Oтчёт по лабораторной работе

Мандатное разграничение прав в Linux

Назарьева Алена Игоревна НФИбд-03-18

Содержание

# Цель работы

Развить навыки администрирования ОС Linux. Получить первое практическое знакомство с технологией SELinux1. Проверить работу SELinx на практике совместно с веб-сервером Apache.

# Выполнение лабораторной работы

1. Вошла в систему с полученными учётными данными и убедилась, что SELinux работает в режиме enforcing политики targeted с помощью команд getenforce и sestatus.
2. Обратилась с помощью браузера к веб-серверу, запущенному на моем компьютере, и убедилась, что последний работает: service httpd status или /etc/rc.d/init.d/httpd status (рис. -fig. 1)



Figure 1: пункты 1-2

1. Нашла веб-сервер Apache в списке процессов, определила его контекст безопасности. Как мы видим, этот процесс запущен на домене httpd\_t. Например, можно использовать команду ps auxZ | grep httpd или ps -eZ | grep httpd
2. Посмотрела текущее состояние переключателей SELinux для Apache с помощью команды sestatus -bigrep httpd (рис. -fig. 2)

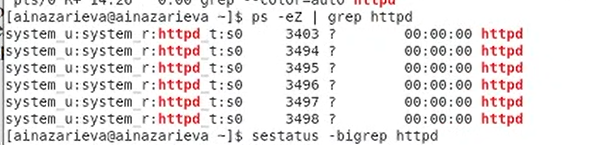


Figure 2: пункты 3-4

1. Посмотрела статистику по политике с помощью команды seinfo, также определила множество пользователей:8, ролей:14, типов:4793. (рис. -fig. 3)

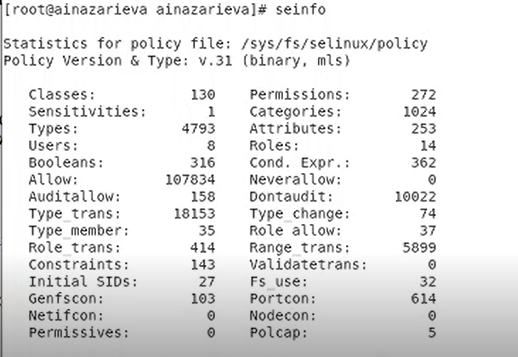


Figure 3: пункт 5

1. Определила тип файлов и поддиректорий, находящихся в директории /var/www, с помощью команды: ls -lZ /var/www: httpd\_sys\_script\_exec\_t httpd\_sys\_content\_t
2. Определила тип файлов, находящихся в директории /var/www/html: ls -lZ /var/www/html (рис. -fig. 4)

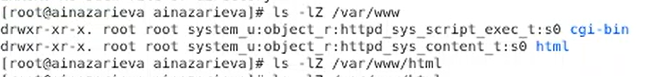
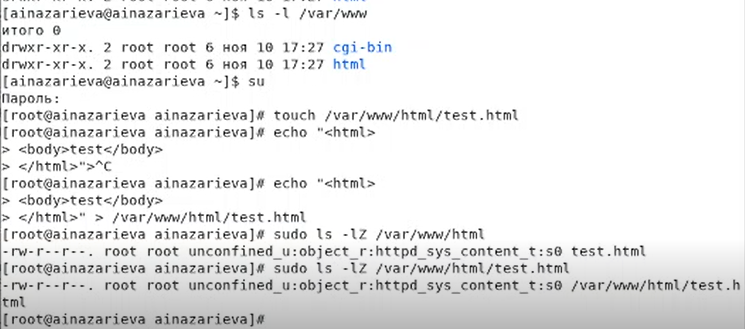


Figure 4: пункты 6-7

1. Определила круг пользователей, которым разрешено создание файлов в директории /var/www/html. rwxr-xr-x
2. Создала 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

1. Проверила контекст созданного файла. Контекст, присваиваемый по умолчанию вновь созданным файлам в директории /var/www/html: unconfined\_u:object\_r:httpd\_sys\_content\_t:s0 (рис. -fig. **¿fig:005?**)

 11. Обратилась к файлу через веб-сервер, введя в браузере адрес http://127.0.0.1/test.html. Убедилась, что файл был успешно отображён (рис. -fig. 5)

