

Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

Елизавета Курникова

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

```
exit
guest@ekurnikova:~$ mkdir lab5
guest@ekurnikova:~$ cd lab5
guest@ekurnikova:~/lab5$ touch simpleid.c
guest@ekurnikova:~/lab5$ gcc simpleid.c
guest@ekurnikova:~/lab5$ gcc simpleid.c -o simpleid
guest@ekurnikova:~/lab5$ ./simpleid
uid=1001, gid=1001
guest@ekurnikova:~/lab5$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfined_r:unconfined_t:~:lab5
0-s0:c0.c1023
guest@ekurnikova:~/lab5$ █
```

Рис. 1: результат программы simpleid

Программа simpleid2

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

Рис. 2: результат программы simpleid2

Программа readfile

```
root@ekurnikova:/home/guest/lab5# chmod root.root readfile
root@ekurnikova:/home/guest/lab5# chmod -rwx readfile.c
root@ekurnikova:/home/guest/lab5# chmod u+s readfile
root@ekurnikova:/home/guest/lab5# cat readfile.c
#include <stdio.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>
#include <fcntl.h>

int main(int argc, char* argv[])
{
    unsigned char buffer[16];
    size_t bytes_read;
    int i;

    int fd=open(argv[1], O_RDONLY);
    do
    {
        bytes_read=read(fd, buffer, sizeof(buffer));
        for (i=0; i<bytes_read; ++i)
            printf("%c", buffer[i]);
    }
    while (bytes_read == (buffer));
    close (fd);
    return 0;
}
root@ekurnikova:/home/guest/lab5# ./readfile readfile.c
#include <stdio.h>root@ekurnikova:/home/guest/lab5#
root@ekurnikova:/home/guest/lab5# ./readfile /etc/shadow
root:5y$j9T$zLZFroot@ekurnikova:/home/guest/lab5#
root@ekurnikova:/home/guest/lab5# █
```

Рис. 3: результат программы readfile

Исследование Sticky-бита

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