

Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

Косолапов Степан ¹

7 октября, 2023, Москва, Россия

¹Российский Университет Дружбы Народов

Цели и задачи 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.

Процесс выполнения лабораторной работы



```
guest@sekosolapov:~/dir2
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>

int
main()
{
    uid_t uid = geteuid();
    gid_t gid = getegid();
    printf("uid=%d, gid=%d\n", uid, gid);
    return 0;
}
```

Рис. 1: simpleid.c

```
[guest@sekosolapov dir2]$ ./simpleid
uid=1001, gid=1001
[guest@sekosolapov dir2]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@sekosolapov dir2]$
```

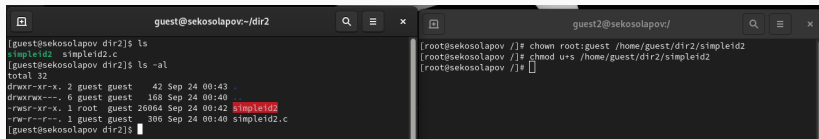
Рис. 2: сравнение вывода simpleid.c и id



```
guest@sekosolapov:~/dir2
[guest@sekosolapov dir2]$ gcc simpleid2.c -o simpleid2; ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
[guest@sekosolapov dir2]$ s
```

Рис. 3: компиляция и запуск simpleid2.c

Установка setuid бита



The image shows two terminal windows side-by-side. The left window is titled 'guest@sekosolapov:~/dir2' and shows the following commands and output:

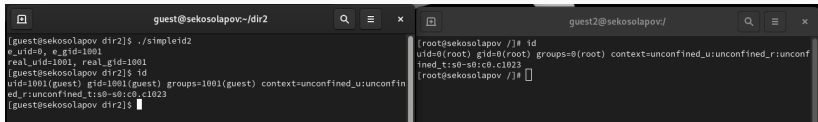
```
[guest@sekosolapov dir2]$ ls
simpleid2  simpleid2.c
[guest@sekosolapov dir2]$ ls -al
total 32
drwxr-xr-x. 2 guest guest  42 Sep 24 00:43 .
drwxrwx---. 6 guest guest 168 Sep 24 00:40 ..
-rwsr-xr-x. 1 root  guest 26064 Sep 24 00:42 simpleid2
-rw-r--r--. 1 guest guest  306 Sep 24 00:40 simpleid2.c
[guest@sekosolapov dir2]$
```

The right window is titled 'guest2@sekosolapov:/' and shows the following commands and output:

```
[root@sekosolapov /]# chown root:guest /home/guest/dir2/simpleid2
[root@sekosolapov /]# chmod u+s /home/guest/dir2/simpleid2
[root@sekosolapov /]#
```

Рис. 4: устанавливаем владельца файла simpleid2 и добавляем setuid бит

Выполнение файла simpleid2 с битом setuid



The image shows two terminal windows side-by-side. The left window is titled 'guest@sekosolapov:~/dir2' and shows the execution of the 'simpleid2' program. The output indicates that the program is running with effective user ID 0 (root). The right window is titled 'guest2@sekosolapov:/' and shows the output of the 'id' command, confirming that the user is now root (uid=0, gid=0, groups=0).

```
guest@sekosolapov:~/dir2$ ./simpleid2
e_uid=0, e_gid=1001
real_uid=1001, real_gid=1001
[guest@sekosolapov dir2]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@sekosolapov dir2]$
```

```
[root@sekosolapov /]# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@sekosolapov /]#
```

Рис. 5: выполняем simpleid2 с битом setuid

```
guest@sekosolapov:~/dir2
#include <fcntl.h>
#include <stdio.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>

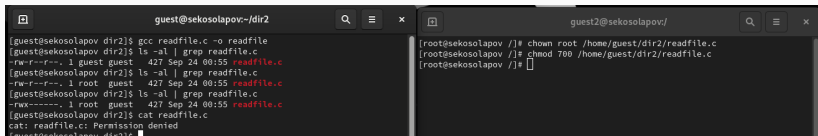
int
main(int argc, char* argv[])
{
    unsigned char buffer[16];
    size_t bytes_read;
    int i;

    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 == sizeof(buffer));
    close(fd);

    return 0;
}
```

