Отчёта по лабораторной работе № 6

Информационная безопасность

Адебайо Ридвануллахи Айофе

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 Tеорическое 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.

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).

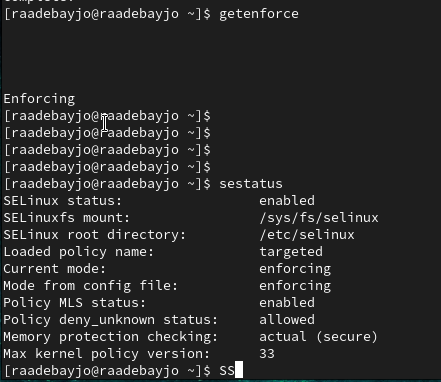
Для чего нужен Apache сервер:

* 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.

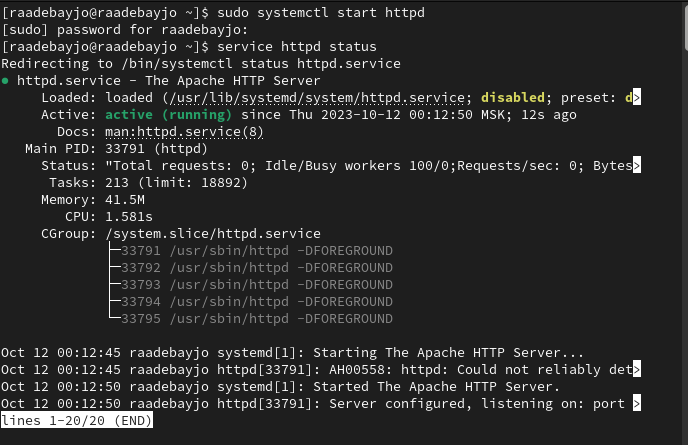
# 3 Выполнение лабораторной работы

Вошел в систему под своей учетной записью и убедился, что SELinux работает в режиме enforcing политики targeted с помощью команд “getenforce” и “sestatus”



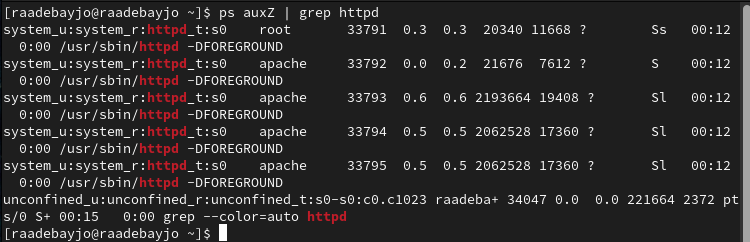
Проверка режима enforcing политики targeted

Обратился с помощью браузера к веб-серверу, запущенному на моем компьютере, и убедился, что последний работает с помощью команды “service httpd status”



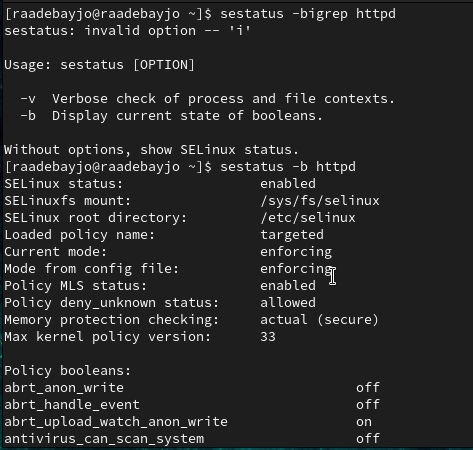
Проверка работы веб-сервера

С помощью команды “ps auxZ | grep httpd” определил контекст безопасности веб-сервера Apache - httpd\_t



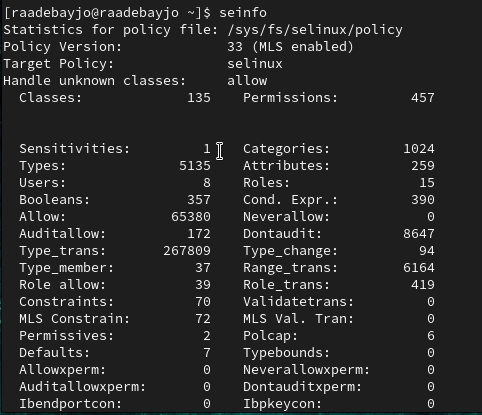
Контекст безопасности веб-сервера Apache