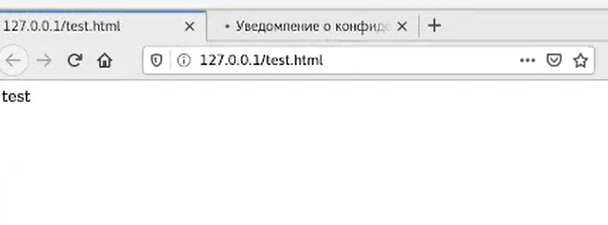


Figure 5: пункты 11

1. Изучила справку man httpd\_selinux и выяснила, какие контексты файлов определены для httpd:httpd\_sys\_content\_t;httpd\_sys\_script\_exec\_t;httpd\_sys\_script\_ro\_t; httpd\_sys\_script\_rw\_t . Они совпадают с типом файла test.html. Проверила контекст файла можно командой ls -Z. ls -Z /var/www/html/test.html
2. Изменила 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 проверила, что контекст поменялся. (рис. -fig. 6)

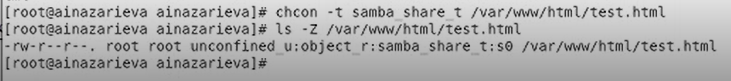


Figure 6: пункты 12-13

1. Попробовала ещё раз получить доступ к файлу через веб-сервер, введя в браузере адрес http://127.0.0.1/test.html. Получила сообщение об ошибке: Forbidden You don’t have permission to access /test.html on this server. (рис. -fig. 7)

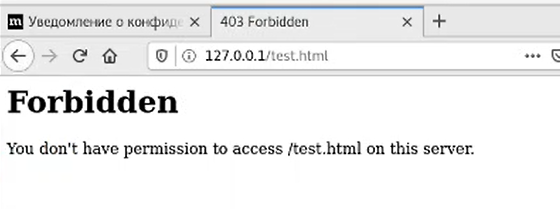


Figure 7: пункт 14

1. Проанализировала ситуацию. Доступ разрешен только между элементами с одинаковым типом, именно поэтому веб-сервер Apache может без проблем читать файл /var/www/html/test.html, который имеет тип httpd\_sys\_content\_t. В то же самое время, так как Apache запущен на домене httpd\_t и не имеет заполненных полей userid:username, он не может получить доступ к файлу home/username/test.html с другим типом, хотя этот файл доступен для чтения процессам, для которых не определена целевая политика ls -l /var/www/html/test.html Просмотрела log-файлы веб-сервера Apache. Также просмотрите системный лог-файл: tail /var/log/messages В системе оказались запущенные процессы setroubleshootd и audtd, мы также смогли увидеть ошибки, аналогичные указанным выше, в файле /var/log/audit/audit.log. (рис. -fig. 8)

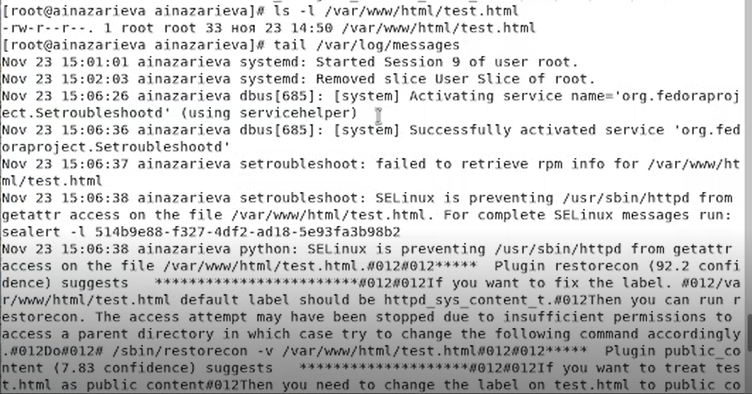


Figure 8: пункт 15

1. Попробовала 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 нашла строчку Listen 80 и заменила её на Listen 81. (рис. -fig. 9)

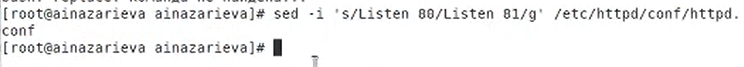


Figure 9: пункт 16

1. Выполнила перезапуск веб-сервера Apache. Сбой не произошел (рис. -fig. 10)

