Презентация по лабораторной работе №5

Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

Лушин А.А.

18 февраля 2005

Российский университет дружбы народов, Москва, Россия

Факультет Физико-математических и естественных наук

Информация

Докладчик

- Лушин Артём Андреевич
- Бакалавр направления компьютерные и информационные науки
- Кафедра теории вероятности и кибербезопасности
- Российский университет дружбы народов
- Редактор Первого Федерального канала
- lusin5745@gmail.com

Вводная часть

Цели и задачи

- Изучение механизмов изменения идентификаторов, применения SetUID- и Sticky-битов. Получение практических навыков работы в консоли с дополнительными атрибутами. Рассмотрение работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

Ход работы

Создание программы

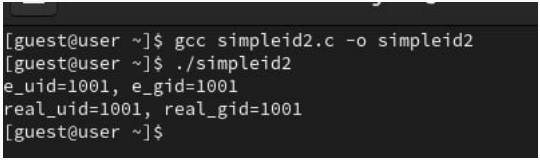
Созданий программ в целом

Для начала работы мы должны создать файл “simpleid.c”. В этот файл просто вписываем код, основанный на языке C++.

```c
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>

int
main ()
{
uid_t real_uid = getuid();
uid_t e_uid = geteuid ();

gid_t real_gid = getgid ();
gid_t e_gid = getegid ();

printf ("e_uid=%d, e_gid=%d\n", e_uid, e_gid);
printf ("real_uid=%d, real_gid=%d\n", real_uid, real_gid);

return 0;
}
```

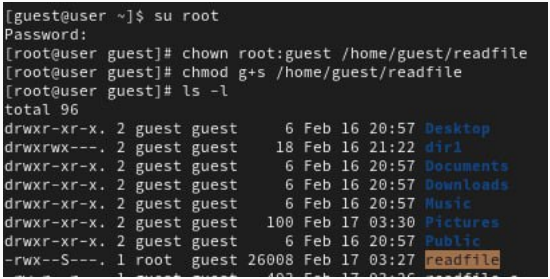
Компиляция программы

С помощью команды “gcc simpleid -o simpleid” мы превращаем наш файл в готовую программу. Если в файле будут синтаксические ошибки,то программа полностью не запустится и не скомпилируется. То есть это своеобразная проверка на синтаксис языка.



Смена владельца файла

Так как мы создавали программу, используя аккаунт “guest”, то и владельцем файла является “guest”. Но с помощью команды “chown root:guest имя_файла” мы меняем владельца файла. Теперь файл принадлежит root пользователя. Так же с помощью команды “chmod g+s” мы сделали Setgid-бит. То есть все файлы и директории, который будут созданы, будут наследовать идентификаторы группы каталога, а не идентификатор группы пользователя, который создал файл.



Запуск программы

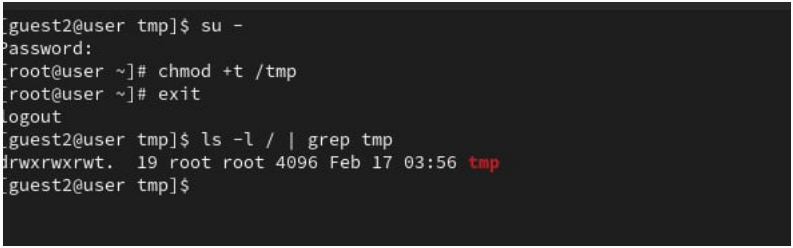
Из-за того, что мы сделали владельцем файла root, то и работать с файлом мы можем только с аккаунта root. Если мы попытаемся запустить программу с аккаунт “guest” или какого-то другого, то у нас просто вылезет ошибка, что нет доступа.

```
[guest@user ~]$ ./readfile readjile.c
bash: ./readfile: Permission denied
[guest@user ~]$ ./readfile /etc/shadow
bash: ./readfile: Permission denied
[guest@user ~]$
```

Исследование Sticky-бита

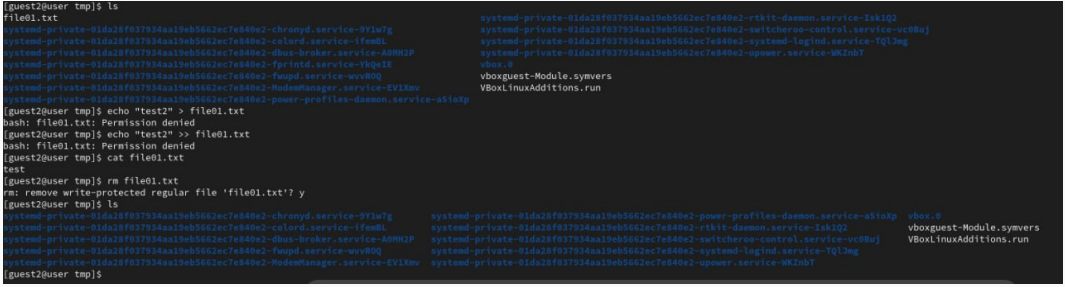
Проверка бита

Чтобы определить есть ли у директории sticky-бит (t) или нет, нам необходимо ввести команду “ls -l”” | grep имя_директории”. Если в конце стоит “t”, то бит включён, а если нет, то мы можем его включить с помощью команды “chmod +t имя_файла”. При этом необходимо быть суперпользователем.



Возможности Sticky-бита

Такой бит нужен для удаления файлов. А конкретно, если бит включен, то пользователи не могут удалять файл. Если же бит выключен, то любой пользователь может удалить файл. Нет смысла использовать этот бит к простым файлам, только к директориям. Убрать или поставить Sticky-бит мы можем только используя аккаунт суперпользователя. Но если мы находимся на аккаунте root и захотим удалить файл, который использует такой бит, то мы не сможем это сделать.



Важное примечание

Важное примечание

В ходе лабораторной работы я проверял работу бита с помощью аккаунт “guest2”, который находится в группе с акаунтом “guest”. Я выдал права на категорию “все остальные” для редактирования файла. Но мой аккаунт находился в категории “группа владельца” и на него эти права не распространялись. То есть я не смог проверить, можно ли редактировать файл, используя бит. Но в данных из интернета написано, что Sticky-бит не влияет на возможность редактирования файла, а только на удаление его.

Результаты

Вывод

Я изучил механизмы изменения идентификатора, применил SetUID-бит и Sticky-бит. Получил практические навыки работы в консоли с дополнительными атрибутами. Рассмотрел работы механизма смены идентификатора процессов пользователя, а так же влияние бита Sticky на запись и удаление файлов.