Отчёт по лабораторной работе №6

Мандатное разграничение прав в Linux

Косолапов Степан Эдуардович НПИбд-01-20

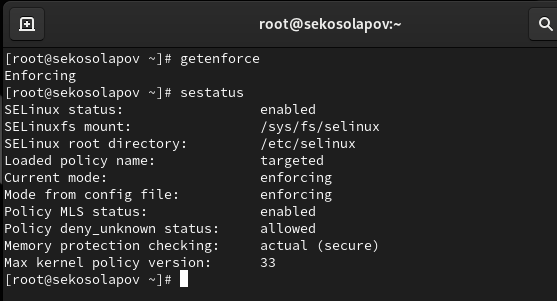
Содержание

# 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.

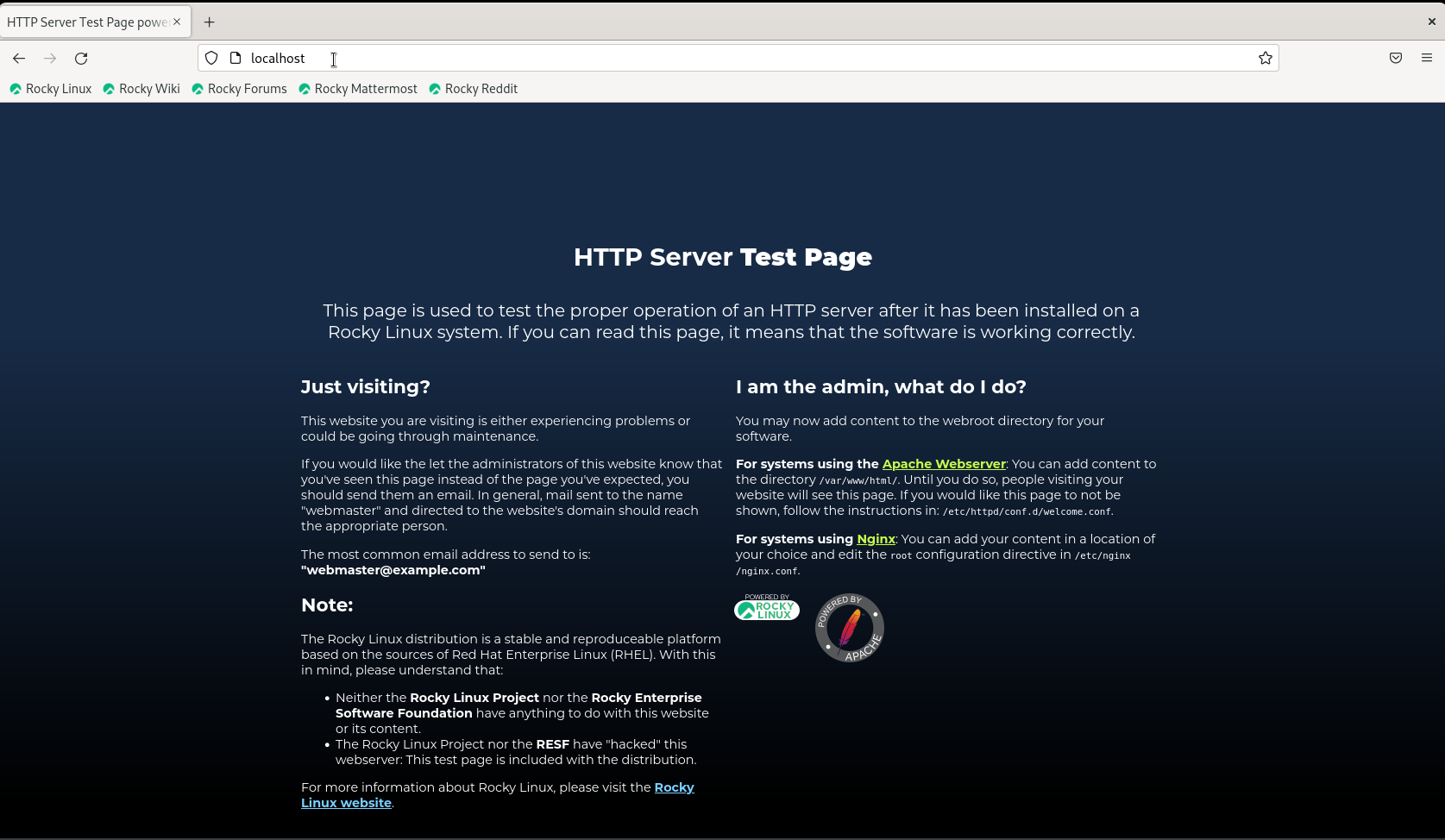
# 2 Выполнение работы

1. Войдите в систему с полученными учётными данными и убедитесь, что SELinux работает в режиме enforcing политики targeted с помощью команд getenforce и sestatus.



