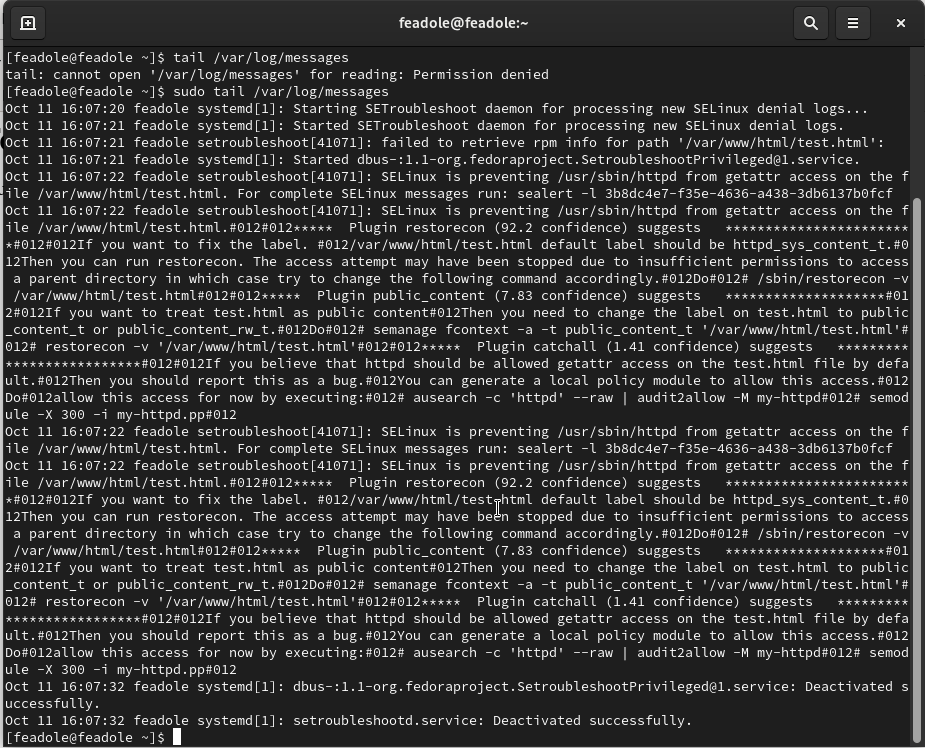


Рис. 6.11: Просмотр log-файла

В файле /etc/httpd/conf/httpd.conf заменила строчку “Listen 80” на “Listen 81”, чтобы установить веб-сервер Apache на прослушивание TCP-порта 81 (рис. 6.12).

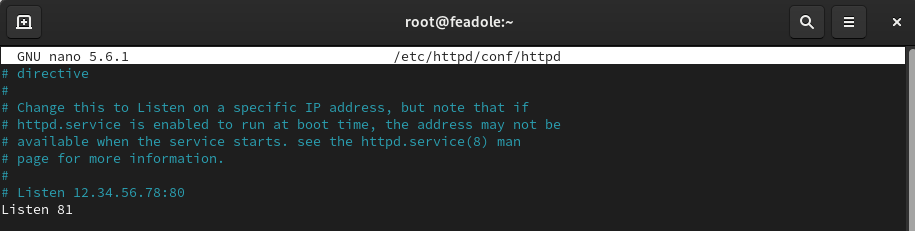


Рис. 6.12: Установка веб-сервера Apache на прослушивание TCP-порта 81

Перезапускаем веб-сервер Apache и анализирует лог-файлы командой “tail -nl /var/log/messages” (рис. 6.13).

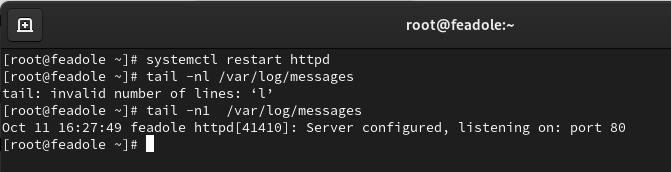


Рис. 6.13: Перезапуск веб-сервера и анализ лог-файлов

Просмотрела файлы “var/log/http/error\_log”, “/var/log/http/access\_log” и “/var/log/audit/audit.log” и выяснила, что запись появилась в последнем файле (рис. 6.14).