Посмотрел текущее состояние переключателей SELinux для Apache с помощью команды “sestatus -bigrep httpd”, многие из переключателей находятся в положении “off”



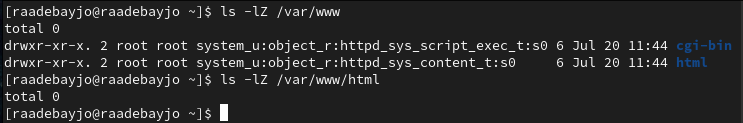
Текущее состояние переключателей SELinux

Посмотрел статистику по политике с помощью команды “seinfo”. Множество пользователей - 8, ролей - 14, типов 4995



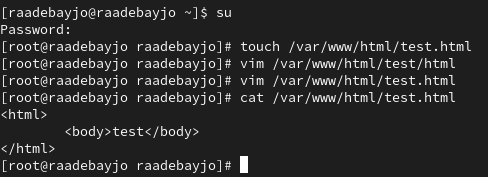
Статистика по политике

С помощью команды “ls -lZ /var/www” посмотрел файлы и поддиректории, находящиеся в директории /var/www. Используя команду “ls -lZ /var/www/html”, определил, что в данной директории файлов нет. Только владелец/суперпользователь может создавать файлы в директории /var/www/html



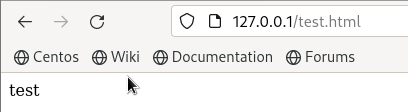
Просмотр файлов и поддиректориий в директории /var/www

От имени суперпользователя создал html-файл /var/www/html/test.html. Контекст созданного файла - httpd\_sys\_content\_t



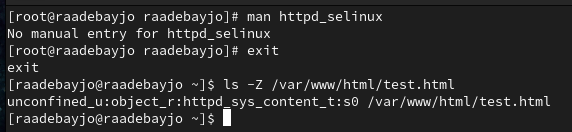
Создание файла /var/www/html/test.html

Обратился к файлу через веб-сервер, введя в браузере адрес “<http://127.0.0.1/test.html”>. Файл был успешно отображен

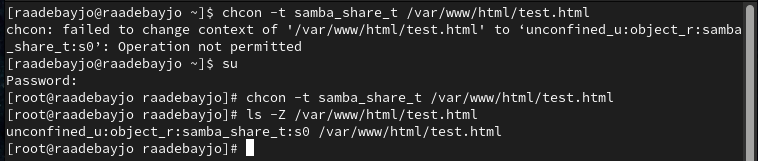


Обращение к файлу через веб-сервер

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). Изменил контекст файла на samba\_share\_t командой “sudo chcon -t samba\_share\_t /var/www/html/test.html” и проверил, что контекст поменялся

