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.

2. 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.

3. 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.

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.

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

service httpd status - Рисунок 5.

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

ps auxZ | grep httpd - Рисунок 6.

В нашем случае контекст безопасности unconfined_u:system_r:httpd_t.

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

sestatus -bigrep httpd - Рисунок 8.

Многие из переключателей находятся в положении «off».

5. Посмотрим статистику по политике с помощью команды

seinfo - Рисунок 9.

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

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

ls -lZ /var/www - Рисунок 9.1.

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

ls -lZ /var/www/html - Рисунок 9.2.

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

Рисунок 10.

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

9. 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.

10. Проверим контекст созданного нами файла.

Рисунок 12.

Контекст, присваиваемый по умолчанию вновь созданным файлам в директории /var/www/html: unconfined_u:object_r:httpd_sys_content_t.

11. 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.

12. 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.

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.

13. 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.

Как можно видеть, контекст успешно сменился.

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

Рисунок 16.

Мы получили сообщение об ошибке.

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

Рисунок 17, Рисунок 18, Рисунок 19.

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.

16. Попробуем запустить веб-сервер Apache на прослушивание TCP-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.

17. 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.

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.

18. Проанализируем лог-файлы:

tail -nl /var/log/messages - Рисунок 22.

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

19. Выполним команду

semanage port -a -t http_port_t -p tcp 81

После этого проверим список портов командой

semanage port -l | grep http_port_t - Рисунок 23.

Убедились, что порт 81 появился в списке.

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

Рисунок 24.

21. 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.

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

Увидели слово содержимое файла - слово «test».

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

Рисунок 27.

23. 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.

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

rm /var/www/html/test.html - Рисунок 29.

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.