Лабораторная работа №6

Основы Информационной Безопасности

Александрова Ульяна Вадимовна

Содержание

# 1 Цель работы

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

# 2 Задание

1. Подготовить рабочую среду;
2. Выполнить основную часть работы;
3. Сделать выводы.

# 3 Теоретическое введение

1. При подготовке стенда обратите внимание, что необходимая для работы и указанная выше политика targeted и режим enforcing используются в данном дистрибутиве по умолчанию, т.е. каких-то специальных настроек не требуется. При этом следует убедиться, что политика и режим включены, особенно когда работа будет проводиться повторно и велика вероятность изменений при предыдущем использовании системы.
2. При необходимости администратор должен разбираться в работе SELinux и уметь как исправить конфигурационный файл /etc/selinux/config, так и проверить используемый режим и политику.
3. Необходимо, чтобы был установлен веб-сервер Apache. При установке системы в конфигурации «рабочая станция» указанный пакет не ставится.
4. В конфигурационном файле /etc/httpd/httpd.conf необходимо задать параметр ServerName: ServerName test.ru, чтобы при запуске веб-сервера не выдавались лишние сообщения об ошибках, не относящихся к лабораторной работе.
5. Также необходимо проследить, чтобы пакетный фильтр был отключён или в своей рабочей конфигурации позволял подключаться к 80-у и 81-у портам протокола tcp. Отключить фильтр можно командами  
   iptables -F  
   iptables -P INPUT ACCEPT  
   iptables -P OUTPUT ACCEPT  
   либо добавить разрешающие правила:  
   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
6. Обратите внимание, что данные правила не являются «точными» и рекомендуемыми на все случаи жизни, они лишь позволяют правильно организовать работу стенда.
7. В работе специально не делается акцент, каким браузером (или какой консольной программой) будет производиться подключение к вебсерверу. По желанию могут использоваться разные программы, такие как консольные links, lynx, wget и графические konqueror, opera, firefox или др.

# 4 Выполнение лабораторной работы

Перед началом работы я обновила ПО (**yum update -y**, затем установила apache (**yum install httpd -y**).

Вошла в систему с и убедилась, что SELinux работает в режиме enforcing политики targeted с помощью команд **getenforce и sestatus** (рис. 1).

![Рис. 1: Проверка работы SELinux](data:application/octet-stream;base64,)

Рис. 1: Проверка работы SELinux

Чтобы работать с библиотекой httpd, скачала ее (рис. 2).

![Рис. 2: Установка библиотеки](data:application/octet-stream;base64,)

Рис. 2: Установка библиотеки

Убедилась, что веб-сервер работает при помощи утилиты **service httpd start** (рис. 3).

![Рис. 3: Запустила работу Apache](data:application/octet-stream;base64,)

Рис. 3: Запустила работу Apache

Контекст безопасности - system\_u:system\_r (рис. 4).

![Рис. 4: Проверка](data:application/octet-stream;base64,)

Рис. 4: Проверка

Посмотрела текущее состояние переключателей SELinux для Apache с помощью команды **sestatus -b | grep httpd** (рис. 5).

![Рис. 5: Текущее состояние переключателей](data:application/octet-stream;base64,)

Рис. 5: Текущее состояние переключателей

Посмотрела статистику по политике с помощью команды seinfo. Типы: 5135; пользователи: 8; роли: 15 (рис. 6).

![Рис. 6: Статистика](data:application/octet-stream;base64,)

Рис. 6: Статистика

Определила тип файлов и поддиректорий, находящихся в директории /var/www, с помощью команды **ls -lZ /var/www** (папки). Определила круг пользователей, которым разрешено создание файлов в директории /var/www/html (суперпользователю) (рис. 7).

![Рис. 7: Папка www](data:application/octet-stream;base64,)

Рис. 7: Папка www

Определила тип файлов, находящихся в директории /var/www/html утилитой **ls -lZ /var/www/html** (не отобразилось ничего) (рис. 8).

![Рис. 8: Папка html](data:application/octet-stream;base64,)

Рис. 8: Папка html

Создала html-файл /var/www/html/test.html (рис. 9).

![Рис. 9: Файл html](data:application/octet-stream;base64,)

Рис. 9: Файл html

Проверила контекст созданного файла (httpd\_sys\_content\_t) (рис. 10).

![Рис. 10: Контекст](data:application/octet-stream;base64,)

Рис. 10: Контекст

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

![Рис. 11: Веб-страничка](data:application/octet-stream;base64,)

Рис. 11: Веб-страничка

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. Изменила контекст файла /var/www/html/test.html с **httpd\_sys\_content\_t** на **samba\_share\_t** с помощью утилиты **chcon -t samba\_share\_t /var/www/html/test.html**. Контекст поменялся (рис. 12).

![Рис. 12: samba_share_t](data:application/octet-stream;base64,)

Рис. 12: samba\_share\_t

Попробовала ещё раз получить доступ к файлу через веб-сервер. Ошибка :(( (рис. 13).

![Рис. 13: Веб-страница](data:application/octet-stream;base64,)

Рис. 13: Веб-страница

Проанализировала ситуацию. Файл не отображается, так как этот тип не позволяет процессу httpd получить доступ к файлу. Также просмотрела системный лог-файл tail /var/log/messages (рис. 14).

![Рис. 14: Лог-файл](data:application/octet-stream;base64,)

Рис. 14: Лог-файл

Попробовала запустить веб-сервер Apache на прослушивание ТСР-порта 81. Для этого в файле /etc/httpd/httpd.conf поменяла строчку Listen 80 на Listen 81 (рис. 15).

![Рис. 15: Изменение файла](data:application/octet-stream;base64,)

Рис. 15: Изменение файла

Выполнила перезапуск веб-сервера Apache. Сбой не произошел…. Проанализировала лог-файлы tail -nl /var/log/messages, /var/log/http/error\_log, /var/log/http/access\_log и /var/log/audit/audit.log (рис. 16), (рис. 17), (рис. 18).

![Рис. 16: Лог](data:application/octet-stream;base64,)

Рис. 16: Лог

![Рис. 17: Лог](data:application/octet-stream;base64,)

Рис. 17: Лог

![Рис. 18: Лог](data:application/octet-stream;base64,)

Рис. 18: Лог

Выполнила команду **semanage port -a -t http\_port\_t -р tcp 81**, проверила список портов командой **semanage port -l | grep http\_port\_t** (рис. 19).

![Рис. 19: Настройка порта 81](data:application/octet-stream;base64,)

Рис. 19: Настройка порта 81

Попробовала запустить веб-сервер Apache ещё раз. Не сработало…. (рис. 20).

![Рис. 20: Веб-страница](data:application/octet-stream;base64,)

Рис. 20: Веб-страница

Вернула контекст httpd\_sys\_cоntent\_\_t к файлу /var/www/html/ test.html.  
После этого попробовала получить доступ к файлу через веб-сервер (рис. 21).

![Рис. 21: Веб-страница](data:application/octet-stream;base64,)

Рис. 21: Веб-страница

Исправила обратно конфигурационный файл apache, вернув Listen 80. Удалила привязку http\_port\_t к 81 порту, но появилась ошибка, что этот порт удалить невозможно, даже через суперпользователя.  
Удалила файл /var/www/html/test.html (рис. 22).

![Рис. 22: Уаление](data:application/octet-stream;base64,)

Рис. 22: Уаление

# 5 Выводы

Я развила навыки администрирования ОС Linux, получила первое практическое знакомство с технологией SELinux.