Лабораторная работа №6

Мандатное разграничение прав в Linux.

Галиева Аделина Руслановна

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 Выполнение лабораторной работы

1. Устанавливаем httpd.
2. Задаем имя сервера.
3. Открываем порты для работы с протоколом httpd.

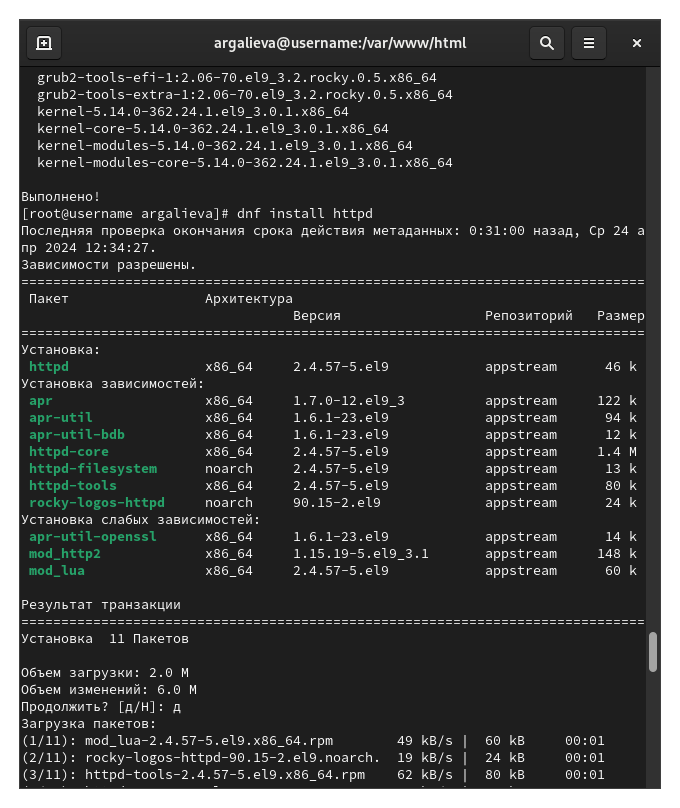


Рис. 1: Запуск httpd



Рис. 2: Запуск httpd

1. Входим в систему с полученными учётными данными и убеждаемся, что SELinux работает в режиме enforcing политики targeted с помощью команд getenforce и sestatus.
2. Обращаемся с помощью браузера к веб-серверу, запущенному на вашем компьютере, и убеждаемся, что последний работает: service httpd status или /etc/rc.d/init.d/httpd status.
3. Находим веб-сервер Apache в списке процессов, определяем его контекст безопасности и заносим эту информацию в отчёт. Используем команду ps auxZ | grep httpd.

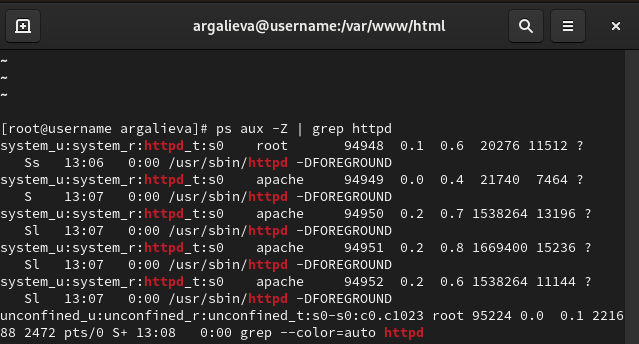


Рис. 3: Контекст безопасности

1. Смотрим текущее состояние переключателей SELinux для Apache с помощью команды sestatus -bigrep httpd. Обращаем внимание, что многие из них находятся в положении «off»

