Отчет по лабораторной работе №6

Администрирование ОС Linux. SELinux

Вакутайпа Милдред

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

Развить навыки администрирования ОС linux. Получить практическое знакомство с SELinux1. Проверить работу SELinux на практике совместно с веб-сервером Apache.

2 Выполнение лабораторной работы

Вошла в систему под своей учетной записью. Убедилась, что SELinux работает в режиме enforcing политики targeted с помощью getenforce и sestatus

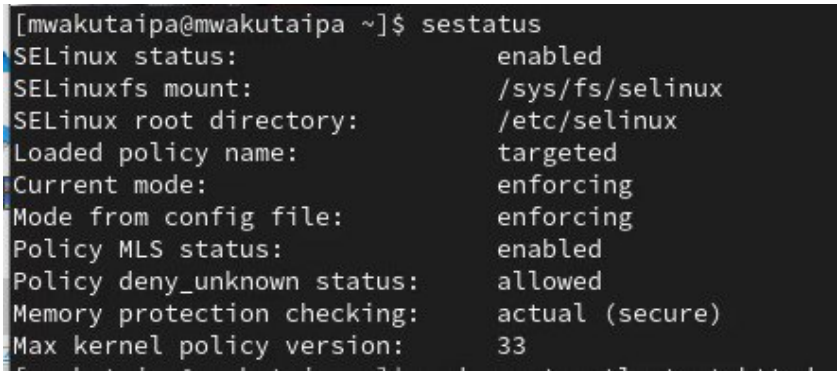

Рис. 2.1: проверка режима работы SELinux

Запускаю сервер apache, далее обращаюсь с помощью браузера к веб-серверу, запущенному на компьютере, он работает, что видно из вывода команды service httpd status

Рис. 2.2: Проверка работы Apache

С помощью команды ps auxZ | grep httpd нашла веб-сервер Apache в списке процессов. Его контекст безопасности - httpd_t

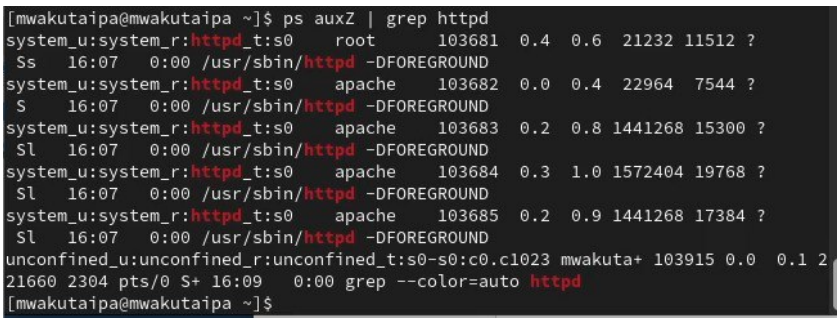

Рис. 2.3: Контекст безопасности Apache

Просмотрела текущее состояние переключателей SELinux для Apache

Рис. 2.4: Состояние переключателей SELinux

Просмотрела статистику по политике с помощью команды seinfo. Множество пользователей - 8, ролей - 39, типов - 5135.

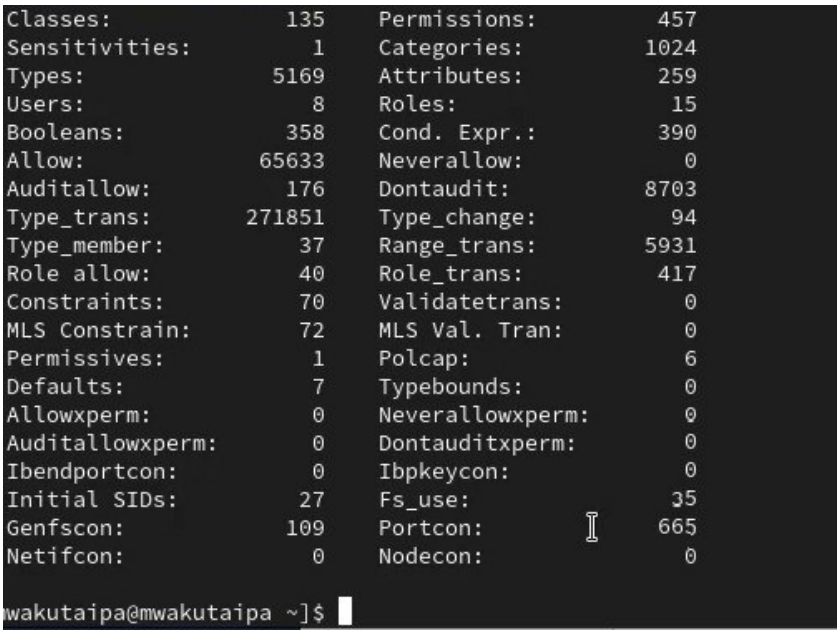

Рис. 2.5: Статистика по политике

Типы поддиректорий, находящихся в директории /var/www, с помощью команды ls -lZ /var/www следующие: владелец - root, права на изменения только у владельца. Файлов в директории нет

Рис. 2.6: Типы поддиректорий

В директории /var/www/html нет файлов.