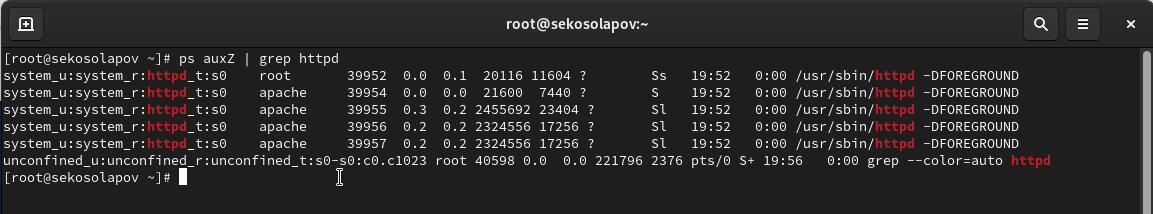
getenforce и sestatus

1. Обращаемся с помощью браузера к веб-серверу, запущенному на компьютере, и видим, что сервер работает



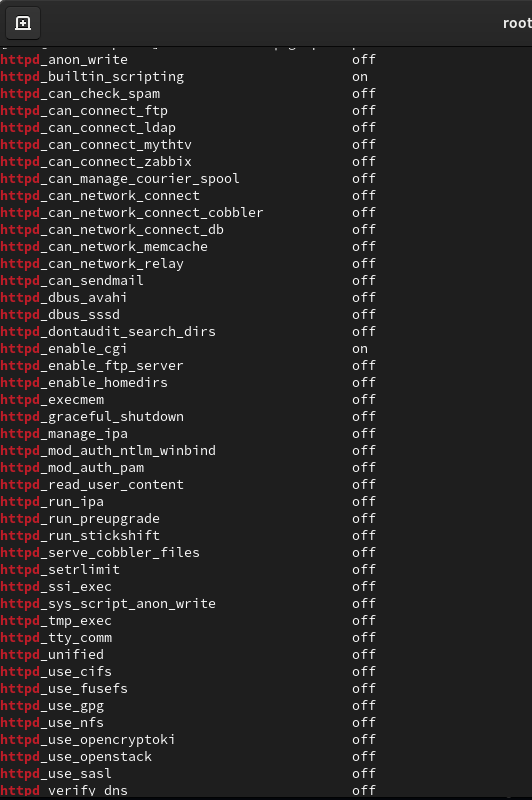
тестовая страница apache

1. Находим веб-сервер Apache в списке процессов, определяем его контекст безопасности - system\_u:system\_r:httpd\_t:s0



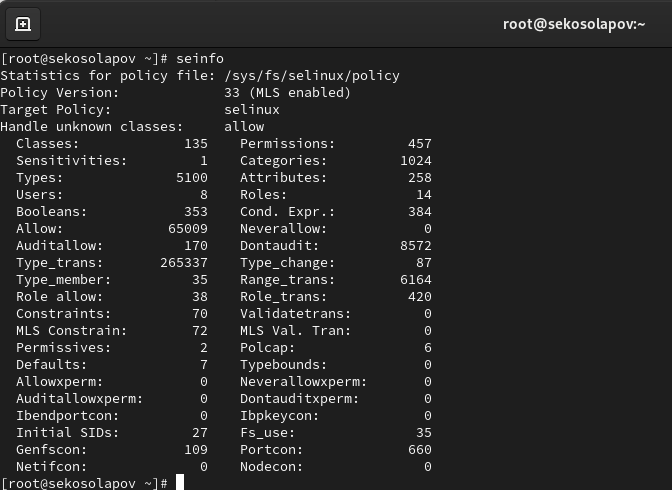
контекст безопасности процессов httpd

1. Смотрим текущее состояние переключателей SELinux для Apache с помощью команды sestatus -b | grep httpd



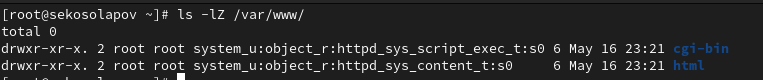
переключатели selinux для httpd

1. Смотрим статистику по политике с помощью команды seinfo. Количесвто пользователей - 8, ролей - 14, типов - 5100.