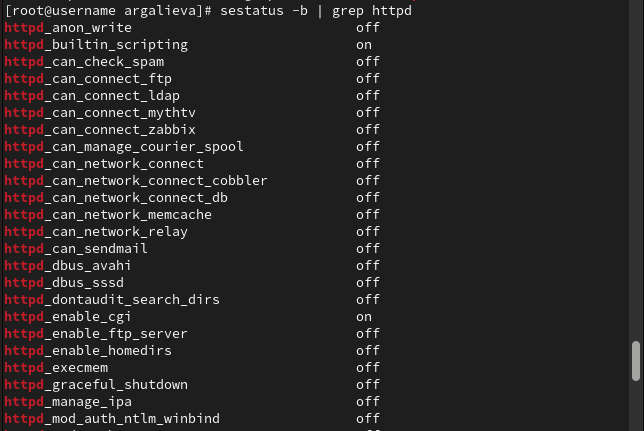


Рис. 4: Переключатели SELinux для httpd

1. Смотрим статистику по политике с помощью команды seinfo, также определяем множество пользователей, ролей, типов.
2. Определяем тип файлов и поддиректорий, находящихся в директории /var/www, с помощью команды ls -lZ /var/www. В поддиректориях могут располагаться системные скрипты и контент для httpd.
3. Определяем тип файлов, находящихся в директории /var/www/html: ls -lZ /var/www/html. В директории изначально нет файлов.
4. Определяем круг пользователей, которым разрешено создание файлов в директории /var/www/html. Создаем файлы может только root.
5. Создаем от имени суперпользователя (так как в дистрибутиве после установки только ему разрешена запись директорию) html-файл /var/www/html/test.html следующего содержания:

* test

1. Проверяем контекст созданного вами файла. Заносим в отчёт контекст, присваиваемый по умолчанию вновь созданным файлам в директории /var/www/html.
2. Обращаемся к файлу через веб-сервер, введя в браузере адрес http://127.0.0.1/test.html. Убеждаемся, что файл был успешно отображён.

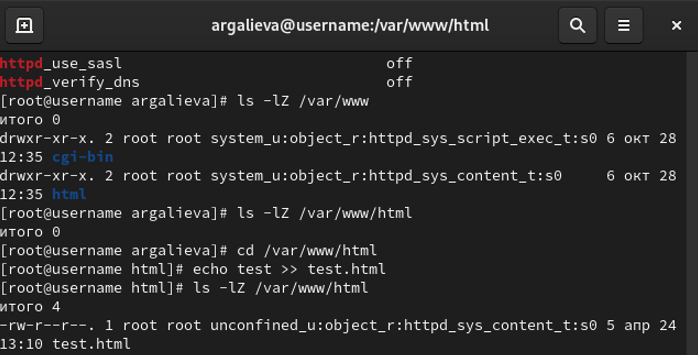


Рис. 5: Создание html-файла и доступ по httpd

1. Изучаем справку man httpd\_selinux и выясняем, какие контексты файлов определены для httpd. Сопоставляем их с типом файла test.html. Проверяем контекст файла можно командой ls -Z. ls -Z /var/www/html/test.html. Основным контекстом является httpd\_sys\_content\_t, его мы и увидим в выводе команды.
2. Изменяем 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 проверяем, что контекст поменялся.
3. Пробуем ещё раз получить доступ к файлу через веб-сервер, введя в браузере адрес http://127.0.0.1/test.html. Мы должны получить сообщение об ошибке: Forbidden You don’t have permission to access /test.html on this server. При изменении контекста файл стал считаться чужим для http и программа не может его прочитать.

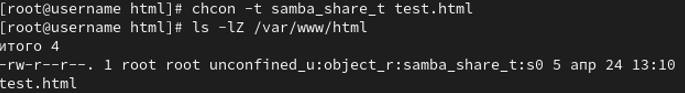


Рис. 6: Ошибка доступа после изменения контекста

1. Проанализируем ситуацию. Почему файл не был отображён, если права доступа позволяют читать этот файл любому пользователю? ls -l /var/www/html/test.html. Просматриваем log-файлы веб-сервера Apache. Также просмотриваем 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. Проверяем это утверждение самостоятельно.

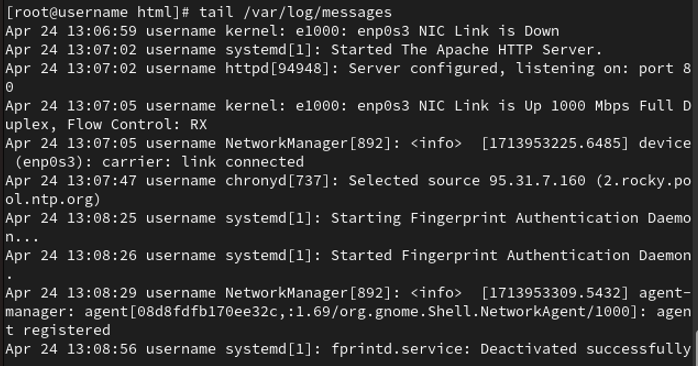


Рис. 7: Лог ошибок

1. Пробуем 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 находим строчку Listen 80 и заменяем её на Listen 81.

