

# Защита лабораторной работы №5 Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

---

Исаханян Эдуард Тигранович

2022 Sep 17th

RUDN University, Moscow, Russian Federation

## Защита лабораторной работы №5

---

Цель

---

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

---

```
#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;
}
```

Figure 1: Программа simpleid.c

```
[guest@edikisakhayan dir2]$ gcc simpleid.c -o simpleid
[guest@edikisakhayan dir2]$ ./simpleid
uid=1001, gid=1001
[guest@edikisakhayan dir2]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfi
ned_r:unconfined_t:s0-s0:c0.c1023
```

Figure 2: Компиляция и выполнение программы simpleid

```
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>
int
main ()
{
    uid_t real_uid = getuid ();
    uid_t e_uid = geteuid ();
    gid_t real_gid = getgid ();
    gid_t e_gid = getegid ();
    printf ("e_uid=%d, e_gid=%d\n", e_uid, e_gid);
    printf ("real_uid=%d, real_gid=%d\n", real_uid,
    ,! real_gid);
    return 0;
}
```

Figure 3: Программа simpleid2.c



```
[guest@edikisakhayan dir2]$ gcc simpleid2.c -o simpleid2  
[guest@edikisakhayan dir2]$ ./simpleid2  
e_uid=1001, e_gid=1001  
real uid=1001, real gid=1001
```

Figure 4: Компиляция и выполнение программы simpleid2

## Смена пользователя. Установка SetUID-бита. Выполнение программы simpleid2

```
[guest@edikisakhayan dir2]$ su
Пароль:
[root@edikisakhayan dir2]# chown root:guest /home/guest/dir2/simpleid2
[root@edikisakhayan dir2]# chmod u+s simpleid2
[root@edikisakhayan dir2]# ls -l simpleid2
-rwsrwxr-x. 1 root guest 8616 сен 17 19:34 simpleid2
[root@edikisakhayan dir2]# su guest
[guest@edikisakhayan dir2]$ ./simpleid2
e_uid=0, e_gid=1001
real_uid=1001, real_gid=1001
[guest@edikisakhayan dir2]$ id
uid=1001(guest) gid=1001(guest) rгруппы=1001(guest) контекст=unconfined_u:unconfi
ned r:unconfined t:s0-s0:c0.c1023
```

Figure 5: Смена пользователя. Установка SetUID-бита. Выполнение программы simpleid2

## Установка SetGID-бита. Выполнение программы simpleid2

```
[root@edikisakhayan dir2]# chmod g+s simpleid2
[root@edikisakhayan dir2]# ls -l simpleid2
-rwxrwsr-x. 1 root guest 8616 сен 17 19:34 simpleid2
[root@edikisakhayan dir2]# su guest
[guest@edikisakhayan dir2]$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
[guest@edikisakhayan dir2]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfi
ned r:unconfined t:s0-s0:c0.c1023
```

Figure 6: Установка SetGID-бита. Выполнение программы simpleid2

```
#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;
}
```

Figure 7: Программа readfile.c

```
[guest@edikisakhayan dir2]$ gcc readfile.c -o readfile
[guest@edikisakhayan dir2]$ su
Пароль:
[root@edikisakhayan dir2]# chown root:guest readfile.c
[root@edikisakhayan dir2]# chmod 700 readfile.c
[root@edikisakhayan dir2]# su guest
[guest@edikisakhayan dir2]$ cat readfile.c
cat: readfile.c: Отказано в доступе
```

Figure 8: Работа с программой readfile.c

```
[root@edikisakhayan dir2]# chown root:guest readfile  
[root@edikisakhayan dir2]# chmod u+s readfile
```

Figure 9: Установка SetUID-бита на программу readfile

```
[root@edikisakhayan 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;
}
```

