Отчет по лабораторной работе №6

-

Овениязов Артур

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 Задание

Лабораторная работа подразумевает практическое исследование дискреционных разграничений в современных системах с открытым кодом на базе ОС Linux, а именно изучение атрибутов для групп пользователей.

# 3 Теоретическое введение

SELinux (SELinux) — это система принудительного контроля доступа, реализованная на уровне ядра. Впервые эта система появилась в четвертой версии CentOS, а в 5 и 6 версии реализация была существенно дополнена и улучшена. Эти улучшения позволили SELinux стать универсальной системой, способной эффективно решать массу актуальных задач. Стоит помнить, что классическая система прав Unix применяется первой, и управление перейдет к SELinux только в том случае, если эта первичная проверка будет успешно пройдена. [1]

# 4 Выполнение лабораторной работы

## 4.1 ВАЖНО!!! Пунктов в отчете было бы необычайного много если отвечать скрином на каждый, поэтому многие пункты были объеденены для более легкого составления отчета .

1-8. Войдите в систему с полученными учётными данными и убедитесь, что SELinux работает в режиме enforcing политики targeted с помощью команд getenforce и sestatus. Обратитесь с помощью браузера к веб-серверу, запущенному на вашем компьютере, и убедитесь, что последний работает: service httpd status или /etc/rc.d/init.d/httpd status Если не работает, запустите его так же, но с параметром start. Найдите веб-сервер Apache в списке процессов, определите его контекст безопасности и занесите эту информацию в отчёт. Например, можно использовать команду ps auxZ | grep httpd или ps -eZ | grep httpd Посмотрите текущее состояние переключателей SELinux для Apache с помощью команды sestatus -bigrep httpd Обратите внимание, что многие из них находятся в положении «off». Посмотрите статистику по политике с помощью команды seinfo, также определите множество пользователей, ролей, типов. Определите тип файлов и поддиректорий, находящихся в директории /var/www, с помощью команды ls -lZ /var/www Определите тип файлов, находящихся в директории /var/www/html: ls -lZ /var/www/html Определите круг пользователей, которым разрешено создание файлов в директории /var/www/html

