Информационная безопасность

Л.6. Мандатное разграничение прав в Linux

Греков Максим Сергеевич

2021

Содержание

# Цель работы

Развить навыки администрирования ОС Linux.

Получить первое практическое знакомство с технологией SELinux.

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

# Подготовка лабораторного стенда

## ServerName

В конфигурационном файле */etc/httpd/httpd.conf* задали параметр ServerName (рис. 1): *ServerName test.ru*

Это нужно для того, чтобы при запуске веб-сервера не выдавались лишние сообщения об ошибках, не относящихся к лабораторной работе.

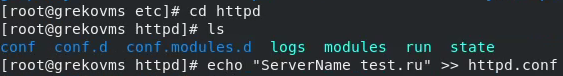


Figure 1: ServerName

## Пакетный фильтр

Проследили, чтобы пакетный фильтр был отключён, для этого воспользовались командами (рис. 2):

* iptables -F
* iptables -P INPUT ACCEPT
* iptables -P OUTPUT ACCEPT

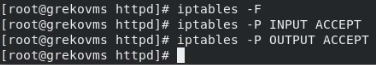


Figure 2: Пакетный фильтр

# Порядок выполнения работы

## Режим SELinux

Вошли в систему со своими учётными данными и убедитесь (рис. 3), что SELinux работает в режиме *enforcing* политики *targeted* с помощью команд:

* getenforce
* sestatus

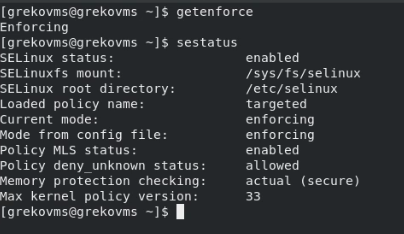


Figure 3: Режим SELinux

## Рабочий веб-сервер

Обратились через терминал к веб-серверу (рис. 4), запущенному на компьютере, и убедились, что последний работает:

* service httpd status
* /etc/rc.d/init.d/httpd status

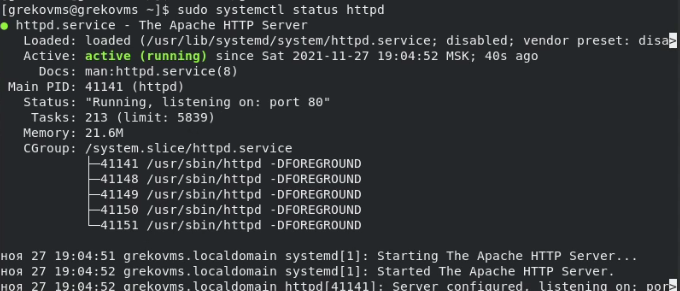


Figure 4: Рабочий веб-сервер

## Веб-сервер Apache

Нашли веб-сервер Apache (рис. 5) в списке процессов, определили его контекст безопасности:

* ps auxZ | grep httpd

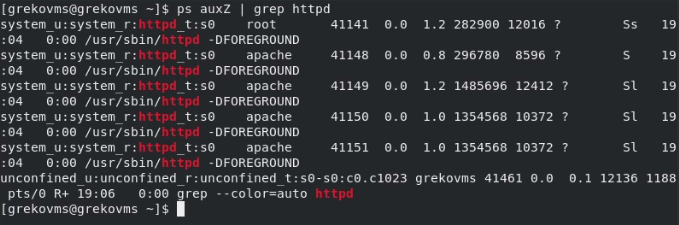


Figure 5: Веб-сервер Apache

## Переключатели SELinux для Apache

Посмотрели текущее состояние переключателей SELinux для Apache с помощью команды:

* sestatus -bigrep httpd

Обратили внимание (рис. 6), что многие из них находятся в положении *«off»*:

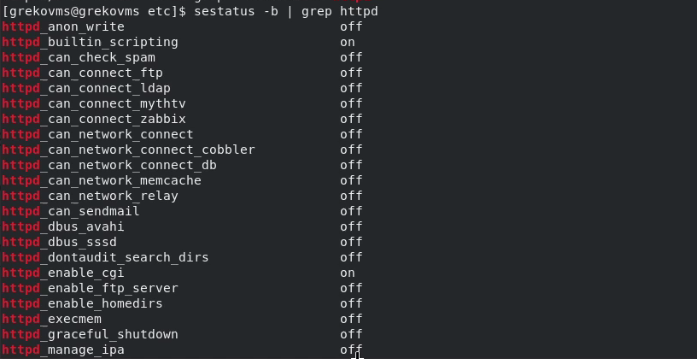


Figure 6: Переключатели SELinux для Apache

## Команда seinfo

Посмотрели статистику по политике с помощью команды seinfo (рис. 7), также определили множество пользователей, ролей, типов:

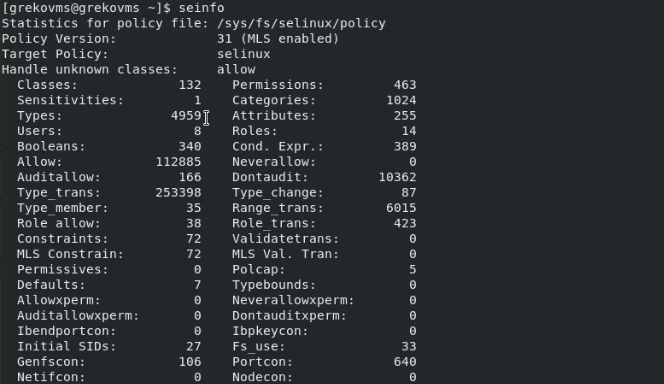


Figure 7: Команда seinfo

## Тип файлов и поддиректорий www

Определили тип файлов и поддиректорий (рис. 8), находящихся в директории */var/www*, с помощью команды:

* ls -lZ /var/www

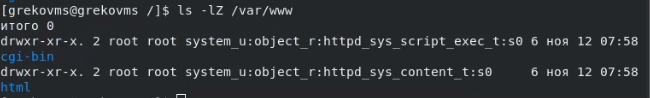


Figure 8: Тип файлов и поддиректорий www

## Тип файлов и поддиректорий html

Определили тип файлов (рис. 9), находящихся в директории */var/www/html* с помощью команды:

* ls -lZ /var/www/html

Убедились, что файлы отсутствуют.

Figure 9: Тип файлов и поддиректорий www

Figure 9: Тип файлов и поддиректорий www

## Cоздание файлов в директории html

Определили круг пользователей (рис. 10), которым разрешено создание файлов в директории */var/www/html* - пользователи с root правами:

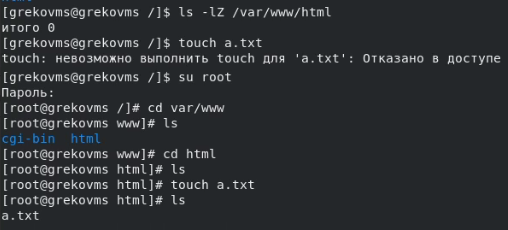


Figure 10: Тип файлов и поддиректорий www

## Создание test.html

Создали от имени суперпользователя html-файл */var/www/html/test.html* (рис. 11) следующего содержания:

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

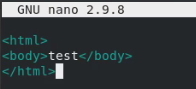


Figure 11: Создание test.html

## Проверка контекста

Проверили контекст созданного файла и контекст, присваиваемый по умолчанию (рис. 12) вновь созданным файлам в директории */var/www/html*:

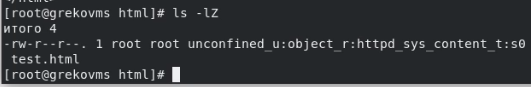


Figure 12: Проверка контекста

## Файл из браузера и справка

Обратитились к файлу через веб-сервер, введя в браузере адрес *http://127.0.0.1/test.html*.

Убедились, что файл был успешно отображён.

Изучили справку *man httpd\_selinux* и выяснили, какие контексты файлов определены для httpd.

## Изменение контекста

Изменили контекст файла */var/www/html/test.html* (рис. 13) с *httpd\_sys\_content\_t* на *samba\_share\_t*, к которому процесс httpd не должен иметь доступа, и проверили, что он поменялся:

* chcon -t samba\_share\_t /var/www/html/test.html
* ls -Z /var/www/html/test.html

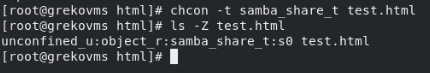


Figure 13: Изменение контекста

## Доступ к файлу

Попробовали получить доступ к файлу через веб-сервер, введя в браузере адрес *http://127.0.0.1/test.html*, и получили сообщение об ошибке.

Хоть права доступа и позволяют читать этот файл любому пользователю (рис. 14), однако из-за контекста файл нет был отображён.