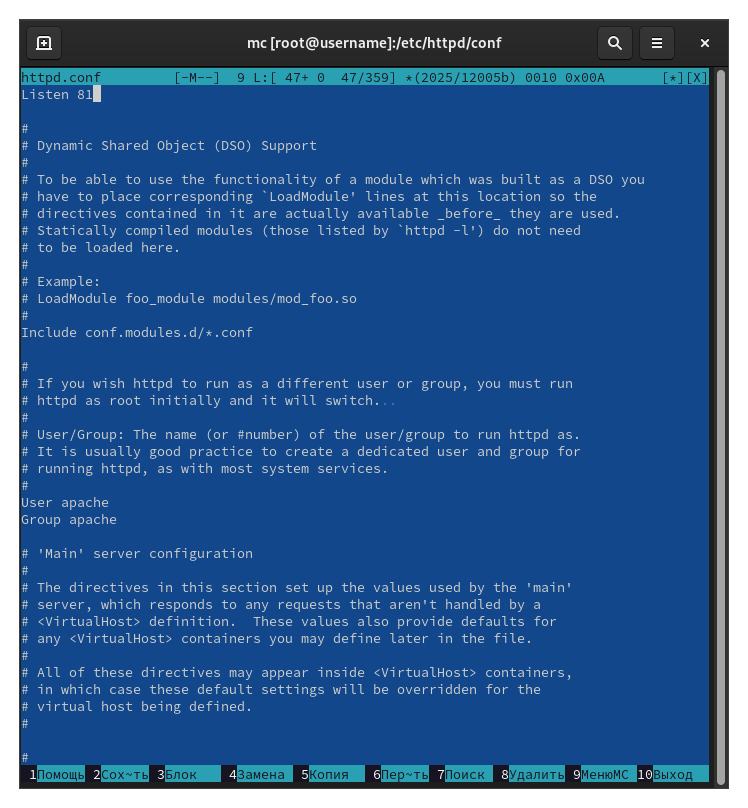


Рис. 8: Переключение порта

1. Выполняем перезапуск веб-сервера Apache. Произошёл сбой? Сбой не происходит, порт 81 уже вписан в разрешенные.
2. Проанализируем лог-файлы: tail -nl /var/log/messages. Просматриваем файлы /var/log/http/error\_log, /var/log/http/access\_log и /var/log/audit/audit.log и выясняем, в каких файлах появились записи.
3. Выполняем команду semanage port -a -t http\_port\_t -р tcp 81. После этого проверяем список портов командой semanage port -l | grep http\_port\_t. Убеждаемся, что порт 81 появился в списке.
4. Пробуем запустить веб-сервер Apache ещё раз.
5. Возвращаем 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 пробуем 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».

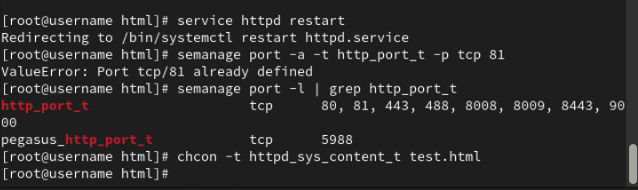


Рис. 9: Доступ по http на 81 порт

1. Исправляем обратно конфигурационный файл apache, вернув Listen 80.
2. Удаляем привязку http\_port\_t к 81 порту: semanage port -d -t http\_port\_t -p tcp 81 и проверяем, что порт 81 удалён.
3. Удаляем файл /var/www/html/test.html: rm /var/www/html/test.html.

# 3 Выводы

В ходе выполнения лабораторной работы я развила навыки администрирования ОС Linux. Получила первое практическое знакомство с технологией SELinux. Проверила работу SELinx на практике совместно с веб-сервером Apache.