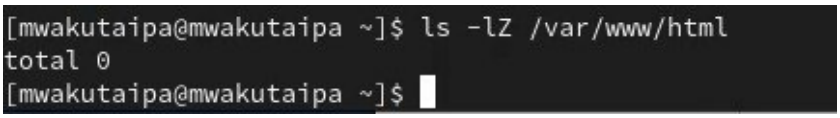

Рис. 2.7: Типы файлов

Создать файл может только суперпользователь, поэтому от его имени создаем файл touch.html со следующим содержанием:

```
<html>
<body>test</body>
</html>
```

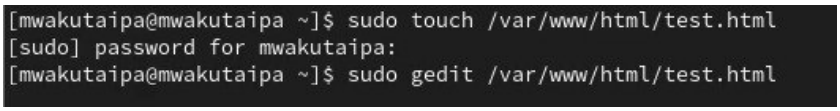

Рис. 2.8: Создание файла

Проверяю контекст созданного файла. По умолчанию это httpd_sys_content_t.

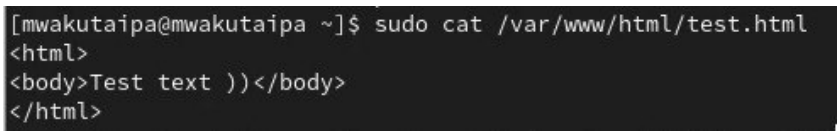

Рис. 2.9: Контекст файла

Обращаюсь к файлу через веб-сервер, введя в браузере адрес http://127.0.0.1/test.html. Файл был успешно отображён.

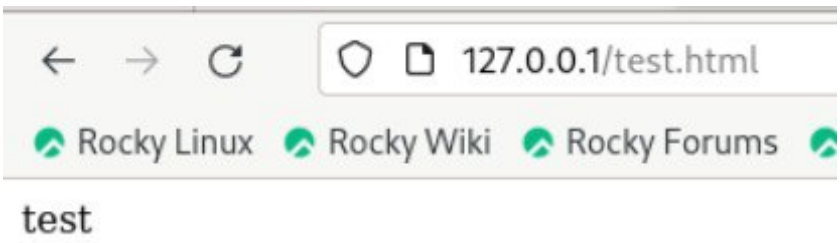

Рис. 2.10: Отображение файла

Изучила справку man httpd_selinux. Рассмотрим полученный контекст детально. Так как по умолчанию пользователи CentOS являются свободными от типа (unconfined в переводе с англ. означает свободный), созданному нами файлу test.html был сопоставлен SELinux, пользователь unconfined_u. Это первая часть контекста. Далее политика ролевого разделения доступа RBAC используется процессами, но не файлами, поэтому роли не имеют никакого значения для файлов. Роль object_r используется по умолчанию для файлов на «постоянных» носителях и на сетевых файловых системах. (В директории/proc файлы, относящиеся к процессам, могут иметь роль system_r. Если активна политика MLS, то могут использоваться и другие роли. Данный случай мы рассматривать не будем, как и предназначение :s0). Тип httpd_sys_content_t позволяет процессу httpd получить доступ к файлу. Благодаря наличию последнего типа мы получили доступ к файлу при обращении к нему через браузер.

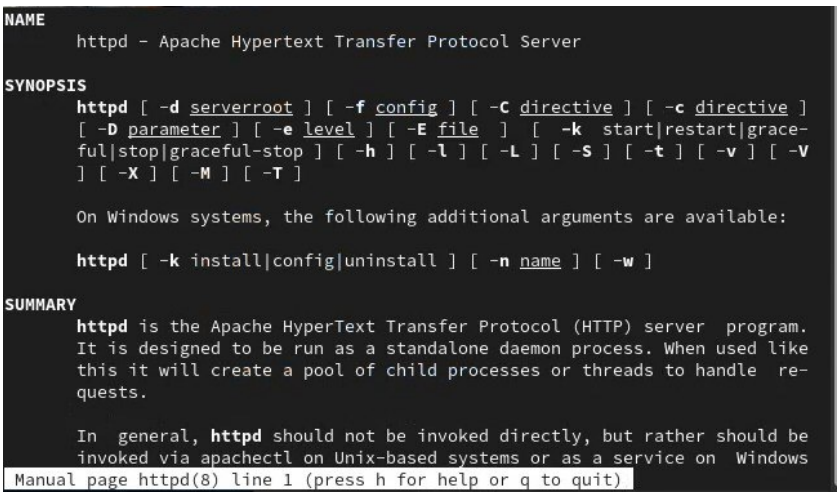

Рис. 2.11: Изучение справки по команде

Изменяю контекст файла test.html с httpd_sys_content_t на любой другой, к которому процесс httpd не должен иметь доступа, например, на samba_share_t: chcon -t samba_share_t /var/www/html/test.html ls -Z /var/www/html/test.html. Контекст действительно поменялся.

Рис. 2.12: Изменение контекста

При попытке отображения файла в браузере получаем сообщение об ошибке.

