

Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

Егор Викторов

1 октября, 2023, Москва, Россия

Российский Университет Дружбы Народов

Цели и задачи

- SUID - разрешение на установку идентификатора пользователя. Это бит разрешения, который позволяет пользователю запускать исполняемый файл с правами владельца этого файла.
- SGID - разрешение на установку идентификатора группы. Принцип работы очень похож на SUID с отличием, что файл будет запускаться пользователем от имени группы, которая владеет файлом.

Цель лабораторной работы

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.

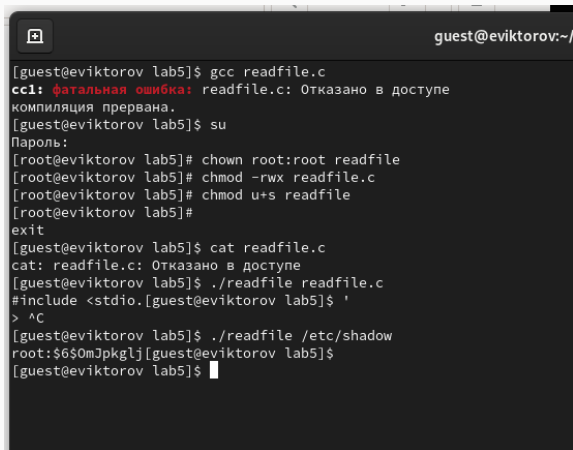
Выполнение лабораторной работы

Программа simpleid

```
[guest@eviktorov ~]$ mkdir lab5
[guest@eviktorov ~]$ cd lab5/
[guest@eviktorov lab5]$ touch simpleid.c
[guest@eviktorov lab5]$ gcc -c simpleid.c
[guest@eviktorov lab5]$ gcc -c simpleid.c -o simpleid
[guest@eviktorov lab5]$ ./simpleid
bash: ./simpleid: Отказано в доступе
[guest@eviktorov lab5]$ gcc -c simpleid.c -o simpleid
[guest@eviktorov lab5]$ ./simpleid
bash: ./simpleid: Отказано в доступе
[guest@eviktorov lab5]$ ls -l
иторо 12
-rw-r--r--. 1 guest guest 1648 окт  1 19:34 simpleid
-rw-r--r--. 1 guest guest 171 окт  1 19:32 simpleid.c
-rw-r--r--. 1 guest guest 1648 окт  1 19:33 simpleid.o
[guest@eviktorov lab5]$ chmod 777 simpleid
[guest@eviktorov lab5]$ ./simpleid
bash: ./simpleid: не удастся запустить бинарный файл: Ошибка формата выполняемого файла
[guest@eviktorov lab5]$ gcc simpleid.c -o simpleid
[guest@eviktorov lab5]$ ./simpleid
uid=1001, gid=1001
[guest@eviktorov lab5]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest),10(wheel) контекст=unconfined_u:uncon
c1023
[guest@eviktorov lab5]$
```

Figure 1: результат программы simpleid

Программа readfile



```
guest@eviktorov:~/lab5$ gcc readfile.c
cc1: фатальная ошибка: readfile.c: Отказано в доступе
компиляция прервана.
[guest@eviktorov lab5]$ su
Пароль:
[root@eviktorov lab5]# chown root:root readfile
[root@eviktorov lab5]# chmod -rwx readfile.c
[root@eviktorov lab5]# chmod u+s readfile
[root@eviktorov lab5]#
exit
[guest@eviktorov lab5]$ cat readfile.c
cat: readfile.c: Отказано в доступе
[guest@eviktorov lab5]$ ./readfile readfile.c
#include <stdio.h>
[guest@eviktorov lab5]$ ^C
[guest@eviktorov lab5]$ ./readfile /etc/shadow
root:$6$0mJpkglj[guest@eviktorov lab5]$
[guest@eviktorov lab5]$
```

Figure 3: результат программы readfile

Исследование Sticky-бита

```
[guest@eviktorov lab5]$  
[guest@eviktorov lab5]$ cd /tmp  
[guest@eviktorov tmp]$ echo test >> file01.txt  
[guest@eviktorov tmp]$ su guest2  
Пароль:  
[guest2@eviktorov tmp]$ cat file01.txt  
test  
[guest2@eviktorov tmp]$ echo test >> file01.txt  
bash: file01.txt: Отказано в доступе  
[guest2@eviktorov tmp]$  
exit  
[guest@eviktorov tmp]$ chmod 777 file01.txt  
[guest@eviktorov tmp]$ su guest2  
Пароль:  
[guest2@eviktorov tmp]$ echo test >> file01.txt  
[guest2@eviktorov tmp]$ echo test > file01.txt  
[guest2@eviktorov tmp]$ rm file01.txt  
rm: невозможно удалить 'file01.txt': Операция не позволена  
[guest2@eviktorov tmp]$ su  
Пароль:  
[root@eviktorov tmp]# chmod -t /tmp  
[root@eviktorov tmp]#  
exit  
[guest2@eviktorov tmp]$ rm file01.txt  
[guest2@eviktorov tmp]$  
[guest2@eviktorov tmp]$
```

Figure 4: исследование Sticky-бита

Выводы

Изучили механизмы изменения идентификаторов, применения SetUID- и Sticky-битов. Получили практические навыки работы в консоли с дополнительными атрибутами. Также мы рассмотрели работу механизма смены идентификатора процессов пользователей и влияние бита Sticky на запись и удаление файлов.