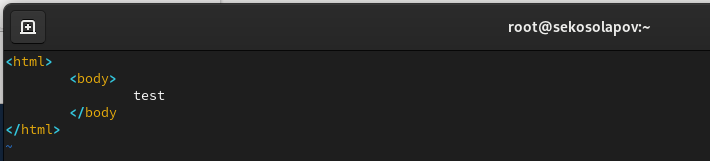
статистика по политике

1. Определяем тип файлов и поддиректорий, находящихся в директории /var/www, с помощью команды ls -lZ /var/www - это типы httpd\_sys\_script\_exec\_t для cgi-bin b httpd\_sys\_content\_t для html
2. Файлов в директории /var/www/html пока что нет.
3. Определяем круг пользователей, которым разрешено создание файлов в директории /var/www/html с помощью той же команды ls, видим, что доступ на создание есть только у root.



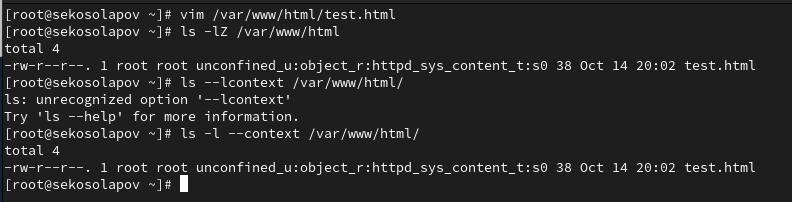
данные о содержимом /var/www/

1. Создаём от имени суперпользователя html-файл /var/www/html/test.html



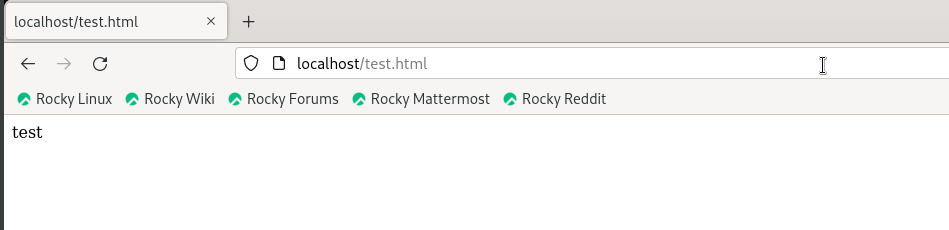
test.html

1. Проверяем контекст созданного нами файла - unconfined\_u:object\_r:httpd\_sys\_content\_t:s0 - это контекст, присваиваемый по умолчанию вновь созданным файлам в директории /var/www/html.



контекст test.html

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



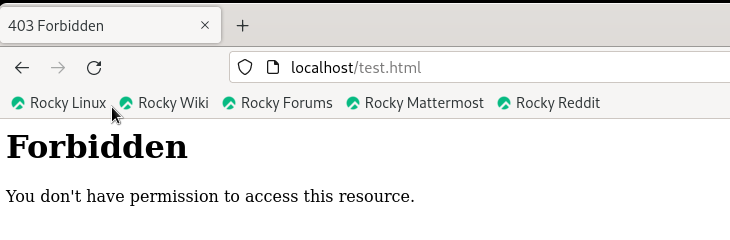
test.html в браузере

1. Изучить справку man httpd\_selinux у нас не получилось, т.к такой справки не было обнаружено, поэтому обратились в интернет за этой справкой.
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



смена контекста

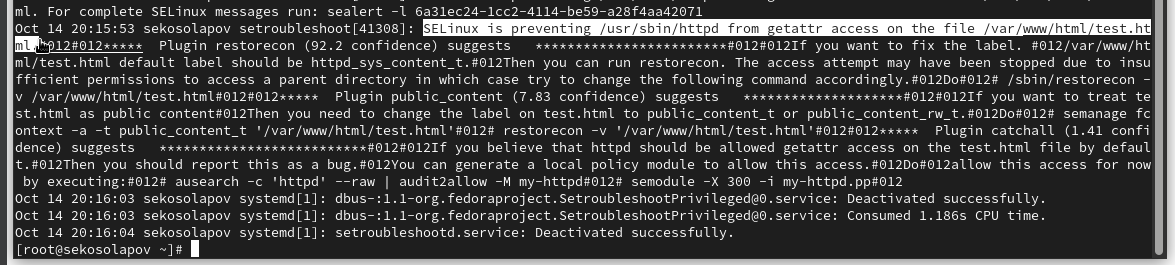
1. Пробуем ещё раз получить доступ к файлу через веб-сервер, введя в браузере адрес http://127.0.0.1/test.html. Мы получаем сообщение об ошибке: Forbidden You don’t have permission to access this resource.



