

Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

Александр Бровкин

24 сентября, 2024, 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@abrovkin ~]$  
[guest@abrovkin ~]$ cd  
[guest@abrovkin ~]$ mkdir lab5  
[guest@abrovkin ~]$ cd lab5/  
[guest@abrovkin lab5]$ touch simpleid.c  
[guest@abrovkin lab5]$ gedit simpleid.c  
[guest@abrovkin lab5]$ gcc simpleid.c  
[guest@abrovkin lab5]$ gcc simpleid.c -o simpleid  
[guest@abrovkin lab5]$ ./simpleid  
uid=1001, gid=1001  
[guest@abrovkin lab5]$ id  
uid=1001(guest) gid=1001(guest) rpyны=1001(guest),10(wheel) контекст=unconfined_u:unconfined_r:unconfined_t  
:s0-s0:c0.c1023  
[guest@abrovkin lab5]$
```

Figure 1: результат программы simpleid

Программа simpleid2

```
[guest@abrovkin lab5]$  
[guest@abrovkin lab5]$ touch simpleid2.c  
[guest@abrovkin lab5]$ gedit simpleid2.c  
[guest@abrovkin lab5]$ gcc simpleid2.c  
[guest@abrovkin lab5]$ gcc simpleid2.c -o simpleid2  
[guest@abrovkin lab5]$ ./simpleid2  
e_uid=1001, e_gid=1001  
real_uid=1001, real_gid=1001  
[guest@abrovkin lab5]$ su  
Napons:  
[root@abrovkin lab5]# chown root:guest simpleid2  
[root@abrovkin lab5]# chmod u+s simpleid2  
[root@abrovkin lab5]# ./simpleid2  
e_uid=0, e_gid=0  
real_uid=0, real_gid=0  
[root@abrovkin lab5]# id  
uid=0(root) gid=0(root) rpnmu=0(root) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023  
[root@abrovkin lab5]# chmod g+s simpleid2  
[root@abrovkin lab5]# ./simpleid2  
e_uid=0, e_gid=1001  
real_uid=0, real_gid=0  
[root@abrovkin lab5]#  
exit  
[guest@abrovkin lab5]$ ./simpleid2  
e_uid=0, e_gid=1001  
real_uid=1001, real_gid=1001  
[guest@abrovkin lab5]$
```

Figure 2: результат программы simpleid2

Программа readfile

```
[guest@abrovkin lab5]$ touch readfile.c
[guest@abrovkin lab5]$ gcc readfile.c
/usr/bin/ld: /usr/lib/gcc/x86_64-redhat-linux/11/../../../../lib64/crt1.o: в функции «_start»:
(.text+0x1b): неопределённая ссылка на «main»
collect2: ошибка: выполнение ld завершилось с кодом возврата 1
[guest@abrovkin lab5]$ gedit readfile.c
[guest@abrovkin lab5]$ gcc readfile.c
readfile.c: В функции «main»:
readfile.c:20:19: предупреждение: сравнение указателя и целого
   20 | while (bytes_read == (buffer));
      |                   ^~
[guest@abrovkin lab5]$ gcc readfile.c -o readfile
readfile.c: В функции «main»:
readfile.c:20:19: предупреждение: сравнение указателя и целого
   20 | while (bytes_read == (buffer));
      |                   ^~
[guest@abrovkin lab5]$ su
Пароль:
[root@abrovkin lab5]# chown root:root readfile
[root@abrovkin lab5]# chmod -rwx readfile.c
[root@abrovkin lab5]# chmod u+s readfile
[root@abrovkin lab5]#
exit
[guest@abrovkin lab5]$ cat readfile.c
cat: readfile.c: Отказано в доступе
[guest@abrovkin lab5]$ ./readfile readfile.c
#include <stdio.h>[guest@abrovkin lab5]$
[guest@abrovkin lab5]$ ./readfile /etc/shadow
root:$6$0mJpkg1j[guest@abrovkin lab5]$
[guest@abrovkin lab5]$
```

Figure 3: результат программы readfile

Исследование Sticky-бита

```
[guest@abrovkin lab5]$  
[guest@abrovkin lab5]$ echo test >> /tmp/file01.txt  
[guest@abrovkin lab5]$ chmod g-rwx /tmp/file01.txt  
[guest@abrovkin lab5]$ su gveest2  
su: user gveest2 does not exist or the user entry does not contain all the required fields  
[guest@abrovkin lab5]$ su guest2  
Пароль:  
[guest2@abrovkin lab5]$ cd /tmp  
[guest2@abrovkin tmp]$ cat file01.txt  
test  
[guest2@abrovkin tmp]$ echo test2 >> /tmp/file01.txt  
[guest2@abrovkin tmp]$ cat file01.txt  
test  
test2  
[guest2@abrovkin tmp]$ echo test3 > file01.txt  
[guest2@abrovkin tmp]$ rm file01.txt  
rm: невозможно удалить 'file01.txt': Операция не позволена  
[guest2@abrovkin tmp]$ su  
Пароль:  
[root@abrovkin tmp]# chmod -t /tmp  
[root@abrovkin tmp]#  
exit  
[guest2@abrovkin tmp]$ rm file01.txt  
[guest2@abrovkin 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.