* ls -l /var/www/html/test.html

Figure 14: Доступ к файлу

Figure 14: Доступ к файлу

## Log-файлы веб-сервера Apache

Просмотрели log-файлы веб-сервера Apache, также просмотрели системный лог-файл (рис. 15):

* tail /var/log/messages

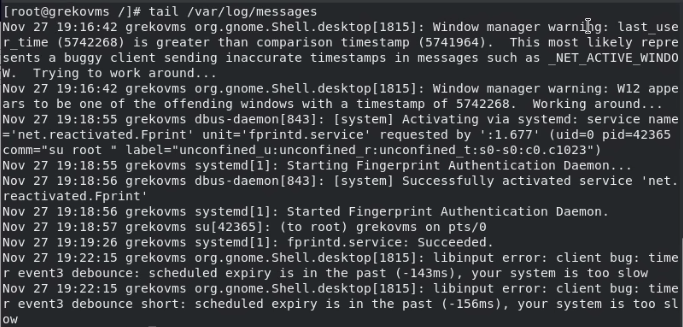


Figure 15: Log-файлы веб-сервера Apache

## Прослушивание ТСР-порта 81

Попробовали запустить веб-сервер Apache на прослушивание ТСР-порта 81 (рис. 16) (а не 80, как рекомендует IANA и прописано в */etc/services*).

Для этого в файле */etc/httpd/httpd.conf* нашли строчку *Listen 80* и заменили её на *Listen 81*, потом выполнили перезапуск сервера.



Figure 16: Прослушивание ТСР-порта 81

## Анализ файлов

Просмотрели (рис. 17) и проанализировали файлы */var/log/messages*, */var/log/http/error\_log*, */var/log/http/access\_log* и */var/log/audit/audit.log*:

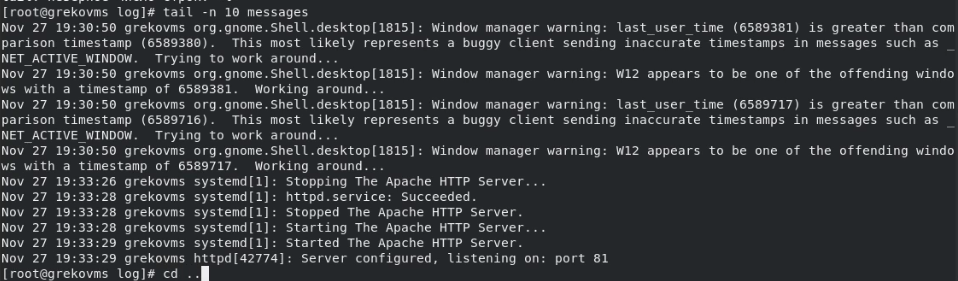


Figure 17: Анализ файлов

## Добавление порта 81

Выполнили команду *semanage port -a -t http\_port\_t -р tcp 81* (рис. 18)

После этого проверили список портов командой *semanage port -l | grep http\_port\_t* и убедились, что порт 81 появился в списке, затем выполнили перезапуск сервера.

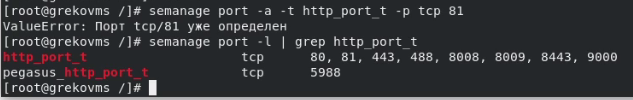


Figure 18: Добавление порта 81

## Возврат контекста

Вернули 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 попробовали получить доступ к файлу через веб-сервер, введя в браузере адрес *http://127.0.0.1:81/test.html*, увидели его содержимое.

## Откат изменений

* Исправили обратно конфигурационный файл *apache*, вернув *Listen 80*.
* Удалили привязку *http\_port\_t* к 81 порту: semanage port -d -t http\_port\_t -p tcp 81 (рис. 19)
* Удалили файл */var/www/html/test.html*: rm /var/www/html/test.html

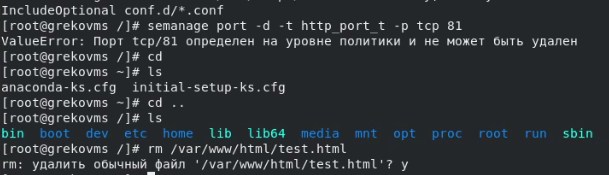


Figure 19: Откат изменений

# Вывод

Развили навыки администрирования ОС Linux.

Получили первое практическое знакомство с технологией SELinux.

Проверили работу SELinx на практике совместно с веб-сервером Apache.