Отчёт по лабораторной работе 6

Макухина Анастасия Вадимовна

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.

# Подготовка лабораторного стенда

1. Установим/обновим (за суперпользователя) веб-сервер Apache с помощью команды

yum install httpd - [Рисунок 1](Images/1.png).

1. В конфигурационном файле /etc/httpd/httpd.conf зададим параметр ServerName, чтобы при запуске веб-сервера не выдавались лишние сообщения об ошибках, не относящихся к лабораторной работе.

ServerName test.ru - [Рисунок 2](Images/2.png).

1. Проследим, чтобы пакетный фильтр был отключен или в своей рабочей конфигурации позволял подключаться к 80-му и 81-му портам протокола tcp. Добавим разрешающие правила с помощью команд:

iptables -I INPUT -p tcp –dport 80 -j ACCEPT

iptables -I INPUT -p tcp –dport 81 -j ACCEPT

iptables -I OUTPUT -p tcp –sport 80 -j ACCEPT

iptables -I OUTPUT -p tcp –sport 81 -j ACCEPT - [Рисунок 3](Images/3.png).

Можно было бы также отключить фильтр командами:

iptables -F

iptables -P INPUT ACCEPT iptables -P OUTPUT ACCEPT.

# Выполнение лабораторной работы

1. Войдём в систему с полученными учётными данными и убедимся, что SELinux работает в режиме enforcing политики targeted с помощью команд

getenforce

sestatus - [Рисунок 4](Images/4.png).

1. Обратимся с помощью браузера к веб-серверу, запущенному на нашем компьютере, и убедимся, что последний работает:

service httpd status - [Рисунок 5](Images/5.png).

1. Найдём веб-сервер Apache в списке процессов, определим его контекст безопасности. Используем команду

ps auxZ | grep httpd - [Рисунок 6](Images/6.png).

В нашем случае контекст безопасности unconfined\_u:system\_r:httpd\_t.

1. Посмотрим текущее состояние переключателей SELinux для Apache с помощью команды

sestatus -bigrep httpd - [Рисунок 8](Images/8.png).

Многие из переключателей находятся в положении «off».

1. Посмотрим статистику по политике с помощью команды

seinfo - [Рисунок 9](Images/9.png).

Таким образом, определим множество пользователей, ролей и типов. Пользователей: 8, ролей: 14, типов: 4793.

1. Определим тип файлов и поддиректорий, находящихся в директории /var/www, с помощью команды

ls -lZ /var/www - [Рисунок 9.1](Images/9.1.png).

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

ls -lZ /var/www/html - [Рисунок 9.2](Images/9.2.png).

1. Определим круг пользователей, которым разрешено создание файлов в директории /var/www/html.

[Рисунок 10](Images/10.png).

Видно, что только суперпользователь может создать файл в данной директории.

1. Создадим 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

[Рисунок 11](Images/11.png).

1. Проверим контекст созданного нами файла.

[Рисунок 12](Images/12.png).

Контекст, присваиваемый по умолчанию вновь созданным файлам в директории /var/www/html: unconfined\_u:object\_r:httpd\_sys\_content\_t.

1. Обратитесь к файлу через веб-сервер, введя в браузере адрес

*http://127.0.0.1/test.html*

Убедимся, что файл был успешно отображён - [Рисунок 13](Images/13.png).

1. Изучим справку man httpd\_selinux и выясним, какие контексты файлов определены для httpd. Сопоставим их с типом файла test.html.

ls -Z /var/www/html/test.html - [Рисунок 14](Images/14.png).

Т.к. по умолчанию пользователи CentOS являются свободными (unconfined) от типа, 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. 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.

1. Изменим 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 - [Рисунок 15](Images/15.png).

Как можно видеть, контекст успешно сменился.

1. Попробуем ещё раз получить доступ к файлу через веб-сервер, введя в браузере адрес http://127.0.0.1/test.html.

[Рисунок 16](Images/16.png).

Мы получили сообщение об ошибке.

1. Проанализируем ситуацию, просмотрев log-файлы веб-сервера Apache, системный log-файл и audit.log при условии уже запущенных процессов setroubleshootd и audtd.

[Рисунок 17](Images/17.png), [Рисунок 18](Images/18.png), [Рисунок 19](Images/19.png).

Исходя из log-файлов, мы можем заметить, что проблема в измененном контексте на шаге 13, т.к. процесс httpd не имеет доступа на samba\_share\_t. В системе оказались запущены процессы setroubleshootd и audtd, поэтому ошибки, связанные с измененным контекстом, также есть в файле /var/log/audit/audit.log.

1. Попробуем 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 найдём строчку Listen 80 и заменим её на Listen 81.

[Рисунок 20](Images/20.png).

1. Перезапустим веб-сервер Apache и попробуем обратиться к файлу через веб-сервер, введя в браузере firefox адрес *http://127.0.0.1/test.html*

[Рисунок 21](Images/21.png).

Из того, что при запуске файла через браузер появилась ошибка, можно сделать предположение, что в списках портов, работающих с веб-сервером Apache, отсутствует порт 81.

1. Проанализируем лог-файлы:

tail -nl /var/log/messages - [Рисунок 22](Images/22.png).

Во всех log-файлах появились записи, кроме /var/log/messages.

1. Выполним команду

semanage port -a -t http\_port\_t -р tcp 81

После этого проверим список портов командой

semanage port -l | grep http\_port\_t - [Рисунок 23](Images/23.png).

Убедились, что порт 81 появился в списке.

1. Попробуем теперь запустить веб-сервер Apache еще раз.

[Рисунок 24](Images/24.png).

1. Вернем контекст httpd\_sys\_content\_t к файлу /var/www/html/test.html:

chcon –t httpd\_sys\_content\_t /var/www/html/test.html - [Рисунок 25](Images/25.png).

После этого вновь попробуем получить доступ к файлу через веб-сервер, введя в браузере firefox адрес *http://127.0.0.1:81/test.html*

[Рисунок 26](Images/26.png)

Увидели слово содержимое файла - слово «test».

1. Исправим обратно конфигурационный файл apache, вернув Listen 80.

[Рисунок 27](Images/27.png).

1. Удалим привязку http\_port\_t к 81 порту: semanage port –d –t http\_port\_t –p tcp 81. Данную команду выполнить невозможно на моей версии CentOS, поэтому получаем ошибку.

[Рисунок 28](Images/28.png).

1. Удалим файл /var/www/html/test.html:

rm /var/www/html/test.html - [Рисунок 29](Images/29.png).

# Выводы

В ходе выяполнения работы я развила навыки администрирования ОС Linux. Получила первое практическое знакомство с технологией SELinux. Проверила работу SELinux на практике совместно с веб-сервером Apache.