Установка прав на файл readfile.c только для root



The image shows two terminal windows side-by-side. The left window is titled 'guest@sekosolapov:~/dir2' and shows the following commands and output:

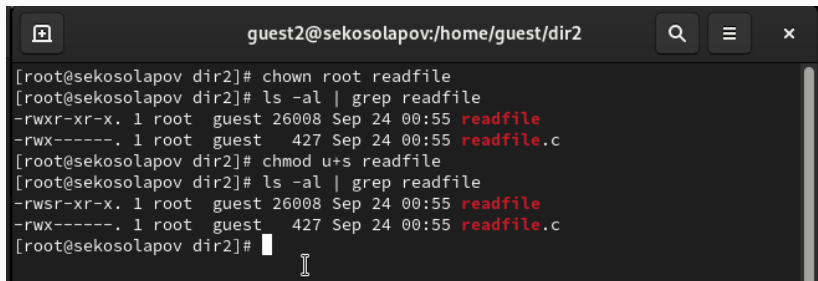
```
guest@sekosolapov dir2]$ gcc readfile.c -o readfile
guest@sekosolapov dir2]$ ls -al | grep readfile.c
-rw-r--r--. 1 guest guest 427 Sep 24 00:55 readfile.c
guest@sekosolapov dir2]$ ls -al | grep readfile.c
-rw-r--r--. 1 root guest 427 Sep 24 00:55 readfile.c
guest@sekosolapov dir2]$ ls -al | grep readfile.c
-rwx-----. 1 root guest 427 Sep 24 00:55 readfile.c
guest@sekosolapov dir2]$ cat readfile.c
cat: readfile.c: Permission denied
guest@sekosolapov dir2]$
```

The right window is titled 'guest2@sekosolapov:/' and shows the following commands and output:

```
root@sekosolapov /]$ chown root /home/guest/dir2/readfile.c
root@sekosolapov /]$ chmod 700 /home/guest/dir2/readfile.c
root@sekosolapov /]$
```

Рис. 7: установка прав на файл readfile.c только для root

Установка setuid бита на файл readfile

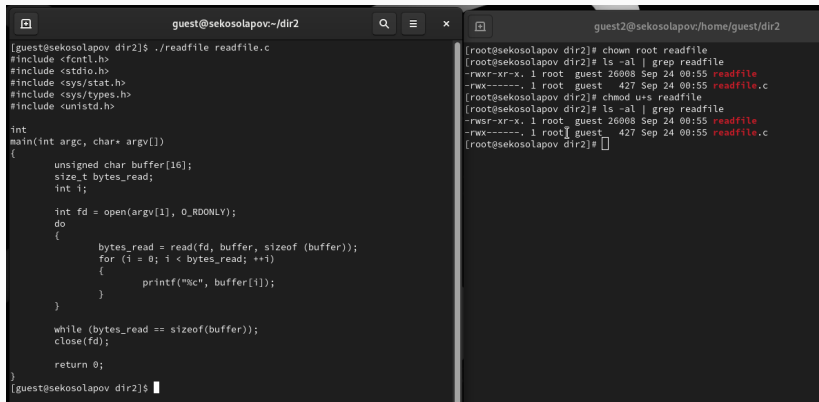


```
guest2@sekosolapov:/home/guest/dir2

[root@sekosolapov dir2]# chown root readfile
[root@sekosolapov dir2]# ls -al | grep readfile
-rwxr-xr-x. 1 root  guest 26008 Sep 24 00:55 readfile
-rwx-----. 1 root  guest  427 Sep 24 00:55 readfile.c
[root@sekosolapov dir2]# chmod u+s readfile
[root@sekosolapov dir2]# ls -al | grep readfile
-rwsr-xr-x. 1 root  guest 26008 Sep 24 00:55 readfile
-rwx-----. 1 root  guest  427 Sep 24 00:55 readfile.c
[root@sekosolapov dir2]#
```

