Отчёт по лабораторной работе №6

Дисциплина: Информационная безопасность

Андрианова Марина Георгиевна

Содержание

[Цель работы 1](#_Toc179464785)

[Выполнение лабораторной работы 1](#_Toc179464786)

[Выводы 10](#_Toc179464787)

# Цель работы

Развить навыки администрирования ОС Linux. Получить первое практическое знакомство с технологией SELinux. Проверить работу SELinx на практике совместно с веб-сервером Apache.

# Выполнение лабораторной работы

Вошла в систему под своей учетной записью. Убедилась, что SELinux работает в режиме enforcing политики targeted с помощью команд getenforce и sestatus (рис. [-@fig:001]).

Изображение выглядит как текст, снимок экрана, Шрифт, черно-белый

Автоматически созданное описание

Проверка режима работы SELinux

Запускаю сервер apache, далее обращаюсь с помощью браузера к веб-серверу, запущенному на компьютере, он работает, что видно из вывода команды service httpd status (рис. [-@fig:002]).

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описание

Проверка работы Apache

С помощью команды ps auxZ | grep httpd нашла веб-сервер Apache в списке процессов. Его контекст безопасности - httpd\_t (рис. [-@fig:003]).

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Контекст безопасности Apache

Просмотрела текущее состояние переключателей SELinux для Apache с помощью команды sestatus -bigrep httpd (рис. [-@fig:004]).

Изображение выглядит как текст, меню, снимок экрана, Шрифт

Автоматически созданное описание

Состояние переключателей SELinux

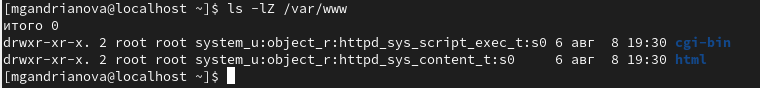
Просмотрела статистику по политике с помощью команды seinfo. Множество пользователей - 8, ролей - 40, типов - 5145. (рис. [-@fig:005]).

Изображение выглядит как текст, меню, снимок экрана, Шрифт

Автоматически созданное описание

Cтатистика по политике

Типы поддиректорий, находящихся в директории /var/www, нашли с помощью команды ls -lZ /var/www, они следующие: владелец - root, права на изменения только у владельца. Файлов в директории нет (рис. [-@fig:006]).



Типы поддиректорий

В директории /var/www/html нет файлов. (рис. [-@fig:007]).



Типы файлов

Создать файл может только суперпользователь, поэтому от его имени создаем файл touch.html cо следующим содержанием:

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

(рис. [-@fig:008]).

Изображение выглядит как текст, снимок экрана, Шрифт, черный

Автоматически созданное описание

Создание файла

Проверяю контекст созданного файла. По умолчанию это httpd\_sys\_content\_t (рис. [-@fig:009]).



Контекст файла

Обращаюсь к файлу через веб-сервер, введя в браузере адрес http://127.0.0.1/test.html. Файл был успешно отображён (рис. [-@fig:010]).

Изображение выглядит как текст, Шрифт, линия, снимок экрана

Автоматически созданное описание

Отображение файла

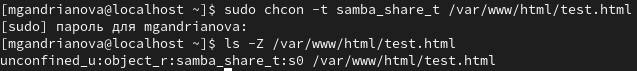
Изучила справку man httpd\_selinux. Справочной страницы не оказалось (рис. [-@fig:011]). Проверила контекст файла командой ls -Z. Рассмотрим полученный контекст детально. Так как по умолчанию пользователи CentOS являются свободными от типа (unconfined в переводе с англ. означает свободный), созданному нами файлу test.html был сопоставлен SELinux, пользователь unconfined\_u. Это первая часть контекста. Далее политика ролевого разделения доступа RBAC используется процессами, но не файлами, поэтому роли не имеют никакого значения для файлов. Роль object\_r используется по умолчанию для файлов на «постоянных» носителях и на сетевых файловых системах. (В директории /ргос файлы, относящиеся к процессам, могут иметь роль system\_r. Если активна политика MLS, то могут использоваться и другие роли, например, secadm\_r. Данный случай мы рассматривать не будем, как и предназначение :s0). Тип httpd\_sys\_content\_t позволяет процессу httpd получить доступ к файлу. Благодаря наличию последнего типа мы получили доступ к файлу при обращении к нему через браузер. (рис. [-@fig:011]).

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

Изучение справки по команде

Изменяю 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 Контекст действительно поменялся (рис. [-@fig:012]).



Изменение контекста

При попытке отображения файла в браузере получаем сообщение об ошибке (рис. [-@fig:013]).

Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Автоматически созданное описание

Отображение файла

Файл не был отображён, хотя права доступа позволяют читать этот файл любому пользователю (рис. [-@fig:014]), потому что установлен контекст, к которому процесс httpd не должен иметь доступа.



Права доступа

Просматриваю log-файлы веб-сервера Apache (рис. [-@fig:015]) и системный лог-файл: tail /var/log/messages(рис. [-@fig:016]).

Изображение выглядит как текст, снимок экрана, Шрифт, черно-белый

Автоматически созданное описание

Попытка прочесть лог-файл

Изображение выглядит как текст, снимок экрана, черно-белый, Шрифт

Автоматически созданное описание

Попытка прочесть системный лог-файл

Чтобы запустить веб-сервер Apache на прослушивание ТСР-порта 81 (а не 80, как рекомендует IANA и прописано в /etc/services) открываю файл /etc/httpd/httpd.conf для изменения. (рис. [-@fig:017]).