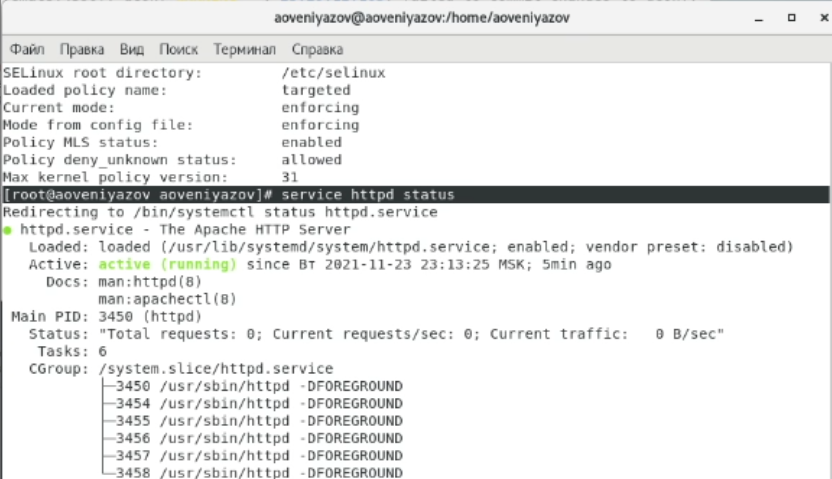


Figure 1: Результат выполнения 1-4

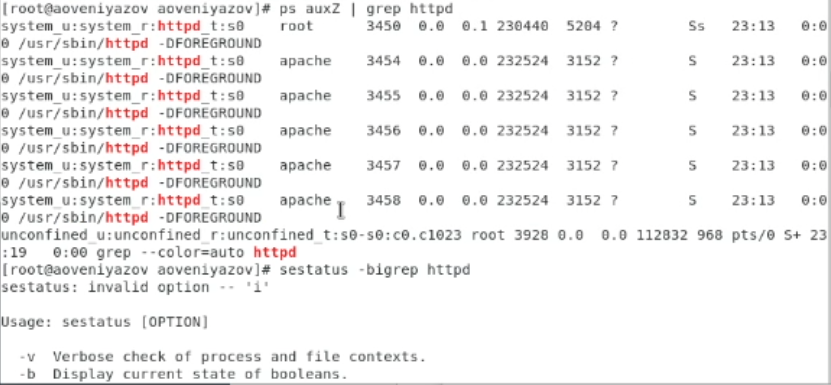


Figure 2: Результат выполнения 5-6

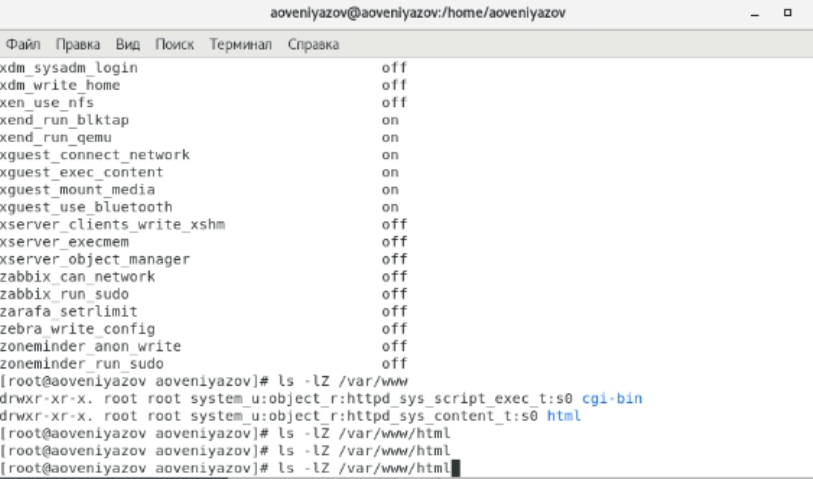


Figure 3: Результат выполнения 7-8

8-12. Создайте от имени суперпользователя (так как в дистрибутиве после установки только ему разрешена запись в директорию) html-файл /var/www/html/test.html следующего содержания:

test

Проверьте контекст созданного вами файла. Занесите в отчёт контекст, присваиваемый по умолчанию вновь созданным файлам в директории /var/www/html. Обратитесь к файлу через веб-сервер, введя в браузере адрес http://127.0.0.1/test.html. Убедитесь, что файл был успешно отображён. Изучите справку man httpd\_selinux и выясните, какие контексты файлов определены для httpd. Сопоставьте их с типом файла test.html. Проверить контекст файла можно командой ls -Z. ls -Z /var/www/html/test.html

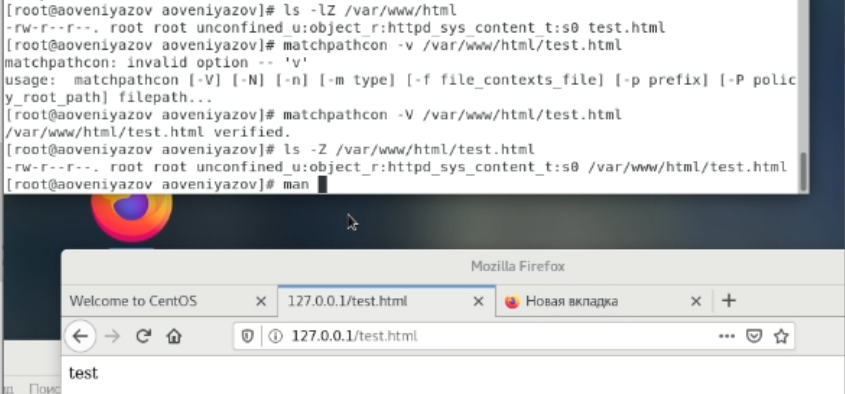


Figure 4: Результат выполнения 8-12

13-24.Измените контекст файла /var/www/html/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 После этого проверьте, что контекст поменялся. 14. Попробуйте ещё раз получить доступ к файлу через веб-сервер, введя в браузере адрес http://127.0.0.1/test.html. Вы должны получить сообщение об ошибке: Forbidden You don’t have permission to access /test.html on this server.

