Отчёт по лабораторной работе №6

дисциплина: Информационная безопасность

Абрамян Артём Арменович

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

Развить навыки администрирования ОС Linux. Получить первое практическое знакомство с технологией SELinux. Проверить работу SELinux на практике совместно 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 действиях, которые нарушают текущую политику безопасности, будут зафиксированы 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).

Выполнение работы возможно как в дисплейном классе факультета физико-математических и естественных наук РУДН, так и дома. Описание выполнения работы приведено для дисплейного класса со следующими характеристиками: – Intel Core i3-550 3.2 GHz, 4 GB оперативной памяти, 20 GB свободного места

на жёстком диске; – ОС Linux Gentoo (http://www.gentoo.ru/); – VirtualBox верс. 6.1 или старше; – каталог с образами ОС для работающих в дисплейном классе: /afs/dk.sci.pfu.edu.ru/common/files/iso/.

В операционной системе Linux есть много отличных функций безопасности, но одна из самых важных - это система прав доступа к файлам. Изначально каждый файл имел три параметра доступа: Чтение - разрешает получать содержимое файла, но на запись нет. Для каталога позволяет получить список файлов и каталогов, расположенных в нем Запись - разрешает записывать новые данные в файл или изменять существующие, а также позволяет создавать и изменять файлы и каталоги Выполнение - невозможно выполнить программу, если у нее нет флага выполнения. Этот атрибут устанавливается для всех программ и скриптов, именно с помощью него система может понять, что этот файл нужно запускать как программу Каждый файл имеет три категории пользователей, для которых можно устанавливать различные сочетания прав доступа:

Владелец - набор прав для владельца файла, пользователя, который его создал или сейчас установлен его владельцем. Обычно владелец имеет все права, чтение, запись и выполнение Группа - любая группа пользователей, существующая в системе и привязанная к файлу. Но это может быть только одна группа и обычно это группа владельца, хотя для файла можно назначить и другую группу Остальные - все пользователи, кроме владельца и пользователей, входящих в группу файла

Команды, которые могут понадобиться при работе с правами доступа: “ls -l” - для просмотра прав доступа к файлам и каталогам “chmod категория действие флаг файл или каталог” - для изменения прав доступа к файлам и каталогам (категорию действие и флаг можно заменить на набор из трех цифр от 0 до 7) Значения флагов прав: — - нет никаких прав –x - разрешено только выполнение файла, как программы, но не изменение и не чтение -w- - разрешена только запись и изменение файла -wx - разрешено изменение и выполнение, но в случае с каталогом, невозможно посмотреть его содержимое r– - права только на чтение r-x - только чтение и выполнение, без права на запись rw- - права на чтение и

запись, но без выполнения rwx - все права

3 Выполнение лабораторной работы

1. Вошёл в систему под своей учетной записью и убедилась, что SELinux работает в режиме enforcing политики targeted с помощью команд “getenforce” и “sestatus” (рис. 3.1)

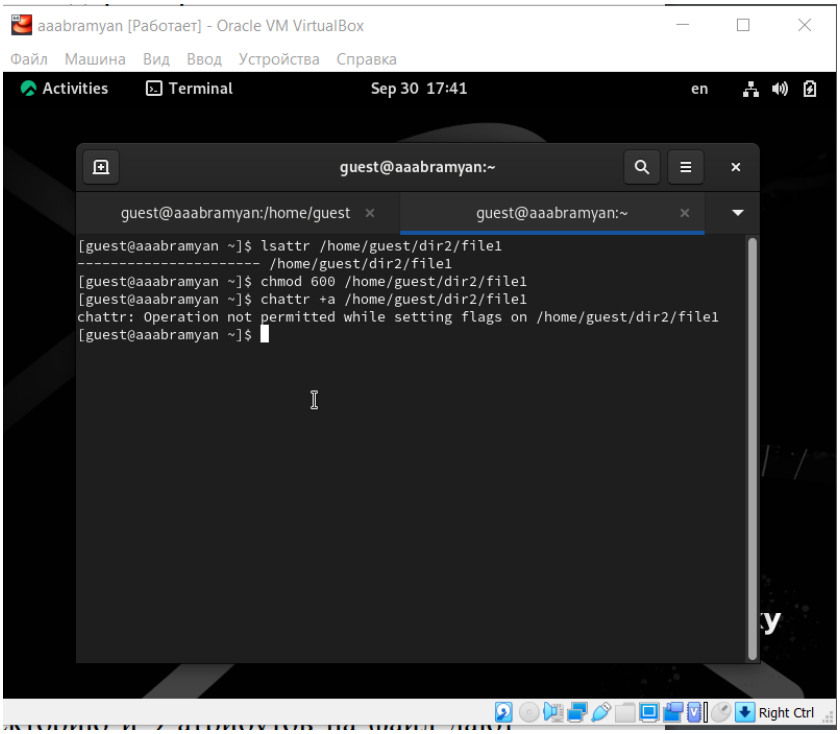

Рис. 3.1: Проверка режима enforcing политики targeted

2. Обратился с помощью браузера к веб-серверу, запущенному на моем компьютере, убедился, что последний работает с помощью команды “service httpd status” (рис. 3.2)

