**Лабораторная работа N° 6**

Информационная безопасность

Леон Фернандо Хосе Фернандо | НПМбд02-20

**Содержание**

[**1 Цель работы** 4](#_Toc147541021)

[**2 Теоретическое введение** 4](#_Toc147541022)

[**3 Выполнение лабораторной работы** 4](#_Toc147541023)

[**3.1 Создание программы** **Ошибка! Закладка не определена.**](#_Toc147541024)

[**3.2Исследование Sticky-бита** **Ошибка! Закладка не определена.**](#_Toc147541025)

[**4 Выводы** 10](#_Toc147541026)

[**5 Список Литературы** 10](#_Toc147541027)

**Список иллюстраций**

[Рисунок 1. Предварительная подготовка **Ошибка! Закладка не определена.**](#_Toc147541074)

[Рисунок 2. Команда “whereis” **Ошибка! Закладка не определена.**](#_Toc147541075)

[Рисунок 3.Вход в систему и создание программы **Ошибка! Закладка не определена.**](#_Toc147541076)

[Рисунок 4. Код программы simpleid.c **Ошибка! Закладка не определена.**](#_Toc147541077)

[Рисунок 5. Компиляция и выполнение программы simpleid **Ошибка! Закладка не определена.**](#_Toc147541078)

[Рисунок 6. Усложнение программы **Ошибка! Закладка не определена.**](#_Toc147541079)

[Рисунок 7. Переименование программы в simpleid2.c **Ошибка! Закладка не определена.**](#_Toc147541080)

[Рисунок 8. Компиляция и выполнение программы simpleid2 **Ошибка! Закладка не определена.**](#_Toc147541081)

[Рисунок 9. Установка новых атрибутов (SetUID) **Ошибка! Закладка не определена.**](#_Toc147541082)

[Рисунок 10. Запуск simpleid2 после установки SetUID **Ошибка! Закладка не определена.**](#_Toc147541083)

[Рисунок 11. Запуск simpleid2 после установки SetGID **Ошибка! Закладка не определена.**](#_Toc147541084)

[Рисунок 12. Код программы readfile.c **Ошибка! Закладка не определена.**](#_Toc147541085)

[Рисунок 13. Смена владельца и прав доступа у файла readfile.c **Ошибка! Закладка не определена.**](#_Toc147541086)

[Рисунок 14. Запуск программы readfile **Ошибка! Закладка не определена.**](#_Toc147541087)

[Рисунок 15. Создание файла file01.txt **Ошибка! Закладка не определена.**](#_Toc147541088)

[Рисунок 16. Попытка выполнить действия над файлом file01.txt от имени пользо. **Ошибка! Закладка не определена.**](#_Toc147541089)

[Рисунок 17. Удаление атрибута t (Sticky-бита) **Ошибка! Закладка не определена.**](#_Toc147541090)

[Рисунок 18. Возвращение атрибута t (Sticky-бита) **Ошибка! Закладка не определена.**](#_Toc147541091)

# **1 Цель работы**

Совершенствуйте навыки администрирования ОС Linux, приобретайте начальный практический опыт работы с технологией SELinux и оценивайте функциональность SELinux на практике в сочетании с веб-сервером Apache.

# **2 Теоретическое введение**

SELinux (Security-Enhanced Linux) обеспечивает усиление защиты путем внесения изменений как на уровне ядра, так и на уровне пространства пользователя, что превращает ее в действительно «непробиваемую» операционную систему. Впервые эта система появилась в четвертой версии CentOS, а в 5 и 6 версии реализация была существенно дополнена и улучшена.

SELinux имеет три основных режим работы:

* Enforcing: Режим по-умолчанию. При выборе этого режима все действия,
* которые каким-то образом нарушают текущую политику безопасности,
* будут блокироваться, а попытка нарушения будет зафиксирована в журнале.
* Permissive: В случае использования этого режима, информация о всех дей-
* ствиях, которые нарушают текущую политику безопасности, будут зафик-
* сированы в журнале, но сами действия не будут заблокированы.
* Disabled: Полное отключение системы принудительного контроля доступа.

Политика SELinux определяет доступ пользователей к ролям, доступ ролей к

доменам и доступ доменов к типам. Контекст безопасности — все атрибуты SELinux —роли, типы и домены.

Apache — это свободное программное обеспечение, с помощью которого можно создать веб-сервер. Данный продукт возник как доработанная версия другого HTTP-клиента от национального центра суперкомпьютерных приложений (NCSA).

Для чего нужен Apache сервер:

* чтобы открывать динамические PHP-страницы,
* для распределения поступающей на сервер нагрузки,
* для обеспечения отказоустойчивости сервера,
* чтобы потренироваться в настройке сервера и запуске PHP-скриптов.

Apache является кроссплатформенным ПО и поддерживает такие операционные системы, как Linux, BSD, MacOS, Microsoft, BeOS и другие.

