Лабораторнаяработа № 6

Мандатное разграничение прав в Linux

Усов Александр Александрович НБибд-02-18

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

1. Запустить серверо
2. Настроить права дотупа к файлам
3. Поменять порт сервера

# 3 Теоретическое введение

Security Enhanced Linux может работать двумя различными способами:

Enforcing: SELinux запрещает доступ на основе правил политики SELinux, набора руководящих принципов, которые управляют механизмом безопасности. Permissive:SELinux не запрещает доступ, но в журнале регистрируются отказы для действий, которые были бы запрещены при запуске в принудительном режиме. SELinux также можно отключить.

Хотя это не сам режим работы, это все же вариант.

Однако научиться использовать этот инструмент лучше, чем просто игнорировать его. Имейте это в виду!

Чтобы отобразить текущий режим SELinux, используйте getenforce.

Если вы хотите переключить режим работы, используйте setenforce 0 (чтобы установить для него Permissive) или setenforce 1 (Enforcing).

Поскольку это изменение не выдержит перезагрузки, вам потребуется отредактировать файл /etc/selinux/config и установить для переменной SELINUX значениеenforcing, permissive, или disabled, чтобы обеспечить постоянство при перезагрузках:

Напомним, что если Getenforce возвращает Disabled, вам нужно отредактировать /etc/selinux/config с нужным режимом работы и перезагрузиться.

В противном случае вы не сможете установить (или переключить) режим работы с помощью setenforce.

Одно из типичных применений setenforce состоит в переключении между режимами SELinux (от принудительного к разрешающему или наоборот) для устранения неполадок в работе приложения, которое работает неправильно или работает не так, как ожидалось.

Если оно работает после того, как вы установили SELinux в режим Permissive, вы можете быть уверены, что надо рыться в проблеме с разрешениями SELinux.

Два классических случая, когда нам, скорее всего, придется иметь дело с SELinux:

Изменение порта по умолчанию, на котором слушает демон. Установка директивы DocumentRoot для виртуального хоста вне /var/www/html. [1]

# 4 Выполнение лабораторной работы

1. Предварительно настроил систему, установил необходимые утилиты.
2. Войдем в систему с полученными учётными данными и убедимся, что SELinux работает в режиме enforcing политики targeted с помощью команд getenforce и sestatus (рис. 1):

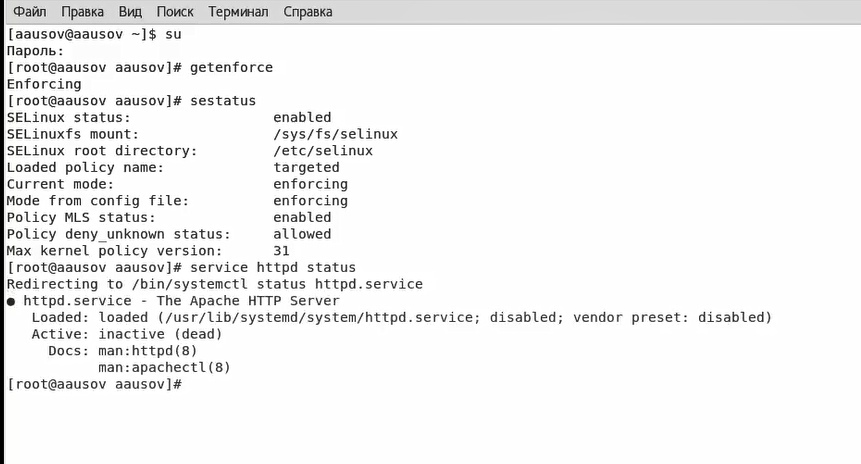


Figure 1: getenforce и sestatus

1. Обратимся с помощью браузера к веб-серверу, запущенному на компьютере, и убедимся, что последний работает: service httpd status

(рис. 2).

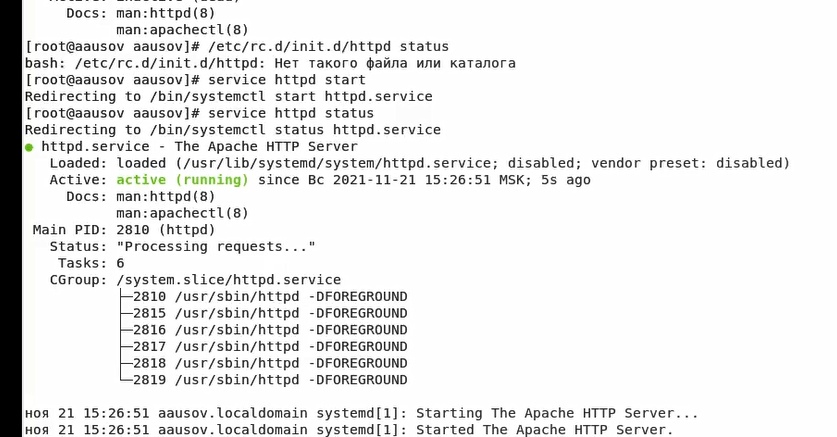


Figure 2: service httpd status

1. Найдем веб-сервер Apache в списке процессов, определим его контекст безопасности. Например, можно использовать команду ps auxZ | grep httpd или ps -eZ | grep httpd

(рис. 3).