Рис. 8: установка setuid бита на файл readfile

чтение readfile.c с помощью readfile с установленным setuid битом



The image shows two terminal windows side-by-side. The left window is titled 'guest@sekosolapov:~/dir2' and shows the source code of a C program named 'readfile.c'. The code includes headers for file operations, standard I/O, system calls, and types. It defines a 'main' function that opens a file specified as an argument in read-only mode, reads it in chunks of 16 bytes, and prints each byte. The right window is titled 'guest2@sekosolapov:/home/guest/dir2' and shows the execution of the program. It first runs 'chown root readfile' to change ownership to root. Then it runs 'ls -al | grep readfile' showing permissions '-rwxr-xr-x' for 'readfile'. Next, it runs 'chmod u+s readfile' to set the setuid bit, which changes the permissions to '-rwsr-xr-x'. A second 'ls' command confirms this. Finally, the program is executed, and the prompt returns to the root user.

```
guest@sekosolapov:~/dir2
[guest@sekosolapov dir2]$ ./readfile readfile.c
#include <fcntl.h>
#include <stdio.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.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]);
        }

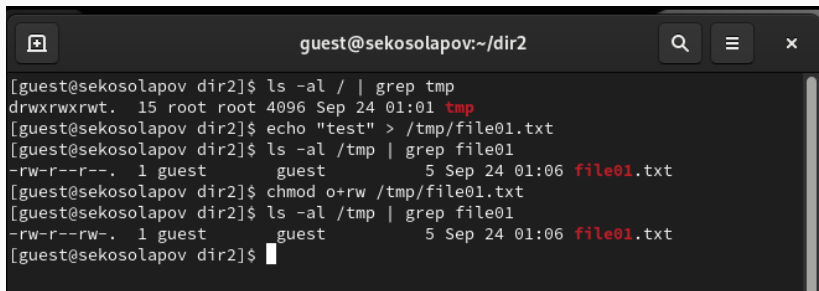
        while (bytes_read == sizeof(buffer));
        close(fd);

        return 0;
    }
[guest@sekosolapov dir2]$

guest2@sekosolapov:/home/guest/dir2
[root@sekosolapov dir2]# chown root readfile
[root@sekosolapov dir2]# ls -al | grep readfile
-rwxr-xr-x. 1 root guest 26008 Sep 24 00:55 readfile
[root@sekosolapov dir2]# chmod u+s readfile
[root@sekosolapov dir2]# ls -al | grep readfile
-rwsr-xr-x. 1 root guest 26008 Sep 24 00:55 readfile
[root@sekosolapov dir2]#
```

Рис. 9: чтение readfile.c с помощью readfile с установленным setuid битом

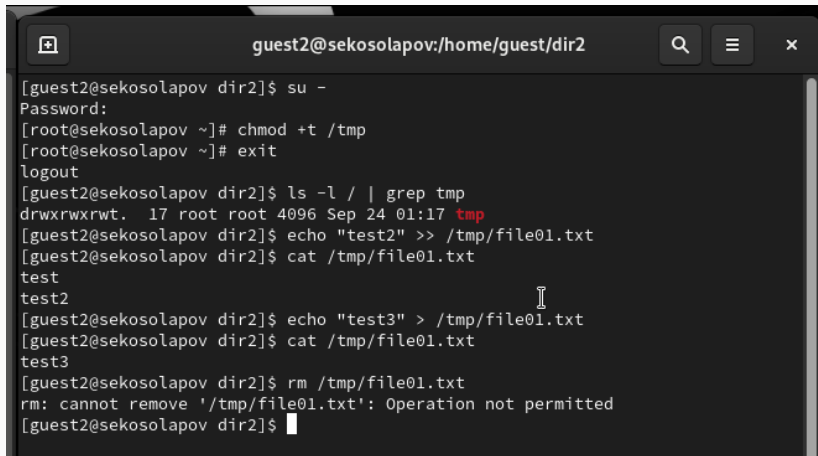
Создание файла в sticky директории



```
guest@sekosolapov:~/dir2
[guest@sekosolapov dir2]$ ls -al / | grep tmp
drwxrwxrwt. 15 root root 4096 Sep 24 01:01 tmp
[guest@sekosolapov dir2]$ echo "test" > /tmp/file01.txt
[guest@sekosolapov dir2]$ ls -al /tmp | grep file01
-rw-r--r--. 1 guest      guest          5 Sep 24 01:06 file01.txt
[guest@sekosolapov dir2]$ chmod o+rw /tmp/file01.txt
[guest@sekosolapov dir2]$ ls -al /tmp | grep file01
-rw-r--rw-. 1 guest      guest          5 Sep 24 01:06 file01.txt
[guest@sekosolapov dir2]$
```