Figure 10: Программа readfile читает readfile.c

## Программа readfile читает /etc/shadow

```
chrony:!!:19245::::::
unbound:!!:19245::::::
qemu:!!:19245::::::
tss:!!:19245::::::
usbmuxd:!!:19245::::::
geoclue:!!:19245::::::
gluster:!!:19245::::::
gdm:!!:19245::::::
rpcuser:!!:19245::::::
nfsnobody:!!:19245::::::
gnome-initial-setup:!!:19245::::::
sshd:!!:19245::::::
avahi:!!:19245::::::
postfix:!!:19245::::::
ntp:!!:19245::::::
tcpdump:!!:19245::::::
edikisakhanyan:$6$m8bL1QIgRCJUuyR$25jN.QZ9uYrarmAQEcWb5/TZysKzT3UdZjTALh9Bf0Tu3
OGqmUw6yvwj rPGXvkdIYHpekIjCeka/60ZFkdLwG1.:0:99999:7:::
vboxadd:!!:19245::::::
guest:$6$ErldSDDk$BYqxG.Hz3V3qA6PS3D4Fpnc9HTjbLc5SQp8awW3RHX0IV4y22eWpaty51Y07Uj
kIc45BJGCL9zGeWj7rC1pam.:19248:0:99999:7:::
guest2:$6$pkIBH0.M$5NCX7Imao06CB2Q4wFbflb1/oKnJPwhJtPUOgtLKVsXEq6aj3mflqJS4HBA#
LyBqTFqHYLqxqAwF1BKVOP.G.:19252:0:99999:7:::
```

Figure 11: Программа readfile читает /etc/shadow



## Исследование Sticky-бита

---

## Исследование Sticky-бита от имени guest

```
[root@edikisakhayan dir2]# su guest
[guest@edikisakhayan dir2]$ ls -l / | grep tmp
drwxrwxrwt. 32 root root 4096 сен 17 19:46 tmp
[guest@edikisakhayan dir2]$ echo "test" > /tmp/file01.txt
[guest@edikisakhayan dir2]$ ls -l /tmp/file01.txt
-rw-rw-r--. 1 guest guest 5 сен 17 19:48 /tmp/file01.txt
[guest@edikisakhayan dir2]$ chmod o+rw /tmp/file01.txt
[guest@edikisakhayan dir2]$ ls -l /tmp/file01.txt
-rw-rw-rw-. 1 guest guest 5 сен 17 19:48 /tmp/file01.txt
```

Figure 12: Исследование Sticky-бита от имени guest

## Работа с file01.txt от имени guest2 при наличии Sticky-бита

```
[guest@edikisakhayan ~]$ su guest2
Пароль:
[guest2@edikisakhayan guest]$ cat /tmp/file01.txt
test
[guest2@edikisakhayan guest]$ echo "test2" >> /tmp/file01.txt
[guest2@edikisakhayan guest]$ cat /tmp/file01.txt
test
test2
[guest2@edikisakhayan guest]$ echo "test3" > /tmp/file01.txt
[guest2@edikisakhayan guest]$ cat /tmp/file01.txt
test3
[guest2@edikisakhayan guest]$ rm /tmp/file01.txt
rm: невозможно удалить «/tmp/file01.txt»: Операция не позволена
```

Figure 13: Работа с file01.txt от имени guest2 при наличии Sticky-бита

```
[guest@edikisakhayan dir2]$ su
Пароль:
[root@edikisakhayan dir2]# chmod -t /tmp
[root@edikisakhayan dir2]# exit
exit
```

Figure 14: Снятие Sticky-бита с /tmp

```
[guest2@edikisakhayan guest]$ cat /tmp/file01.txt
test3
[guest2@edikisakhayan guest]$ echo "test4" >> /tmp/file01.txt
[guest2@edikisakhayan guest]$ cat /tmp/file01.txt
test3
test4
[guest2@edikisakhayan guest]$ echo "test5" > /tmp/file01.txt
[guest2@edikisakhayan guest]$ cat /tmp/file01.txt
test5
[guest2@edikisakhayan guest]$ rm /tmp/file01.txt
```

Figure 15: Работа с file01.txt от имени guest2 без Sticky-бита

```
[guest@edikisakhayan dir2]$ su
Пароль:
[root@edikisakhayan dir2]# chmod +t /tmp
[root@edikisakhayan dir2]# exit
exit
[guest@edikisakhayan dir2]$
```

Figure 16: Возвращение Sticky-бита на /tmp

## Вывод

---

В ходе работы, мы изучили механизмы изменения идентификаторов, применения SetUID- и Sticky-битов. Получили практические навыки работы в консоли с дополнительными атрибутами. Рассмотрели работу механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.