

Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

Файсал Ахмад

25 марта, 2025, 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@faisalanhmad:~$  
guest@faisalanhmad:~$ mkdir lab5  
guest@faisalanhmad:~$ cd lab5  
guest@faisalanhmad:~/lab5$ touch simpleid.c  
guest@faisalanhmad:~/lab5$  
guest@faisalanhmad:~/lab5$ gcc simpleid.c  
guest@faisalanhmad:~/lab5$ gcc simpleid.c -o simpleid  
guest@faisalanhmad:~/lab5$ ./simpleid  
uid=1001, gid=1001  
guest@faisalanhmad:~/lab5$ id3  
bash: id3: команда не найдена...  
guest@faisalanhmad:~/lab5$ id  
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfined_r:sandbox_t:s0  
23  
guest@faisalanhmad:~/lab5$ █
```

Рис. 1: результат программы simpleid

Программа simpleid2

```
guest@faisalahmad:~/lab5$
guest@faisalahmad:~/lab5$ touch simpleid2.c
guest@faisalahmad:~/lab5$ gcc simpleid2.c
guest@faisalahmad:~/lab5$ gcc simpleid2.c -o simpleid2
guest@faisalahmad:~/lab5$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
guest@faisalahmad:~/lab5$ su
Пароль:
root@faisalahmad:/home/guest/lab5# chown root:guest simpleid2
root@faisalahmad:/home/guest/lab5# chmod u+s simpleid2
root@faisalahmad:/home/guest/lab5#
root@faisalahmad:/home/guest/lab5# ./simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
root@faisalahmad:/home/guest/lab5# id
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
root@faisalahmad:/home/guest/lab5# chmod g+s simpleid2
root@faisalahmad:/home/guest/lab5# ./simpleid2
e_uid=0, e_gid=1001
real_uid=0, real_gid=0
root@faisalahmad:/home/guest/lab5#
exit
guest@faisalahmad:~/lab5$ ./simpleid2
e_uid=0, e_gid=1001
real_uid=1001, real_gid=1001
guest@faisalahmad:~/lab5$ █
```

Рис. 2: результат программы simpleid2

Программа readfile

```
guest@faisalanhmad:~/lab5$  
guest@faisalanhmad:~/lab5$ gcc readfile.c  
readfile.c: В функции «main»:  
readfile.c:20:19: предупреждение: сравнение указателя и целого  
20 | while (bytes_read == (buffer));  
    |                      ^~  
guest@faisalanhmad:~/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@faisalanhmad:~/lab5$ su  
Пароль:  
root@faisalanhmad:/home/guest/lab5# chown root:root readfile  
root@faisalanhmad:/home/guest/lab5# chmod -rwx readfile.c  
root@faisalanhmad:/home/guest/lab5# chmod u+s readfile  
root@faisalanhmad:/home/guest/lab5#  
exit  
guest@faisalanhmad:~/lab5$ cat readfile.c  
cat: readfile.c: Отказано в доступе  
guest@faisalanhmad:~/lab5$ ./readfile readfile.c  
#include <stdio.h> guest@fais./readfile /etc/shadowfile /etc/shadow  
root:$y$j9T$zLZFguest@faisalanhmad:~/lab5$  
guest@faisalanhmad:~/lab5$ █
```

Рис. 3: результат программы readfile

Исследование Sticky-бита

```
guest@faisalrahmad:~/lab03$
guest@faisalrahmad:~/lab5$ echo "test" >> /tmp/file01.txt
guest@faisalrahmad:~/lab5$ chmod g+rxw /tmp/file01.txt
guest@faisalrahmad:~/lab5$ su guest2
Пароль:
guest2@faisalrahmad:/home/guest/lab5$
guest2@faisalrahmad:/home/guest/lab5$ cd /tmp
cat file01.txt
test
test
test
guest2@faisalrahmad:/tmp$
^[[200~echo "test2" >> /tmp/file01.txt
^[[201~guest2@faisalrahmad:echo "test2" >> /tmp/file01.txt
guest2@faisalrahmad:/tmp$ cat file01.txt
test
test
test2
guest2@faisalrahmad:/tmp$ echo "test3" > /tmp/file01.txt
guest2@faisalrahmad:/tmp$ rm file01.txt
rm: невозможно удалить 'file01.txt': Операция не позволена
guest2@faisalrahmad:/tmp$ su
Пароль:
root@faisalrahmad:/tmp# chmod -t /tmp
root@faisalrahmad:/tmp#
exit
guest2@faisalrahmad:/tmp$
^[[200~echo "test2" >> /tmp/file01.txt
^[[201~guest2@faisalrahmad:echo "test2" >> /tmp/file01.txt
guest2@faisalrahmad:/tmp$ rm file01.txt
guest2@faisalrahmad:/tmp$
```

Рис. 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.