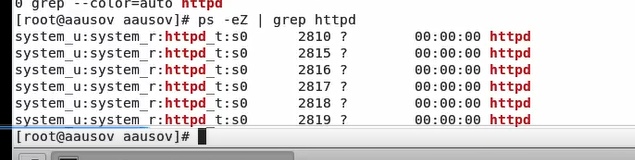


Figure 3: ps -eZ | grep httpd

1. Посмотрим текущее состояние переключателей SELinux для Apache с помощью команды sestatus –b | grep httpd, обращая внимание, что многие из них находятся в положении «off».

(рис. 4).

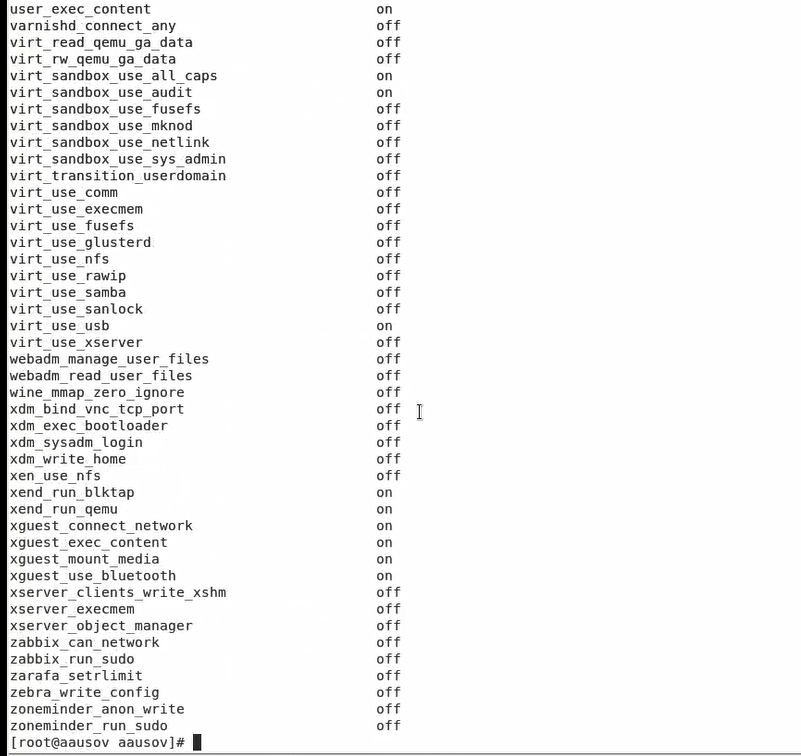


Figure 4: sestatus –b | grep httpd

1. Посмотрим статистику по политике с помощью команды seinfo, также определим множество пользователей, ролей, типов. (рис. 5).

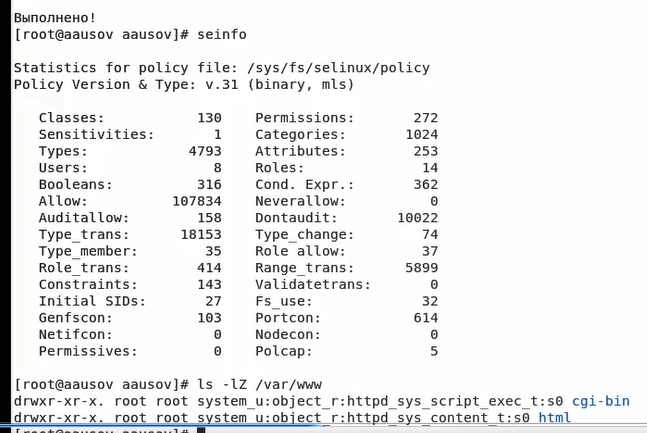


Figure 5: seinfo

1. СОпределим тип файлов и поддиректорий, находящихся в директории /var/www, с помощью команды ls -lZ /var/www Определим тип файлов, находящихся в директории /var/www/html: ls -lZ /var/www/html (рис. 5)
2. Определим круг пользователей, которым разрешено создание файлов в директории /var/www/html. Создание файлов в директории разрешено только суперпользователю.Создадим 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