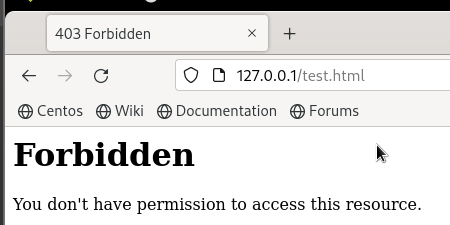


Изменение контекста



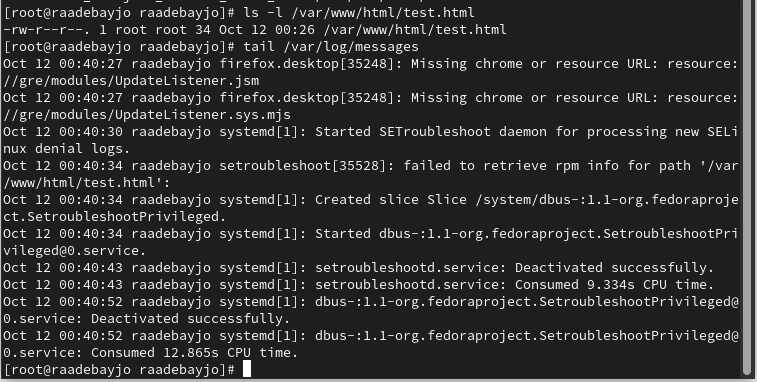
Изменение контекста

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



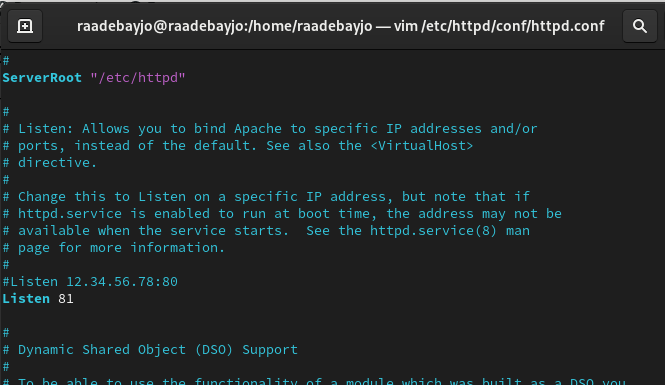
Обращение к файлу через веб-сервер

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



Просмотр log-файла

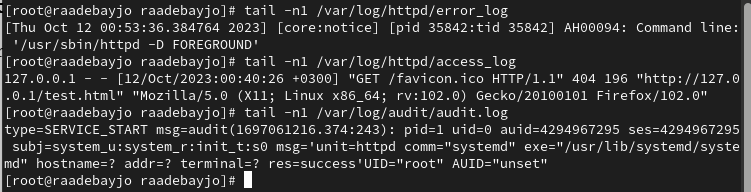
В файле /etc/httpd/conf/httpd.conf заменил строчку “Listen 80” на “Listen 81”, чтобы установить веб-сервер Apache на прослушивание TCP-порта 81



Установка веб-сервера Apache на прослушивание TCP-порта 81

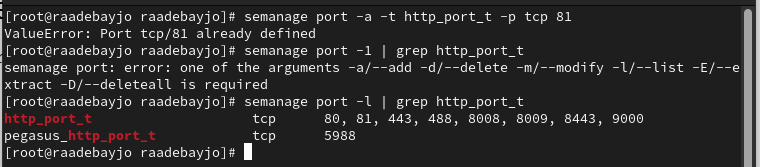
Перезапускаем веб-сервер Apache и анализирует лог-файлы командой “tail -nl /var/log/messages”

Просмотрел файлы “var/log/http/error\_log”, “/var/log/http/access\_log” и “/var/log/audit/audit.log” и выяснил, что запись появился в последнем файле



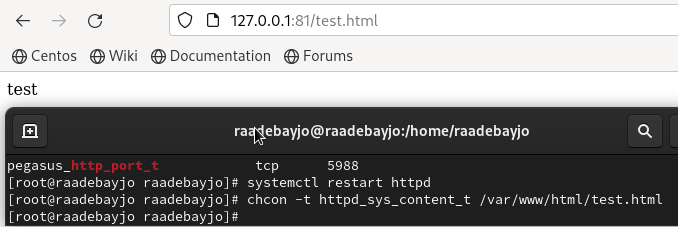
Содержание файла var/log/audit/audit.log

Выполнил команду “semanage port -a -t http\_port\_t -р tcp 81” и убедился, что порт TCP-81 установлен. Проверил список портов командой “semanage port -l | grep http\_port\_t”, убедился, что порт 81 есть в списке и запускаем веб-сервер Apache снова



Проверка установки порта 81

Вернул контекст “httpd\_sys\_cоntent\_t” файлу “/var/www/html/test.html” командой “chcon -t httpd\_sys\_content\_t /var/www/html/test.html” (рис. 3.16) и после этого попробовал получить доступ к файлу через веб-сервер, введя адрес “<http://127.0.0.1:81/test.html”>, в результате чего увидел содежимое файла - слово “test”



Возвращение исходного контекста файлу

Исправил обратно конфигурационный файл apache, вернув “Listen 80”. Попытался 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. Удалил файл “/var/www/html/test.html” командой “rm /var/www/html/test.html”



Возвращение Listen 80 и попытка удалить порт 81

# 4 Выводы

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

# 5 Список литературы

1. Кулябов Д. С. \*Лабораторная работа №5\*\*: 006-lab\_selinux.pdf\*
2. Использование SETUID, SETGID и Sticky bit для расширенной настройки прав доступа в операционных системах Linux [Электронный ресурс]. 2023.URL: <https://ruvds.com/ru/helpcenter/suid-sgid-sticky-bit-linux/> (дата обращения: 05.10.2023)