# **3 Выполнение лабораторной работы**

Вошел в систему под своей учетной записью и убедилась, что SELinux работает в режиме enforcing политики targeted с помощью команд “getenforce” и “sestatus”

Изображение выглядит как текст, электроника, снимок экрана, программное обеспечение

Автоматически созданное описание

Обратился с помощью браузера к веб-серверу, запущенному на моем компьютере, и убедилась, что последний работает с помощью команды “service httpd status”

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

С помощью команды “ps auxZ | grep httpd” определил контекст безопасности

веб-сервера Apache - httpd\_t

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Автоматически созданное описание

Посмотрел текущее состояние переключателей SELinux для Apache с помощью команды “sestatus -bigrep httpd”, многие из переключателей находятся в положении “off”

Изображение выглядит как текст, снимок экрана, программное обеспечение, компьютер

Автоматически созданное описание

Изображение выглядит как текст, программное обеспечение, Шрифт, снимок экрана

Автоматически созданное описание

Я использовал команду "ls -lZ /var/www" для просмотра файлов и подкаталогов, расположенных в каталоге /var/www. Используя команду "ls -lZ /var/www/html", я определил, что в этом каталоге нет файлов. Только владелец/суперпользователь имеет привилегию создавать файлы в каталоге /var/www/html.

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

От имени суперпользователя создал html-файл /var/www/html/test.html. Контекст созданного файла - httpd\_sys\_content\_t.

Обратился к файлу через веб-сервер, введя в браузере адрес “http://127.0.0.1/test.html”.

Файл был успешно отображен.

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Изучив справочную страницу "man httpd\_selinux", я обнаружил, что для httpd определены следующие файловые контексты: httpd\_sys\_content\_t, httpd\_sys\_script\_exec\_t, httpd\_sys\_script\_ro\_t, httpd\_sys\_script\_rw\_t, httpd\_sys\_script\_ra\_t и httpd\_unconfined\_script\_exec\_trace.

Контекст моего файла изначально был установлен в httpd\_sys\_content\_t (в этом случае содержимое должно быть доступно для всех скриптов httpd и самого демона httpd). Я изменил контекст файла на samba\_share\_t, используя команду "sudo chcon -t samba\_share\_t /var/www/html/test.html " и убедился, что контекст был успешно изменен.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описание

Попробовал еще раз получить доступ к файлу через веб-сервер, введя в браузере адрес “http://127.0.0.1/test.html” и получила сообщение об ошибке (т.к. к установленному ранее контексту процесс httpd не имеет доступа)

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

С помощью команды "ls -l /var/www/html/test.html , " Я подтвердил, что любой

пользователь может прочитать этот файл. Я также просмотрел файл системного журнала веб-сервера Apache, используя команду "sudo tail /var/log/messages", которая отображает ошибки.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описание

В файле /etc/httpd/conf/httpd.conf заменил строчку “Listen 80” на “Listen 81”, чтобы установить веб-сервер Apache на прослушивание TCP-порта 81

Изображение выглядит как текст, программное обеспечение, снимок экрана, Шрифт

Автоматически созданное описание

Перезапускаем веб-сервер Apache и анализирует лог-файлы командой “tail -nl

/var/log/messages”

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описание

Просмотрел файлы “var/log/http/error\_log”, “/var/log/http/access\_log” и “/var/log/audit/audit.log” и выяснил, что запись появился в последнем файле

Выполнил команду “semanage port -a -t http\_port\_t -р tcp 81” и убедился, что порт TCP-81 установлен. Проверил список портов командой “semanage port -l | grep http\_port\_t”, убедился, что порт 81 есть в списке и запускаем веб-сервер Apache снова

Изображение выглядит как текст, Шрифт, линия, число

Автоматически созданное описание

Вернул контекст “httpd\_sys\_cоntent\_t” файлу “/var/www/html/test.html” командой “chcon -t httpd\_sys\_content\_t /var/www/html/test.html” (рис. 3.16) и после этого попробовала получить доступ к файлу через веб-сервер, введя адрес “http://127.0.0.1:81/test.html”, в результате чего увидела содежимое файла - слово “test”

Я исправил конфигурационный файл Apache обратно на "Listen 80". Я попытался удалить привязку http\_port к порту 81, используя команду "semanage port -d -t http\_port\_t -p tcp 81", но поскольку этот порт определен на уровне политики, он не может быть удален.

Удалил файл “/var/www/html/test.html” командой “rm /var/www/html/test.html”

Изображение выглядит как текст, программное обеспечение, Шрифт, веб-страница

Автоматически созданное описание

# **4 Выводы**

В ходе этой лабораторной работы я развил навыки администрирования ОС Linux, получил свое первое практическое представление о технологии SELinux и проверил функциональность SELinux на практике в сочетании с веб-сервером Apache.

# **5 Список Литературы**

1. SELinux – описание и особенности работы с системой [Электронный ресурс]. URL: <https://habr.com/ru/company/kingservers/blog/209644/>.
2. Что такое Apache и зачем он нужен? [Электронный ресурс]. URL: https: //2domains.ru/support/vps-i-servery/shto-takoye-apache.