Отчёт по лабораторной работе №6

Мандатное разграничение прав в Linux

Жижченко Валерия Викторовна

Содержание

[Цель работы 1](#_Toc88935663)

[Выполнение лабораторной работы 1](#_Toc88935664)

[Вывод 9](#_Toc88935665)

# Цель работы

Развить навыки администрирования ОС Linux. Получить первое практическое знакомство с технологией SELinux.

Проверить работу SELinux на практике совместно с веб-сервером Apache.

# Выполнение лабораторной работы

1. Вошли в систему с полученными учётными данными и убедились, что *SELinux* работает в режиме *enforcing* политики *targeted*:

$ sestatus

1. Убедились, что веб-сервер работает:

$ service httpd status

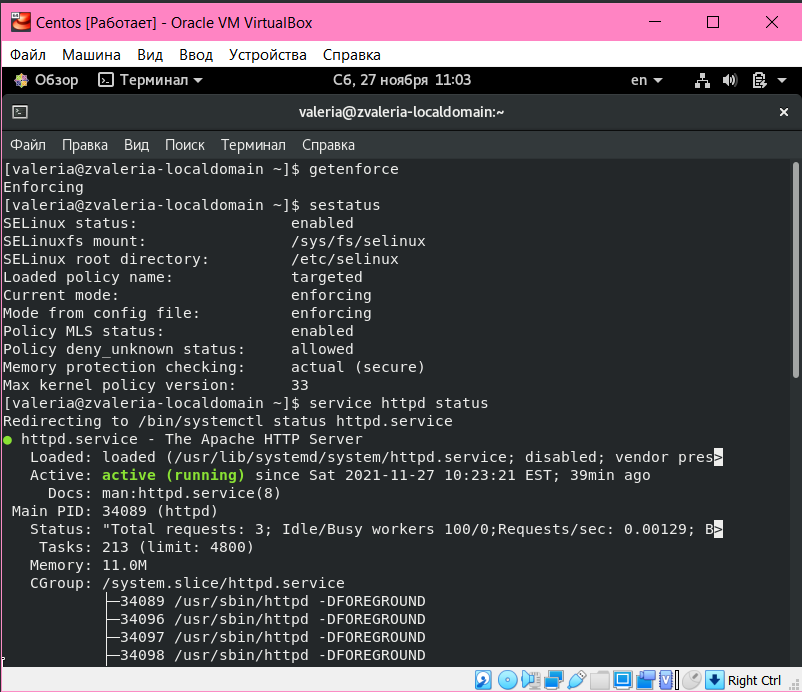


Figure 1: Выполнение пунктов 1-2

1. Нашли веб-сервер *Apache* в списке процессов, определили его контекст безопасности:

$ ps -eZ | grep httpd

1. Посмотрели текущее состояние переключателей *SELinux* для *Apache*:

$ sestatus -b | grep httpd

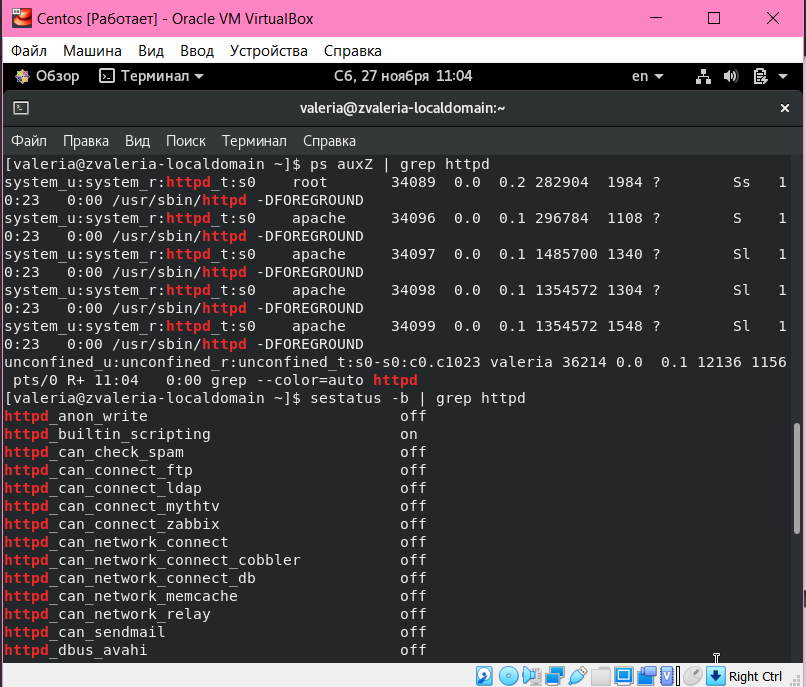


Figure 2: Выполнение пунктов 3-4

1. Посмотрели статистику по политике, также определили множество пользователей, ролей, типов:

$ seinfo

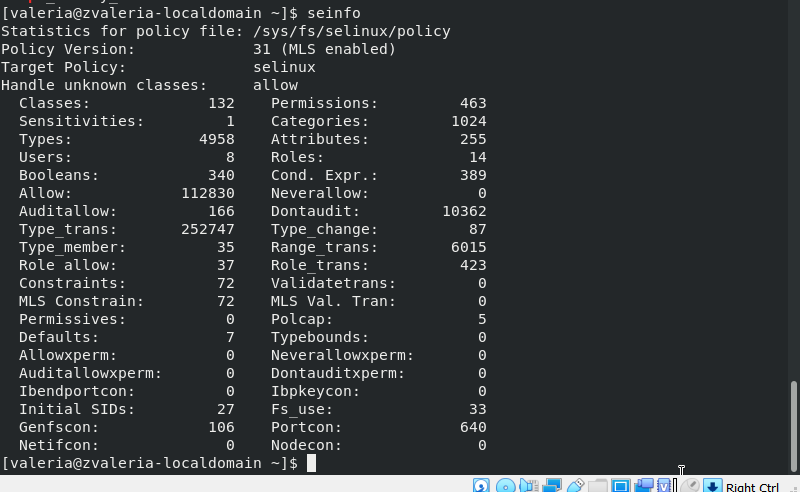


Figure 3: Вывод команды seinfo

1. Определили тип файлов и поддиректорий, находящихся в директории */var/www*:

$ ls -lZ /var/www

1. Определили тип файлов, находящихся в директории */var/www/html*:

$ ls -lZ /var/www/html

1. Определили что создание файлов в директории */var/www/html* разрешено только пользователю *root*.
2. Создали от имени суперпользователя файл */var/www/html/test.html*:

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

1. Проверили контекст созданного файла:

$ ls -Z /var/www/html/test.html

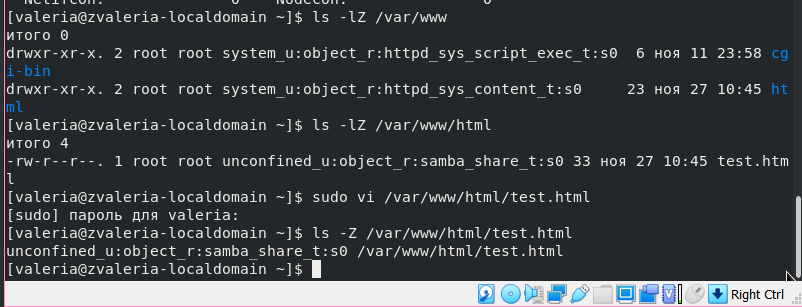


Figure 4: Выполнение пунктов 6-10

1. Обратились к файлу через веб-сервер при помощи браузера:

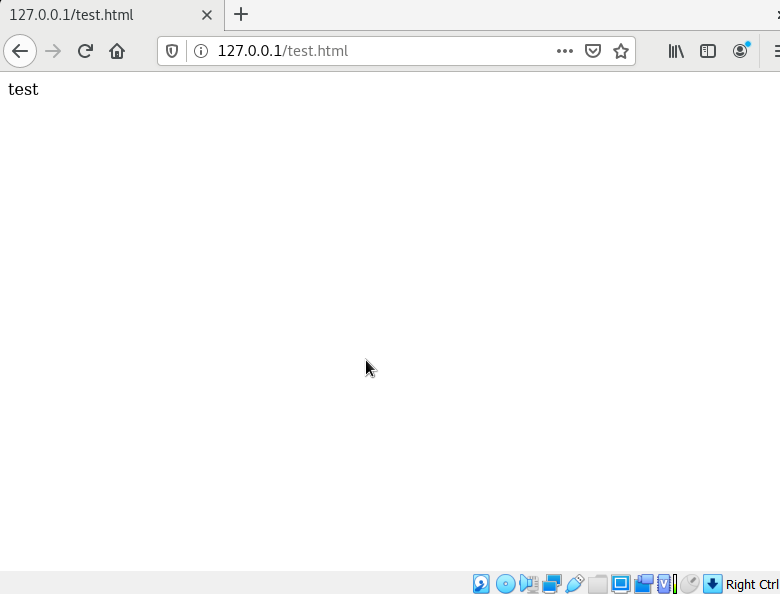


Figure 5: Файл test.html в браузере

1. Изучили справку *httpd\_selinux* и выяснили, что для файлов httpd определены контексты *httpd\_sys\_content\_t, httpd\_sys\_script\_exec\_t, httpd\_sys\_script\_ro\_t, httpd\_sys\_script\_rw\_t, httpd\_sys\_script\_ra\_t, httpd\_unconfined\_script\_exec\_t*.