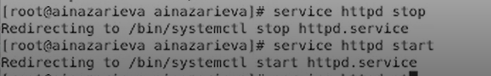


Figure 10: пункт 17

1. Проанализировала лог-файлы: tail -nl /var/log/messages Просмотрите файлы /var/log/http/error\_log, /var/log/http/access\_log и /var/log/audit/audit.log и выяснила, что записи появились только в /var/log/audit/audit.log (рис. -fig. 11)

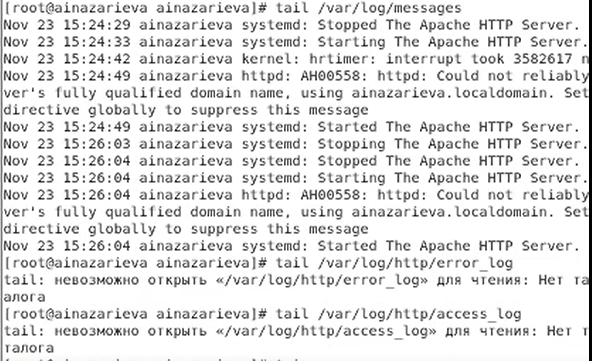


Figure 11: пункт 18-1

(рис. -fig. 12)

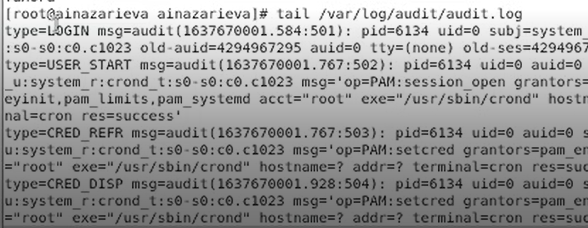


Figure 12: пункт 18-2

1. Выполнила команду semanage port -a -t http\_port\_t -р tcp 81 После этого проверила список портов командой semanage port -l | grep http\_port\_t Убедилась, что порт 81 появился в списке.
2. Попробовала запустить веб-сервер Apache ещё раз.
3. Вернула контекст httpd\_sys\_cоntent\_\_t к файлу /var/www/html/ test.html: chcon -t httpd\_sys\_content\_t /var/www/html/test.html (рис. -fig. 13)



Figure 13: пункт 19-21-1

После этого попробовала получить доступ к файлу через веб-сервер, введя в браузере адрес http://127.0.0.1:81/test.html. Увидела содержимое файла — слово «test». (рис. -fig. 14)

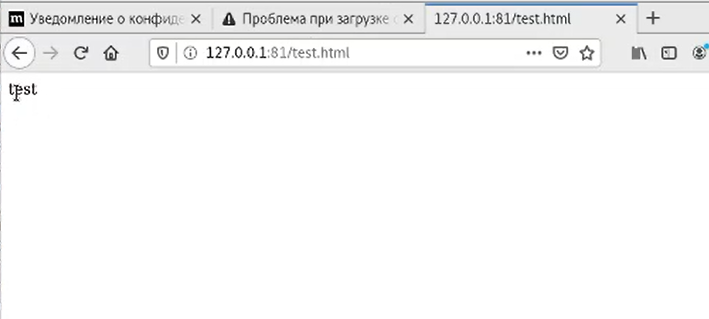


Figure 14: пункт 19-21-2

1. Исправила обратно конфигурационный файл apache, вернув Listen 80.
2. Попробовала удалить привязку http\_port\_t к 81 порту: semanage port -d -t http\_port\_t -p tcp 81 порт 81 нельзя удалить, т.к. он определен на уровне политики
3. Удалила файл /var/www/html/test.html: rm /var/www/html/test.html (рис. -fig. 15)

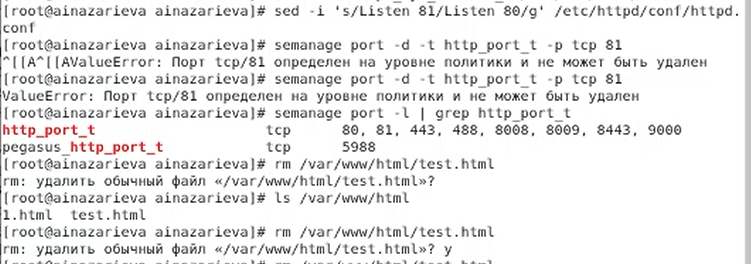


Figure 15: пункты 22-24

# Выводы

В результате выполнения работы я Развила навыки администрирования ОС Linux, Получида первое практическое знакомство с технологией SELinux1, Проверила работу SELinx на практике совместно с веб-сервером Apache.