(рис. 6)

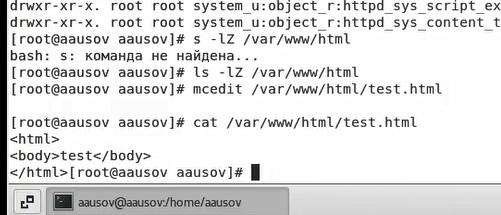


Figure 6: html-файл /var/www/html/test.html

1. Обратимся к файлу через веб-сервер. Убедимся, что файл был успешно отображён.(рис. 7)

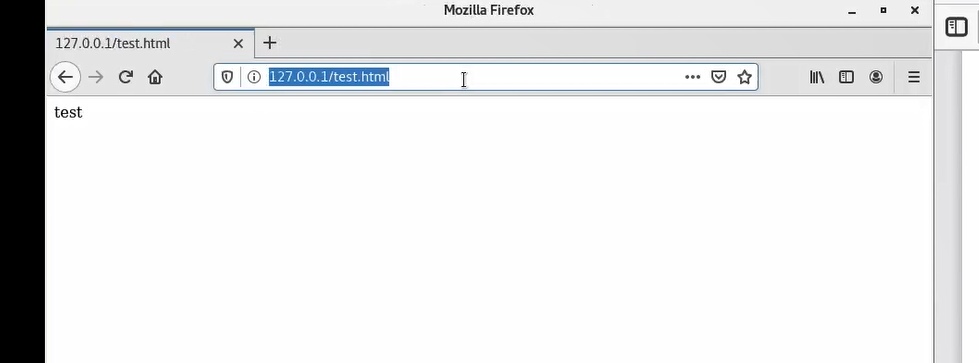


Figure 7: Обратимся к файлу через веб-сервер

1. Изучим справку man httpd\_selinux и выясним, какие контексты файлов определены для httpd. Сопоставим их с типом файла test.html. Проверить контекст файла можно командой ls -Z. ls -Z /var/www/html/test.html SELinux требует наличия у файлов расширенных атрибутов, определяющих тип файла. Политика управляет видом доступа демона к этим файлам. Политика SELinux для демона httpd позволяет пользователям настроить web-службы максимально безопасным методом с высокой степенью гибкости. рис. 8)

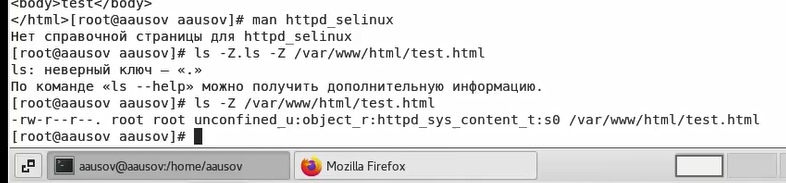


Figure 8: ls -Z /var/www/html/test.html

1. Попробуем ещё раз получить доступ к файлу через веб-сервер, введя в браузере адрес http://127.0.0.1/test.html. Однако, мы получаем сообщение об ошибке: (рис. 9)

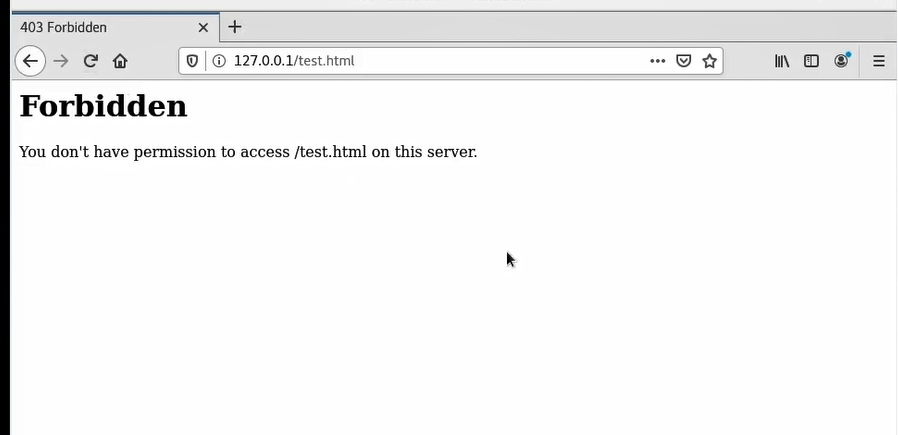


Figure 9: сообщение об ошибке

1. Проанализируем ситуацию. Почему файл не был отображён, если права доступа позволяют читать этот файл любому пользователю? Из-за смены контекста доступ к файлу из браузера запрещен. SELinux требует наличия у файлов расширенных атрибутов, определяющих тип файла. ls -l /var/www/html/test.html Просмотрим log-файлы веб-сервера Apache. Также просмотрим системный лог-файл: tail /var/log/messages Если в системе окажутся запущенными процессы setroubleshootd и audtd, то мы также сможем увидеть ошибки, аналогичные указанным выше, в файле /var/log/audit/audit.log.
2. Попробуем 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 найдем строчку Listen 80 и заменим её на Listen 81. (рис. 10)