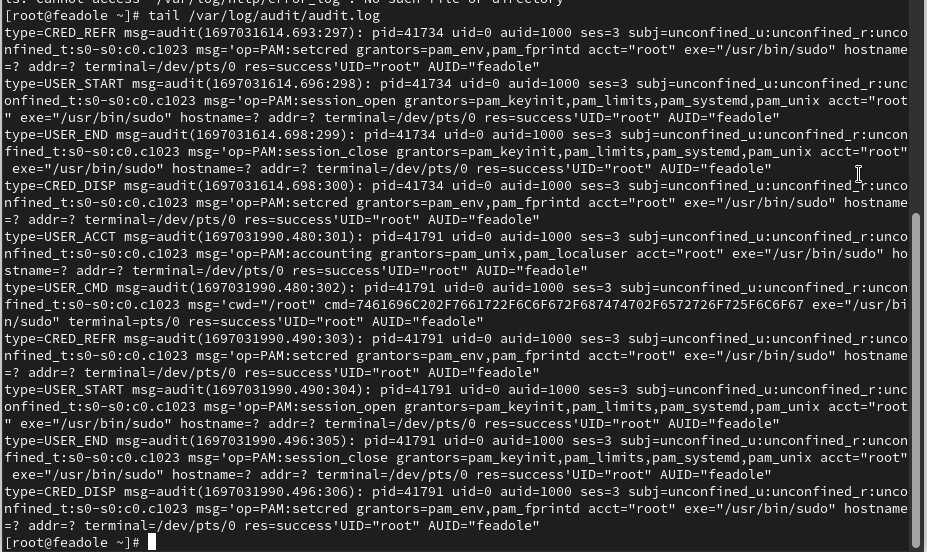


Рис. 6.14: Содержание файла var/log/audit/audit.log

Выполнила команду “semanage port -a -t http\_port\_t -р tcp 81” и убедилась, что порт TCP-81 установлен. Проверила список портов командой “semanage port -l | grep http\_port\_t”, убедилась, что порт 81 есть в списке и запускаем веб-сервер Apache снова (рис. 6.15).

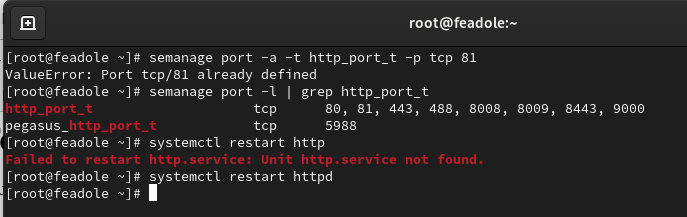


Рис. 6.15: Проверка установки порта 81

Вернула контекст “httpd\_sys\_cоntent\_t” файлу “/var/www/html/test.html” командой “chcon -t httpd\_sys\_content\_t /var/www/html/test.html” (рис. 6.16) и после этого попробовала получить доступ к файлу через веб-сервер, введ “http://127.0.0.1:81/test.html”, в результате чего увидела содежимое файла - слово “test” (рис. 6.17).

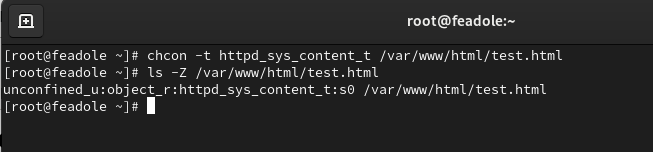


Рис. 6.16: Возвращение исходного контекста файлу

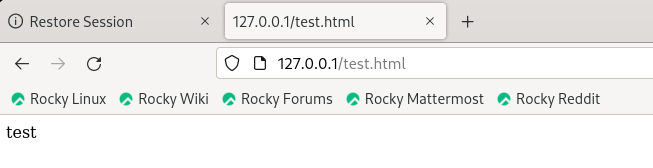


Рис. 6.17: Обращение к файлу через веб-сервер

справила обратно конфигурационный файл apache, вернув “Listen 80”. Попыталась удалить привязку http\_port к 81 порту командой “semanage port -d -t http\_port\_t -p tcp 81”, но этот порт определен на уровне политики, поэтому его нельзя удалить (рис. 6.18).

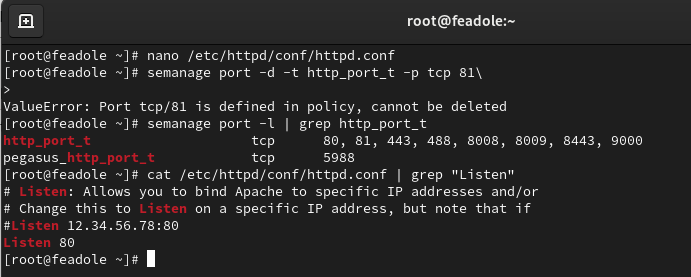


Рис. 6.18: Возвращение Listen 80 и попытка удалить порт 81

Удалила файл “/var/www/html/test.html” командой “rm /var/www/html/test.html” (рис. 6.19).



Рис. 6.19: Удаление файла test.html

## 0.4 Выводы

В ходе выполнения данной лабораторной работы я развила навыки администрирования ОС Linux, получила первое практическое знакомство с технологией SELinux и проверила работу SELinux на практике совместно с веб-сервером Apache.