

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

- 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

- 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++.

```
1 #include <sys/types.h>
2 #include <unistd.h>
3 #include <stdio.h>
4
5 int
6 main ()
7 {
8     uid_t real_uid = getuid();
9     uid_t e_uid = geteuid ();
10
11     gid_t real_gid = getgid ();
12     gid_t e_gid = getegid ();
13
14     printf ("e_uid=%d, e_gid=%d\n", e_uid, e_gid);
15     printf ("real_uid=%d, real_gid=%d\n", real_uid, real_gid);
16
17     return 0;
18 }
```


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.

```
[guest@user ~]$ gcc simpleid2.c -o simpleid2
[guest@user ~]$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
[guest@user ~]$
```

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 пользователю. 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.

```
[guest@user ~]$ su root
Password:
[root@user guest]# chown root:guest /home/guest/readfile
[root@user guest]# chmod g+s /home/guest/readfile
[root@user guest]# ls -l
total 96
drwxr-xr-x. 2 guest guest    6 Feb 16 20:57 Desktop
drwxrwx---. 2 guest guest   18 Feb 16 21:22 dir1
drwxr-xr-x. 2 guest guest    6 Feb 16 20:57 Documents
drwxr-xr-x. 2 guest guest    6 Feb 16 20:57 Downloads
drwxr-xr-x. 2 guest guest    6 Feb 16 20:57 Music
drwxr-xr-x. 2 guest guest  100 Feb 17 03:30 Pictures
drwxr-xr-x. 2 guest guest    6 Feb 16 20:57 Public
-rwx--S---. 1 root  guest 26008 Feb 17 03:27 readfile
```

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

Исследование Sticky-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.

```
[guest2@user tmp]$ su -  
Password:  
[root@user ~]# chmod +t /tmp  
[root@user ~]# exit  
logout  
[guest2@user tmp]$ ls -l / | grep tmp  
drwxrwxrwt. 19 root root 4096 Feb 17 03:56 tmp  
[guest2@user tmp]$
```

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.

```
[guest2@user tmp]$ ls
file01.txt
systemd-private-01da28f037934aa19eb5662ec7e840e2-chronyd.service-9Ylw7g
systemd-private-01da28f037934aa19eb5662ec7e840e2-colord.service-ifemBL
systemd-private-01da28f037934aa19eb5662ec7e840e2-dbus-broker.service-A0MH2P
systemd-private-01da28f037934aa19eb5662ec7e840e2-fprintd.service-YkQeIE
systemd-private-01da28f037934aa19eb5662ec7e840e2-fuupd.service-wvvR0Q
systemd-private-01da28f037934aa19eb5662ec7e840e2-ModemManager.service-EVIXmv
systemd-private-01da28f037934aa19eb5662ec7e840e2-power-profiles-daemon.service-a5IoXp
[guest2@user tmp]$ echo "test2" > file01.txt
bash: file01.txt: Permission denied
[guest2@user tmp]$ echo "test2" >> file01.txt
bash: file01.txt: Permission denied
[guest2@user tmp]$ cat file01.txt
test
[guest2@user tmp]$ rm file01.txt
rm: remove write-protected regular file 'file01.txt'? y
[guest2@user tmp]$ ls
systemd-private-01da28f037934aa19eb5662ec7e840e2-chronyd.service-9Ylw7g
systemd-private-01da28f037934aa19eb5662ec7e840e2-colord.service-ifemBL
systemd-private-01da28f037934aa19eb5662ec7e840e2-dbus-broker.service-A0MH2P
systemd-private-01da28f037934aa19eb5662ec7e840e2-fuupd.service-wvvR0Q
systemd-private-01da28f037934aa19eb5662ec7e840e2-ModemManager.service-EVIXmv
systemd-private-01da28f037934aa19eb5662ec7e840e2-power-profiles-daemon.service-a5IoXp
systemd-private-01da28f037934aa19eb5662ec7e840e2-rtkit-daemon.service-Isk1Q2
systemd-private-01da28f037934aa19eb5662ec7e840e2-switcheroo-control.service-vc08Uj
systemd-private-01da28f037934aa19eb5662ec7e840e2-systemd-logind.service-TQl3mg
systemd-private-01da28f037934aa19eb5662ec7e840e2-upower.service-WKZnbT
vbox.8
vboxguest-Module.symvers
VBoxLinuxAdditions.run
```

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

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.