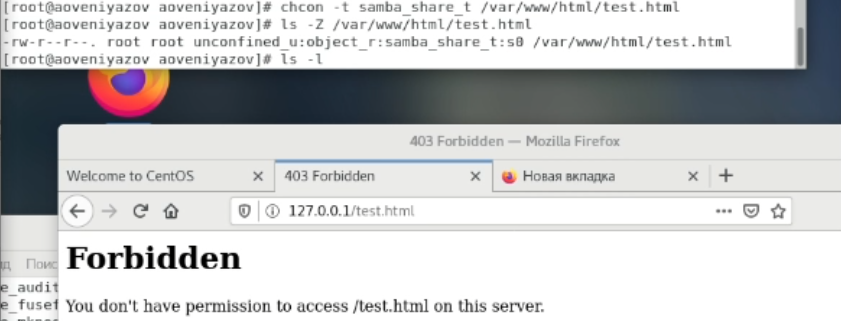


Figure 5: Результат выполнения 13-14

Просмотрите log-файлы веб-сервера Apache. Также просмотрите 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.

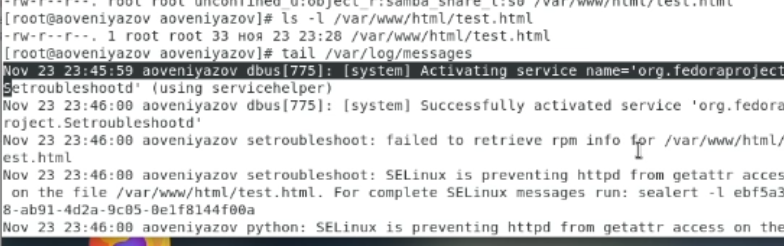


Figure 6: Результат выполнения 15

1. Попробуйте запустить веб-сервер Apache на прослушивание ТСР-порта 81 (а не 80, как рекомендует IANA и прописано в /etc/services). Для этого в файле /etc/httpd/httpd.conf найдите строчку Listen 80 и замените её на Listen 81.

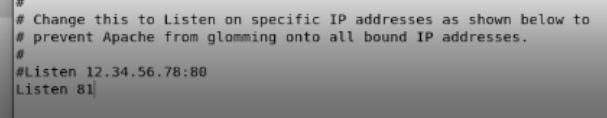


Figure 7: Результат выполнения 16

1. Выполните команду semanage port -a -t http\_port\_t -р tcp 81 После этого проверьте список портов командой semanage port -l | grep http\_port\_t Убедитесь, что порт 81 появился в списке.



Figure 8: Результат выполнения 19

1. Верните контекст httpd\_sys\_cоntent\_\_t к файлу /var/www/html/ test.html: chcon -t httpd\_sys\_content\_t /var/www/html/test.html
2. Исправьте обратно конфигурационный файл apache, вернув Listen 80.
3. Удалите привязку http\_port\_t к 81 порту: semanage port -d -t http\_port\_t -p tcp 81 и проверьте, что порт 81 удалён.
4. Удалите файл /var/www/html/test.html: rm /var/www/html/test.html

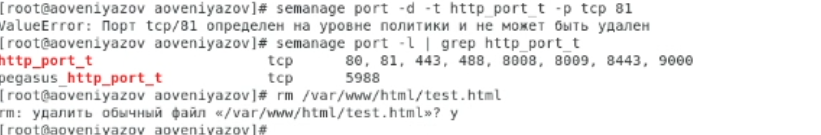


Figure 9: Результат выполнения 21-24

# 5 Выводы

Сегодня я приобрел практические навыки работы в консоли с атрибутами файлов, закрепил теоретические основы дискреционного разграничения доступа в совытым кодом на базе ОС Linux и проверил работу SELinux на практике совместно с веб-сервером Apache

# Список литературы

1. Дискреционное разграничение доступа Linux [Электронный ресурс]. Сайт, 2021. URL: <http://debianinstall.ru/diskretsionnoe-razgranichenie-dostupa-linux/>.