Изменение файла

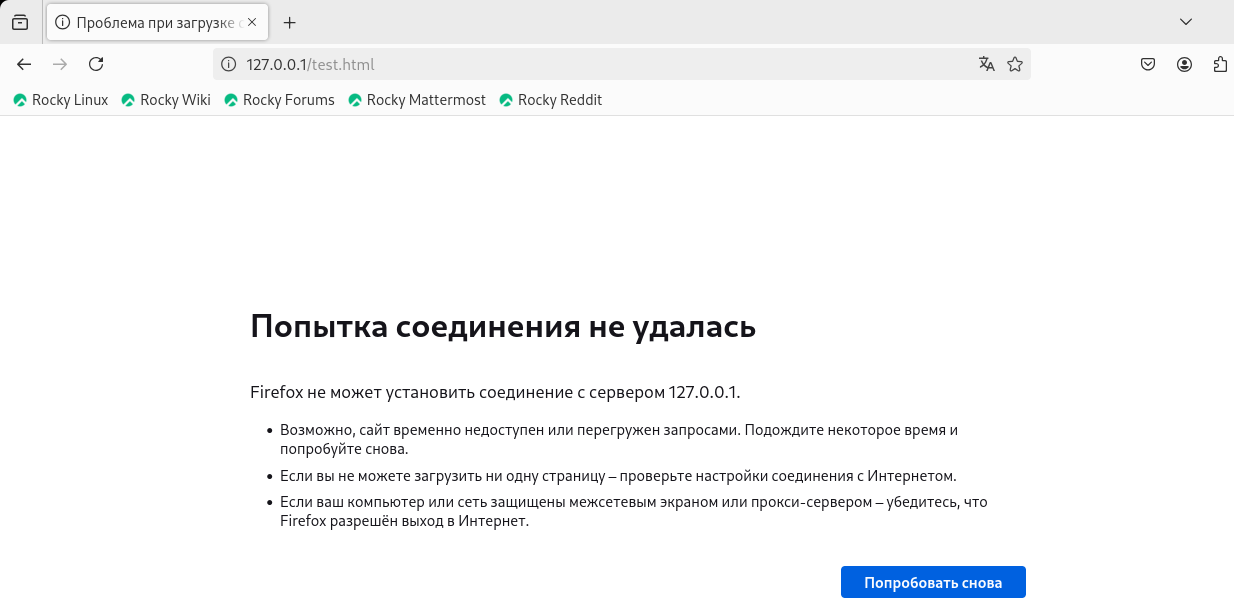
Нахожу строчку Listen 80 и заменяю её на Listen 81. (рис. [-@fig:018]).

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Изменение порта

Выполняю перезапуск веб-сервера Apache. Произошёл сбой, потому что порт 80 для локальной сети, а 81 нет (рис. [-@fig:019]).



Попытка прослушивания другого порта

Проанализируем лог-файлы: tail -nl /var/log/messages (рис. [-@fig:020]).



Проверка лог-файлов

Просмотрела файлы /var/log/http/error\_log(рис. [-@fig:021]), /var/log/http/access\_log(рис. [-@fig:022]) и /var/log/audit/audit.log(рис. [-@fig:023]) и выяснила, в каких файлах появились записи. Запись появилась в файле error\_log (рис. [-@fig:021]).

Изображение выглядит как текст, снимок экрана, Шрифт, документ

Автоматически созданное описание

Проверка лог-файлов

Изображение выглядит как текст, снимок экрана, Шрифт, черный

Автоматически созданное описание

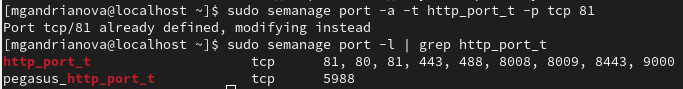
Проверка лог-файлов

Изображение выглядит как текст, снимок экрана, черно-белый, Шрифт

Автоматически созданное описание

Проверка лог-файлов

Выполняю команду semanage port -a -t http\_port\_t -р tcp 81 После этого проверяю список портов командой semanage port -l | grep http\_port\_t Порт 81 появился в списке (рис. [-@fig:024]).



Проверка портов

Перезапускаю сервер Apache (рис. [-@fig:025]).



Перезапуск сервера

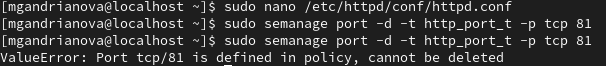
Теперь он работает, ведь мы внесли порт 81 в список портов htttpd\_port\_t (рис. [-@fig:026]).

Изображение выглядит как текст, Шрифт, линия, снимок экрана

Автоматически созданное описание

Проверка сервера

Возвращаю в файле /etc/httpd/httpd.conf порт 80, вместо 81. Проверяю, что порт 81 удален, это правда (рис. [-@fig:027]).



Проверка порта 81

Далее удаляю файл test.html, проверяю, что он удален (рис. [-@fig:028]).



Удаление файла

# Выводы

В ходе выполнения данной лабораторной работы были развиты навыки администрирования ОС Linux, получено первое практическое знакомство с технологией SELinux и проверена работа SELinux на практике совместно с веб-сервером Apache.