

Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

Плето Плето Мбамби НФИбд-05-22

27 февраля, 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

```
[root@pletho guest]#  
exit  
[guest@pletho ~]$ cd  
[guest@pletho ~]$ mkdir lab5  
[guest@pletho ~]$ cd lab5/  
[guest@pletho lab5]$ toucg simpleid.c  
bash: toucg: command not found...  
[guest@pletho lab5]$ touch simpleid.c  
[guest@pletho lab5]$ gedit simpleid.c  
[guest@pletho lab5]$ gcc simpleid.c  
[guest@pletho lab5]$ gcc simpleid.c -o simpleid  
[guest@pletho lab5]$ ./simpleid  
uid=1001, gid=1001  
[guest@pletho lab5]$ id  
uid=1001(guest) gid=1001(guest) rpynmw=1001(guest),10(wheel) контекст=unconfined_u:unconfined  
_r:unconfined_t:s0-s0:c0.c1023  
[guest@pletho lab5]$
```

Figure 1: результат программы simpleid

Программа simpleid2

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

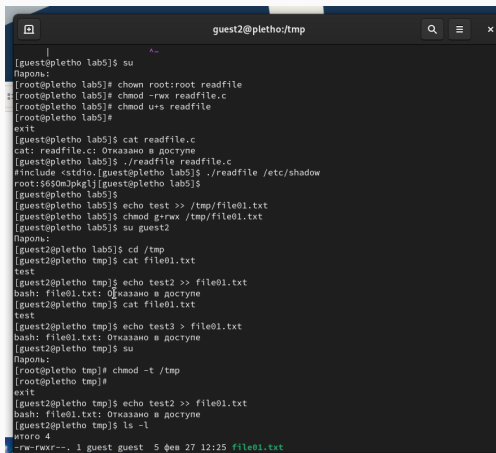
Figure 2: результат программы simpleid2

Программа readfile

```
[guest@pletho lab5]$  
[guest@pletho lab5]$ touch readfile.c  
[guest@pletho lab5]$ gedit readfile.c  
[guest@pletho lab5]$ gcc readfile.c  
readfile.c: В функции «main»:  
readfile.c:20:19: предупреждение: сравнение указателя и целого  
20 | while (bytes_read == (buffer));  
    |                   ^~  
[guest@pletho 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@pletho lab5]$ su  
Пароль:  
[root@pletho lab5]# chown root:root readfile  
[root@pletho lab5]# chmod -rwx readfile.c  
[root@pletho lab5]# chmod u+s readfile  
[root@pletho lab5]#  
exit  
[guest@pletho lab5]$ cat readfile.c  
cat: readfile.c: Отказано в доступе  
[guest@pletho lab5]$ ./readfile readfile.c  
#include <stdio.h>[guest@pletho lab5]$ ./readfile /etc/shadow  
root:$6$0mJpkg1j[guest@pletho lab5]$
```

Figure 3: результат программы readfile

Исследование Sticky-бита



```
guest2@pletho/tmp
[guest@pletho lab5]$ su
Пароль:
[root@pletho lab5]# chown root:root readfile
[root@pletho lab5]# chmod -rwx readfile.c
[root@pletho lab5]# chmod u+s readfile
[root@pletho lab5]#
exit
[guest@pletho lab5]$ cat readfile.c
cat: readfile.c: Отказано в доступе
[guest@pletho lab5]$ ./readfile readfile.c
#include <stdio.h>[guest@pletho lab5]$ ./readfile /etc/shadow
root:$6$0mJpkg1j[guest@pletho lab5]$
[guest@pletho lab5]$
[guest@pletho lab5]$ echo test >> /tmp/file01.txt
[guest@pletho lab5]$ chmod g-rwx /tmp/file01.txt
[guest@pletho lab5]$ su guest2
Пароль:
[guest2@pletho lab5]$ cd /tmp
[guest2@pletho tmp]$ cat file01.txt
test
[guest2@pletho tmp]$ echo test2 >> file01.txt
bash: file01.txt: Отказано в доступе
[guest2@pletho tmp]$ cat file01.txt
test
[guest2@pletho tmp]$ echo test3 > file01.txt
bash: file01.txt: Отказано в доступе
[guest2@pletho tmp]$ su
Пароль:
[root@pletho tmp]# chmod -t /tmp
[root@pletho tmp]#
exit
[guest2@pletho tmp]$ echo test2 >> file01.txt
bash: file01.txt: Отказано в доступе
[guest2@pletho tmp]$ ls -l
итого 4
-rw-rwxr--. 1 guest guest 5 фев 27 12:25 file01.txt
```

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.