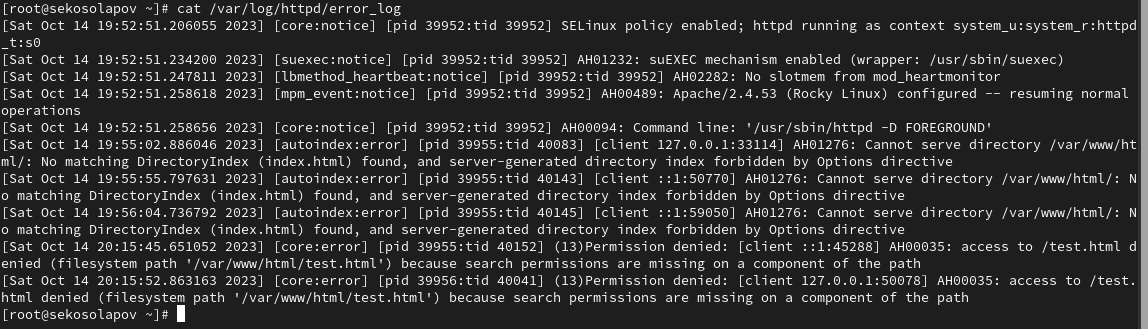
ошибка при попытке доступа к файлу с другим контекстом

1. В данной ситуации у сервиса httpd не было доступа до файла из-за изменённого контекста.

Смотрим логи и видим, что действительно не было доступа у самого процесса до файла test.html

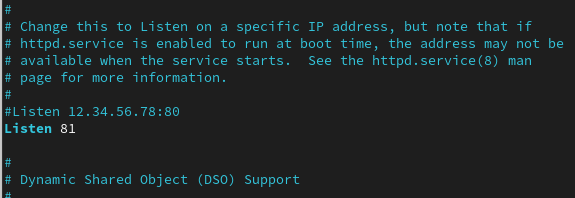


системные логи



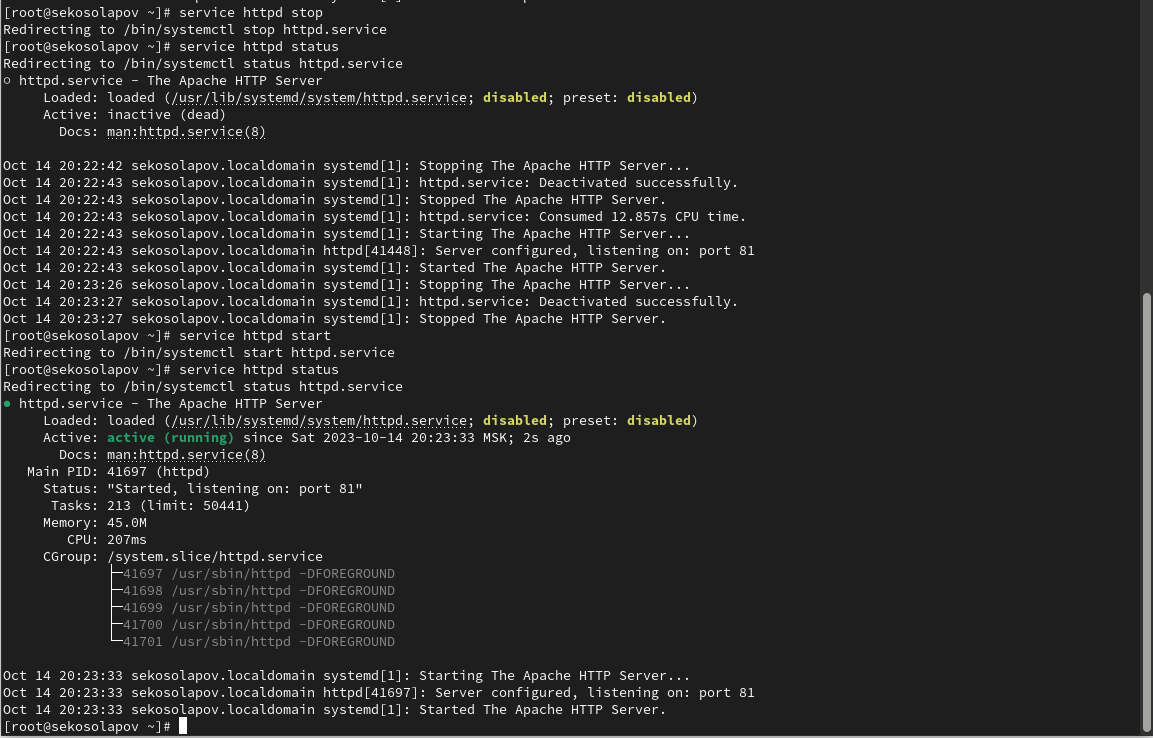
логи apache

1. Пробуем запустить веб-сервер Apache на прослушивание ТСР-порта 81. Для этого в файле /etc/httpd/conf/httpd.conf находим строчку Listen 80 и заменяем её на Listen 81.