Рис. 2.13: Отображение файла

файл не был отображён, хотя права доступа позволяют читать этот файл любому пользователю, потому что установлен контекст, к которому процесс httpd не должен иметь доступа. Просматриваю log-файлы веб-сервера Apache и системный лог-файл: tail /var/log/messages. Если в системе окажутся запущенными процессы setroubleshootd и audtd, то вы также сможете увидеть ошибки, аналогичные указанным выше, в файле /var/log/audit/audit.log.

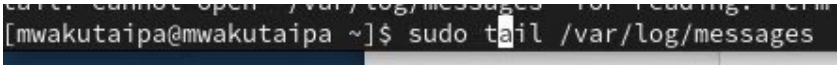

Рис. 2.14: Попытка прочесть лог-файл

Чтобы запустить веб-сервер Apache на прослушивание TCP-порта 81 (а не 80, как рекомендует IANA и прописано в /etc/services) открываю файл /etc/httpd/httpd.conf для изменения. Нахожу строчку Listen 80 и заменяю её на Listen 81.

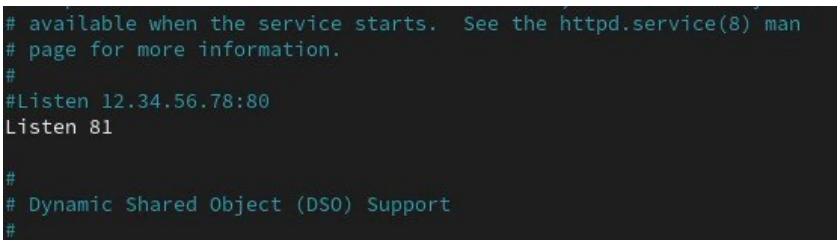

Рис. 2.15: Изменение файла

Выполняю перезапуск веб-сервера Apache. Произошёл сбой, потому что порт 80 для локальной сети, а 81 нет.

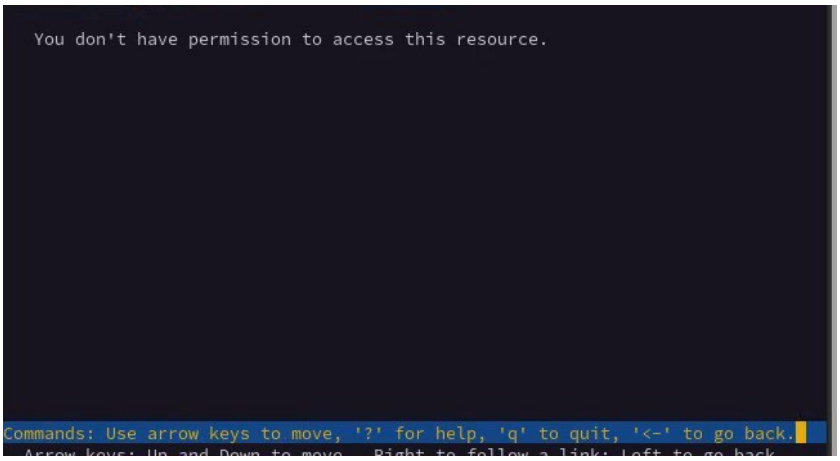

Рис. 2.16: Попытка прослушивания другого порта

Проанализировала лог-файлы: tail -nl /var/log/messages

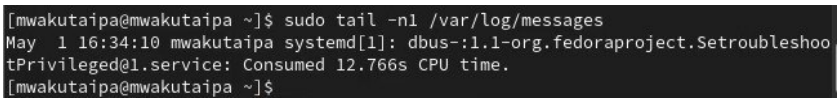

Рис. 2.17: Проверка лог-файлов

Запись появилась в файлу error_log.

Рис. 2.18: Проверка лог-файлов

Выполняю команду semanage port -a -t http_port_t -p tcp 81. После этого проверяю список портов командой semanage port -l | grep http_port_t. Порт 81 появился в списке

Рис. 2.19: Проверка портов

Перезапускаю сервер Apache

Рис. 2.20: Перезапуск сервера

Теперь он работает, ведь мы внесли порт 81 в список портов htttpd_port_t. Возвращаю в файле /etc/httpd/httpd.conf порт 80, вместо 81. Проверяю, что порт 81 удален, это правда.

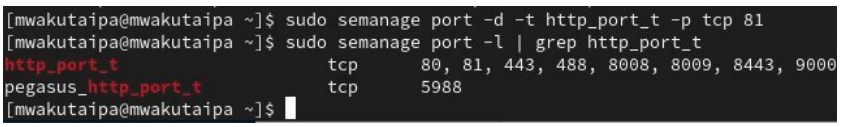

Рис. 2.21: Проверка порта 81

Далее удаляю файл test.html, проверяю, что он удален

Рис. 2.22: Удаление файла

3 Выводы

При выполнении данной лабораторной работы были развиты навыки администрирования ОС Linux, получено первое практическое знакомство с технологией SELinux и проверена работа SELinux на практике совместно с веб-сервером Apache.

Список литературы