Проверка работы веб-сервера

Рис. 3.2: Проверка работы веб-сервера

3. С помощью команды “ps auxZ | grep httpd” определил контекст безопасности веб-сервера Apache - httpd_t (рис. 3.3)

Контекст безопасности веб-сервера Apache

Рис. 3.3: Контекст безопасности веб-сервера Apache

4. Посмотрел текущее состояние переключателей SELinux для Apache с помощью команды “sestatus -bigrep httpd”, многие из переключателей находятся в положении “off” (рис. 3.4)

Текущее состояние переключателей SELinux

Рис. 3.4: Текущее состояние переключателей SELinux

5. С помощью команды “ls -lZ /var/www” посмотрел файлы и поддиректории, находящиеся в директории /var/www. Используя команду “ls -lZ /var/www/html”, определил, что в данной директории файлов нет. Только владелец/суперпользователь может создавать файлы в директории /var/www/html (рис. **??**)

Просмотр файлов и поддиректориий в директории /var/www

Рис. 3.5: Просмотр файлов и поддиректориий в директории /var/www

6. Обратился к файлу через веб-сервер, введя в браузере адрес “http://127.0.0.1/test.html”. Файл был успешно отображен. Изучив справку man httpd_selinux, выяснил, что для httpd определены следующие контексты файлов: httpd_sys_content_t, httpd_sys_script_exec_t, httpd_sys_script_ro_t,

httpd_sys_script_rw_t, httpd_sys_script_ra_t, httpd_unconfined_script_exec_t. Контекст моего файла - httpd_sys_content_t (в таком случае содержимое должно быть доступно для всех скриптов httpd и для самого демона). Изменил контекст файла на samba_share_t командой “sudo chcon -t samba_share_t /var/www/html/test.html” и проверил, что контекст поменялся (рис. 3.10)

Запуск программы readfile

Рис. 3.6: Запуск программы readfile

7. Командой “ls -l /var/www/html/test.html” убедился, что читать данный файл может любой пользователь. Просмотрел системный лог-файл веб-сервера Apache командой “sudo tail /var/log/messages”, отображающий ошибки (рис. **??**)

Просмотр log-файла

Рис. 3.7: Просмотр log-файла

8. От имени пользователя guest2 попробовал прочитать файл командой “cat /tmp/file01.txt” - это удалось. Далее попытался дозаписать в файл слово test2, проверить содержимое файла и записать в файл слово test3, стерев при этом всю имеющуюся в файле информацию - эти операции удалось выполнить только в случае, если еще дополнительно разрешить чтение и запись для группы пользователей командой “chmod g+rw /tmp/file01.txt”. От имени пользователя guest2 попробовал удалить файл - это не удается ни в каком из случаев, возникает ошибка (рис. **??**)

Попытка выполнить действия над файлом file01.txt от имени пользователя guest2

Рис. 3.8: Попытка выполнить действия над файлом file01.txt от имени пользователя guest2

9. Повысил права до суперпользователя командой “su -” и выполнил команду, снимающую атрибут t с директории /tmp “chmod -t /tmp”. После чего покинул режим суперпользователя командой “exit”. Повторил предыдущие шаги. Теперь мне удалось удалить файл file01.txt от имени пользователя, не являющегося его владельцем (рис. **??**)

Удаление атрибута t (Sticky-бита) и повторение действий

Рис. 3.9: Удаление атрибута t (Sticky-бита) и повторение действий

10. Удалил файл “/var/www/html/test.html” командой “rm /var/www/html/test.html” /tmp (рис. **??**)

Удаление файла test.html

Рис. 3.10: Удаление файла test.html

4 Выводы

В данной лабораторной работе мне успешно удалось Развить навыки администрирования ОС Linux. Получить первое практическое знакомство с технологией SELinux. Проверить работу SELinux на практике совместно с веб-сервером Apache.
Библиографический список

1. Документация Rocky (https://docs.rockylinux.org/)