Рис. 10: создание файла в sticky директории

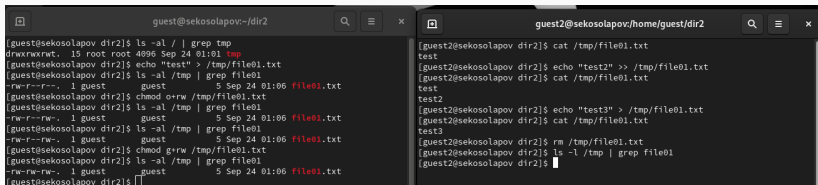
Операции над файлом в sticky директории

A terminal window titled 'guest2@sekosolapov:/home/guest/dir2' with search, menu, and close icons. The terminal shows a sequence of commands: switching to root, setting permissions on /tmp, exiting root, listing /tmp, appending 'test2' to /tmp/file01.txt, and attempting to remove it, which fails with a permission error.

```
[guest2@sekosolapov dir2]$ su -  
Password:  
[root@sekosolapov ~]# chmod +t /tmp  
[root@sekosolapov ~]# exit  
logout  
[guest2@sekosolapov dir2]$ ls -l / | grep tmp  
drwxrwxrwt. 17 root root 4096 Sep 24 01:17 tmp  
[guest2@sekosolapov dir2]$ echo "test2" >> /tmp/file01.txt  
[guest2@sekosolapov dir2]$ cat /tmp/file01.txt  
test  
test2  
[guest2@sekosolapov dir2]$ echo "test3" > /tmp/file01.txt  
[guest2@sekosolapov dir2]$ cat /tmp/file01.txt  
test3  
[guest2@sekosolapov dir2]$ rm /tmp/file01.txt  
rm: cannot remove '/tmp/file01.txt': Operation not permitted  
[guest2@sekosolapov dir2]$
```

Рис. 11: операции над файлом в sticky директории

Операции над файлом в директории без sticky



The image shows two terminal windows side-by-side. The left window is titled 'guest@sekosolapov:~/dir2' and shows a series of commands and their outputs. The right window is titled 'guest2@sekosolapov:/home/guest/dir2' and shows the continuation of the operations from the perspective of a second user.

```
guest@sekosolapov:~/dir2$ ls -al / | grep tmp
drwxrwxrwt. 15 root root 4096 Sep 24 01:01 tmp
guest@sekosolapov:~/dir2$ echo "test" > /tmp/file01.txt
guest@sekosolapov:~/dir2$ ls -al /tmp | grep file01
-rw-r--r--. 1 guest guest 5 Sep 24 01:06 file01.txt
guest@sekosolapov:~/dir2$ chmod o+rw /tmp/file01.txt
guest@sekosolapov:~/dir2$ ls -al /tmp | grep file01
-rw-r--rw-. 1 guest guest 5 Sep 24 01:06 file01.txt
guest@sekosolapov:~/dir2$ ls -al /tmp | grep file01
-rw-r--rw-. 1 guest guest 5 Sep 24 01:06 file01.txt
guest@sekosolapov:~/dir2$ chmod g+rw /tmp/file01.txt
guest@sekosolapov:~/dir2$ ls -al /tmp | grep file01
-rw-rw-rw-. 1 guest guest 5 Sep 24 01:06 file01.txt
guest@sekosolapov:~/dir2$
```

```
guest2@sekosolapov:/home/guest/dir2$ cat /tmp/file01.txt
test
guest2@sekosolapov:~/dir2$ echo "test2" >> /tmp/file01.txt
guest2@sekosolapov:~/dir2$ cat /tmp/file01.txt
test
test2
guest2@sekosolapov:~/dir2$ echo "test3" > /tmp/file01.txt
guest2@sekosolapov:~/dir2$ cat /tmp/file01.txt
test3
guest2@sekosolapov:~/dir2$ rm /tmp/file01.txt
guest2@sekosolapov:~/dir2$ ls -l /tmp | grep file01
guest2@sekosolapov:~/dir2$
```

Рис. 12: операции над файлом в директории без sticky

Выводы по проделанной работе

В данной работе мы изучили атрибуты sticky, и биты setgid и setuid и их влияние на различные аспекты работы системы.