Отчёт по лабораторной работе №5
Информационная безопасность

Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

Выполнил: Маляров Семён Сергеевич,
НПИбд-01-21, 1032209505

Содержание

1 Цель работы

Изучить механизмы изменения идентификаторов, применения SetUID- и Sticky-битов. Получить практические навыки работы в консоли с дополнительными атрибутами. Рассмотреть работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

2 Теоретическое введение

1. Дополнительные атрибуты файлов Linux

В Linux существует три основных вида прав — право на чтение (read), запись (write) и выполнение (execute), а также три категории пользователей, к которым они могут применяться — владелец файла (user), группа владельца (group) и все остальные (others). Но, кроме прав чтения, выполнения и записи, есть еще три дополнительных атрибута [1].

- **Sticky bit**

Используется в основном для каталогов, чтобы защитить в них файлы. В такой каталог может писать любой пользователь. Но, из такой директории пользователь может

удалить только те файлы, владельцем которых он является. Примером может служить директория /tmp, в которой запись открыта для всех пользователей, но нежелательно удаление чужих файлов.

- **SUID (Set User ID)**

Атрибут исполняемого файла, позволяющий запустить его с правами владельца. В Linux приложение запускается с правами пользователя, запустившего указанное приложение. Это обеспечивает дополнительную безопасность т.к. процесс с правами пользователя не сможет получить доступ к важным системным файлам, которые принадлежат пользователю root.

- **SGID (Set Group ID)**

Аналогичен suid, но относиться к группе. Если установить sgid для каталога, то все файлы созданные в нем, при запуске будут принимать идентификатор группы каталога, а не группы владельца, который создал файл в этом каталоге.

- **Обозначение атрибутов sticky, suid, sgid**

Специальные права используются довольно редко, поэтому при выводе программы ls -l символ, обозначающий указанные атрибуты, закрывает символ стандартных прав доступа.

Пример:

```
rwsrwsrwt
```

где первая s — это suid, вторая s — это sgid, а последняя t — это sticky bit

В приведенном примере не понятно, rwt — это rw- или rwx? Определить это просто. Если t маленькое, значит x установлен. Если T большое, значит x не установлен. То же самое правило распространяется и на s.

В числовом эквиваленте данные атрибуты определяются первым символом при четырехзначном обозначении (который часто опускается при назначении прав), например в правах 1777 — символ 1 обозначает sticky bit. Остальные атрибуты имеют следующие числовое соответствие:

```
1 – установлен sticky bit
2 – установлен sgid
4 – установлен suid
```

2. Компилятор GCC

GCC - это свободно доступный оптимизирующий компилятор для языков C, C++. Собственно программа gcc это некоторая надстройка над группой компиляторов, которая способна анализировать имена файлов, передаваемые ей в качестве аргументов, и определять, какие действия необходимо выполнить. Файлы с расширением .cc или .C рассматриваются, как файлы на языке C++, файлы с расширением .c как программы на языке C, а файлы с расширением .o считаются объектными [2].

3 Выполнение лабораторной работы

3.1 Подготовка лабораторного стенда

Убедимся, что в системе установлен компилятор gcc. После чего отключим систему запретов до очередной перезагрузки системы и выполним проверку

3.2 Создание программы

1. Войдём в систему от имени пользователя guest и создадим программу simpleid.c
2. Скомплилируем программу и убедимся, что файл программы создан
3. Затем выполним программу simpleid
4. Выполним системную программу id и сравним полученный нами результат с данными предыдущего пункта задания
5. Усложним программу, добавив вывод действительных идентификаторов. После чего получившуюся программу назовём simpleid2.c
6. Скомпилируем и запустим simpleid2.c
7. От имени суперпользователя выполним команды
8. Повысив временно свои права с помощью su, поясним, что делают эти команды

От имени суперпользователя выполнил команды “sudo chown root:guest /home/guest/simpleid2” и “sudo chmod u+s /home/guest/simpleid2”, затем выполнил проверку правильности установки новых атрибутов и смены владельца файла simpleid2 командой “sudo ls -l /home/guest/simpleid2”. Этими командами была произведена смена пользователя файла на root и установлен SetUID-бит.

9. Далее выполним проверку правильности установки новых атрибутов и смены владельца файла simpleid2
10. Запустим simpleid2 и id и сравним результаты
11. Теперь проделаем тоже самое относительно SetGID-бита
12. Создадим программу readfile.c
13. Теперь её откомпилируем
14. Сменим владельца у файла readfile.c и изменим права так, чтобы только суперпользователь (root) мог прочитать его, а guest не мог. Затем сменим у программы readfile владельца и установим SetU’D-бит
15. Проверим, что пользователь guest не может прочитать файл readfile.c, может ли программа readfile прочитать файл readfile.c, и может ли программа readfile прочитать файл /etc/shadow?

3.3 Исследование Sticky-бита

1. Выясним, установлен ли атрибут Sticky на директории /tmp
2. От имени пользователя guest создадим файл file01.txt в директории /tmp со словом test
3. Просмотрим атрибуты у только что созданного файла и разрешим чтение и запись для категории пользователей «все остальные»

4. От пользователя guest2 (не являющегося владельцем) попробуем прочитать файл /tmp/file01.txt
5. От пользователя guest2 попробуем дозаписать в файл
6. Проверим содержимое файла
7. От пользователя guest2 попробуем записать в файл /tmp/file01.txt слово test3, стерев при этом всю имеющуюся в файле информацию
8. Проверим содержимое файла
9. От пользователя guest2 попробуем удалить файл /tmp/file01.txt
10. Повысим свои права до суперпользователя командой su и выполним после этого команду, снимающую атрибут t (Sticky-бит) с директории /tmp
11. Покинем режим суперпользователя и от пользователя guest2 проверим, что атрибута t у директории /tmp нет
12. Повторим предыдущие шаги
13. Повысим свои права до суперпользователя и вернём атрибут t на директорию /tmp

4 Вывод

В ходе выполнения лабораторной работы были изучены механизмы изменения идентификаторов и применения SetUID- и Sticky-битов. Получены практические навыки работы в консоли с дополнительными атрибутами, а также была рассмотрена работа механизма смены идентификатора процессов пользователей и влияние бита Sticky на запись и удаление файлов.

5 Список литературы. Библиография

[1] Дополнительные атрибуты: https://tokmakov.msk.ru/blog/item/141

[2] Компилятор GSS: http://parallel.imm.uran.ru/freesoft/make/instrum.html