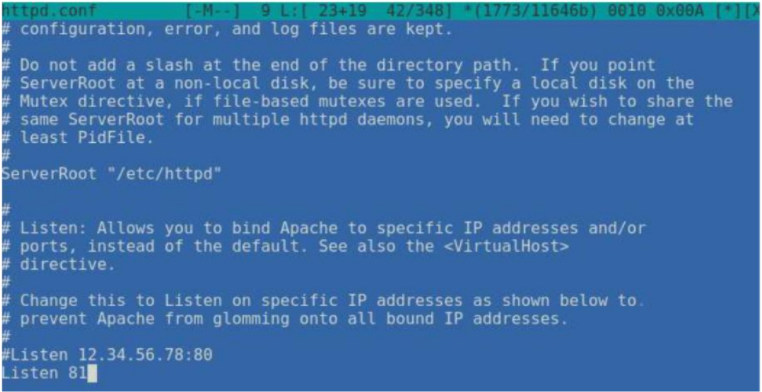


Figure 10: найдем строчку Listen 80 и заменим её на Listen 81.

1. Выполним перезапуск веб-сервера Apache и проанализируем лог-файлы: tail –nl /var/log/messages. Просмотрим файлы и выясним, в каких файлах появились записи: /var/log/http/error\_log /var/log/http/access\_log /var/log/audit/audit.log

Перезапуск прошел успешно, т.к. 81 порт в данной системе прописан в политике. Ошибок не выскочило, только записи о перезапуске.

1. Выполним команду semanage port -a -t http\_port\_t -р tcp 81 После этого проверим список портов командой semanage port -l | grep http\_port\_t Убедимся, что порт 81 появился в списке. Команда ни на что не повлияла, порт там уже был. В списке есть. (рис. 11)

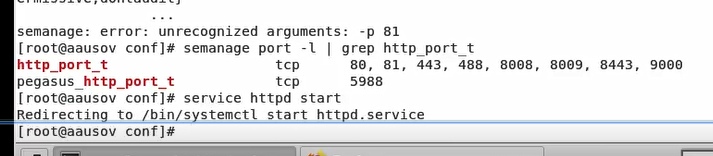


Figure 11: Убедимся, что порт 81 появился в списке

1. Попробуем запустить веб-сервер Apache ещё раз. Поняли ли вы, почему он сейчас запустился, а в предыдущем случае не смог? Успешно перезапустили. И тогда, и сейчас это было успешно, т.к. порт 81 уже был прописан.
2. Вернем контекст httpd\_sys\_cоntent\_\_t к файлу /var/www/html/ test.html: chcon -t httpd\_sys\_content\_t /var/www/html/test.html После этого попробуем получить доступ к файлу через веб-сервер, введя в браузере адрес http://127.0.0.1:81/test.html. Мы видим содержимое файла – слово «test». (рис. 12)

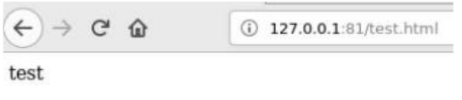


Figure 12: Убедимся, что порт 81 появился в списке

1. Исправим обратно конфигурационный файл apache, вернув Listen 80.
2. Удалим привязку http\_port\_t к 81 порту: semanage port -d -t http\_port\_t -p tcp 81
3. Удалим файл /var/www/html/test.html: rm /var/www/html/test.html

# 5 Выводы

В ходе данной лабораторной работы мы развили навыки администрирования ОС Linux, впервые практически познакомились с технологией SELinux1 и проверили работу SELinux на практике совместно с веб-сервером Apache.

# Список литературы

1. suid [Электронный ресурс]. Сайт, 2021. URL: <https://itsecforu.ru/2019/07/25/%F0%9F%9B%A1%EF%B8%8F-%D1%80%D0%B5%D0%B0%D0%BB%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F-%D0%BC%D0%B0%D0%BD%D0%B4%D0%B0%D1%82%D0%BD%D0%BE%D0%B3%D0%BE-%D0%BA%D0%BE%D0%BD%D1%82%D1%80%D0%BE%D0%BB%D1%8F-%D0%B4/>.