Проверили контекст файла:

$ ls -Z /var/www/html/test.html

1. 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

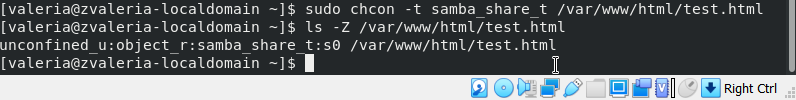


Figure 6: Выполнение пункта 13

1. Попробовали получить доступ к файлу через веб-сервер:

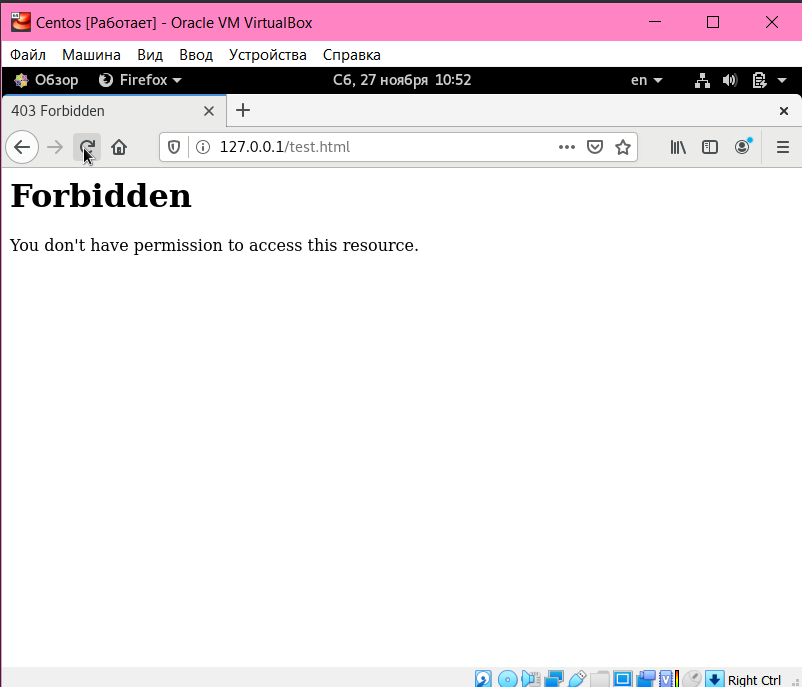


Figure 7: Попытка открыть файл test.html в браузере

1. Проанализировали почему файл не был отображён и посмотрели *log*-файлы веб-сервера *Apache*:

$ ls -l /var/www/html/test.html  
$ tail /var/log/httpd/error\_log

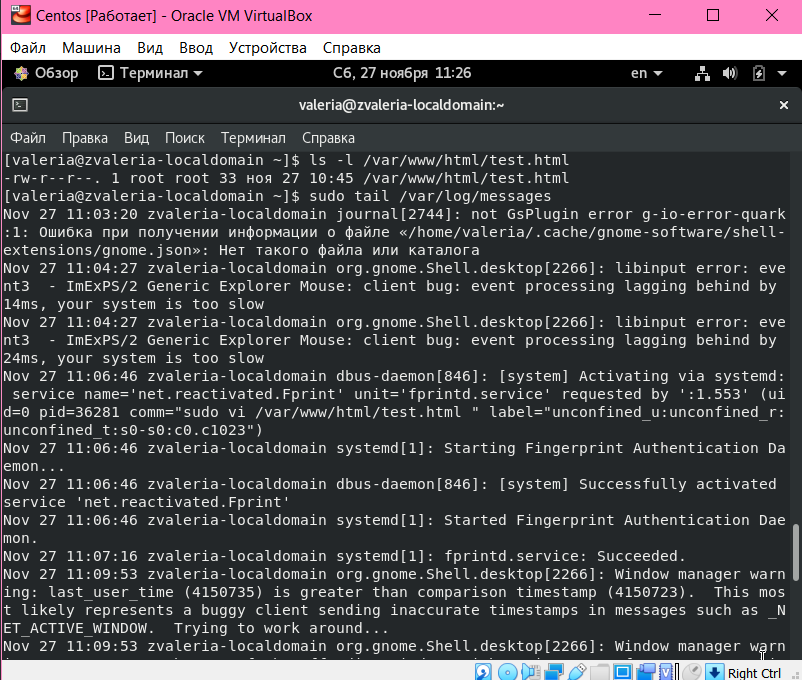


Figure 8: Выполнение пункта 15

1. Изменили конфигурацию веб-сервера *Apache*, чтобы прослушивался порт 81.
2. Выполните перезапуск веб-сервера *Apache*. Так как все завершилось успешно, переходим в пункту 21.
3. Вернули контекст *httpd\_sys\_cоntent\_t* к файлу */var/www/html/test.html* и попробовали получить доступ к файлу через веб-сервер:

$ chcon -t httpd\_sys\_content\_t /var/www/html/test.html

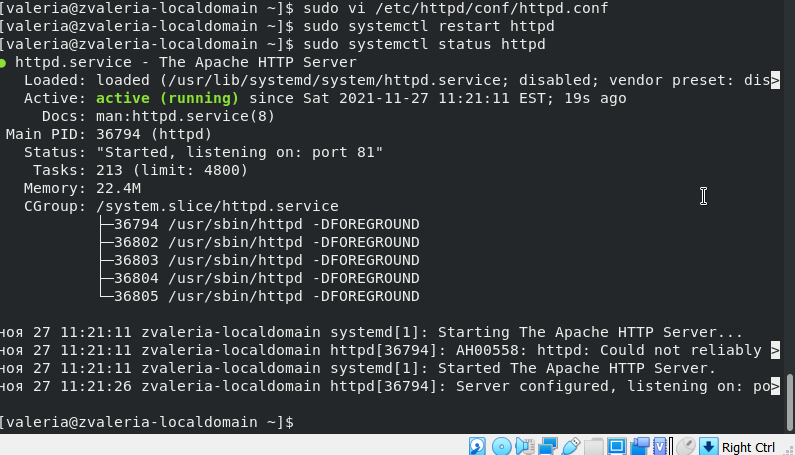


Figure 9: Выполнение пунктов 16-21

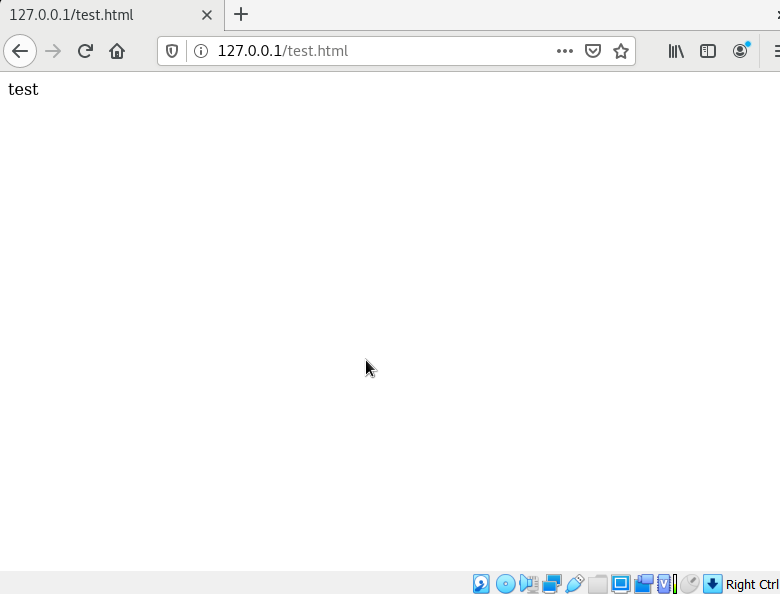


Figure 10: Файл test.html в браузере через 81 порт

1. Исправили обратно конфигурационный файл *apache*, вернув *Listen 80*.
2. Удалили файл */var/www/html/test.html*:

$ rm /var/www/html/test.html

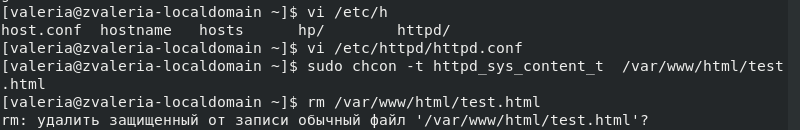


Figure 11: Файл test.html удален

# Вывод

Развили навыки администрирования ОС Linux. Получили первое практическое знакомство с технологией SELinux. Проверили работу SELinx на практике совместно с веб-сервером Apache.