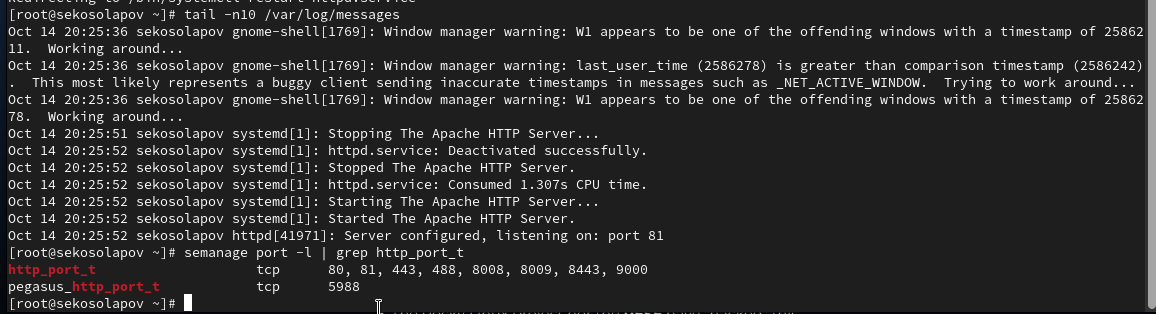
установка listen 81

1. Выполняем перезапуск веб-сервера Apache. Сбоя не произошло, т.к у нас уже был добавлен порт 81 в http\_port\_t.



перезапуск httpd с портом 81

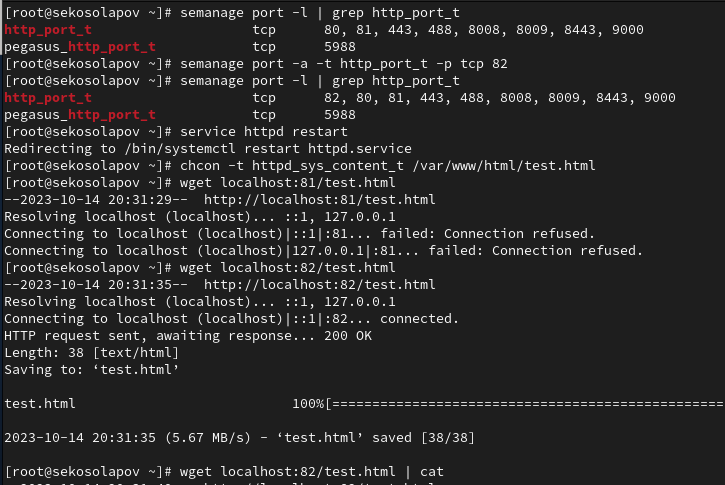
1. Анализируем лог-файлы: tail -nl /var/log/messages, и видим, что проблем не возникло. После просмотра semanage port -l | grep http\_port\_t, замечаем, что 81 порт уже добавлен.



системные логи при прослушивании 81 порта httpd

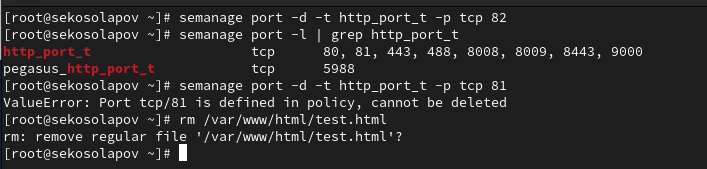
1. Пробуем провести все действия с портом 82. Добавляем его в /etc/httpd/conf/httpd.conf b и перезапускаем httpd сервис. Видим ошибку, все из-за того, то не добавлен порт в http\_port\_t. Добавляем командой semanage port -a -t http\_port\_t -p tcp 82. После этого пробуем перезапустить httpd и все проходит на этот раз без ошибок.

Возвращаем контекст файлу test.html и у нас получается доступиться теперь к этому файлу, все работает.



повторение действий для порта 82

1. Удаляем файл test.html и удаляем связь порта 82 из http\_port\_t.



удаление test.html и порта 82 из http\_port\_t

# 3 Выводы

В данной работе мы развили навыки администрирования ОС Linux. Получили первое практическое знакомство с технологией SELinux. Проверили работу SELinx на практике совместно с веб-сервером Apache.