

# **Отчёт по лабораторной работе №5**

**Дискреционное разграничение прав в Linux. Исследование влияния  
дополнительных атрибутов**

Ярметов Камран НФИбд-01-18

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>5</b>
2.1	Подготовка . . . . .	5
2.2	Изучение механики SetUID . . . . .	6
2.3	Исследование Sticky-бита . . . . .	11
<b>3</b>	<b>Выводы</b>	<b>14</b>
	<b>Список литературы</b>	<b>15</b>

# List of Figures

2.1	подготовка к работе . . . . .	5
2.2	программа simpleid . . . . .	6
2.3	результат программы simpleid . . . . .	7
2.4	программа simpleid2 . . . . .	7
2.5	результат программы simpleid2 . . . . .	8
2.6	программа readfile . . . . .	9
2.7	результат программы readfile . . . . .	10
2.8	результат программы readfile . . . . .	10
2.9	исследование Sticky-бита . . . . .	13

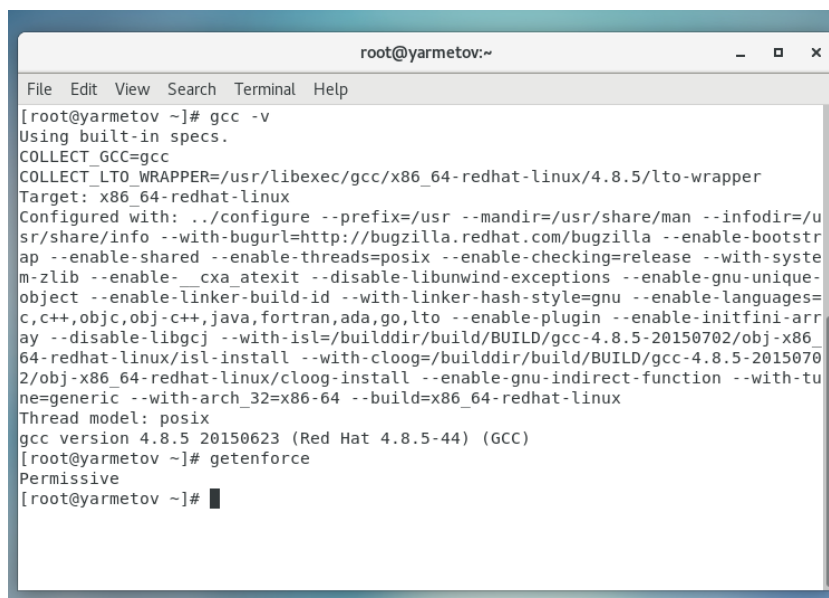
# 1 Цель работы

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.

## 2 Выполнение лабораторной работы

### 2.1 Подготовка

1. Для выполнения части заданий требуются средства разработки приложений. Проверили наличие установленного компилятора gcc командой gcc -v: компилятор обнаружен.
2. Чтобы система защиты SELinux не мешала выполнению заданий работы, отключили систему запретов до очередной перезагрузки системы командой setenforce 0:
3. Команда getenforce вывела Permissive:

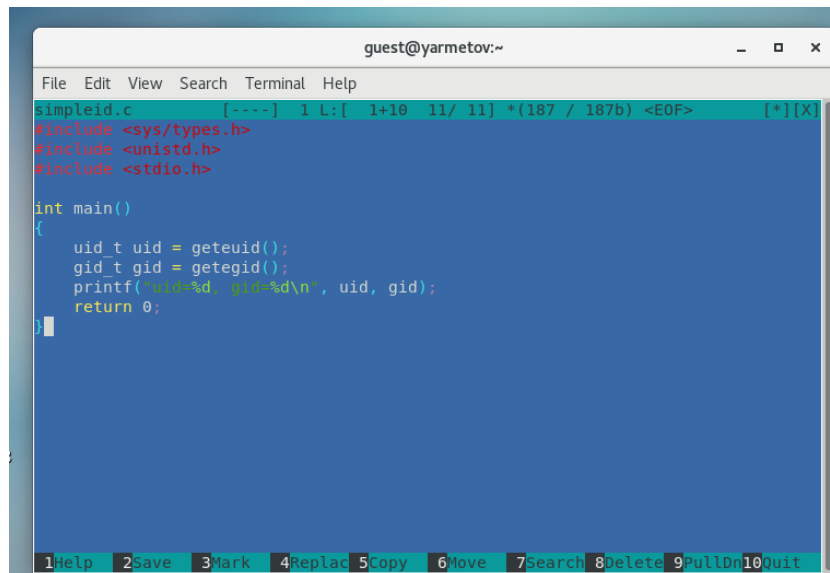


```
root@yarmetov:~  
File Edit View Search Terminal Help  
[root@yarmetov ~]# gcc -v  
Using built-in specs.  
COLLECT_GCC=gcc  
COLLECT_LTO_WRAPPER=/usr/libexec/gcc/x86_64-redhat-linux/4.8.5/lto-wrapper  
Target: x86_64-redhat-linux  
Configured with: ../configure --prefix=/usr --mandir=/usr/share/man --infodir=/usr/share/info --with-bugurl=http://bugzilla.redhat.com/bugzilla --enable-bootstrap --enable-shared --enable-threads=posix --enable-checking=release --with-system-zlib --enable-cxx-atomic --disable-libunwind-exceptions --enable-gnu-unique-object --enable-linker-build-id --with-linker-hash-style=gnu --enable-languages=c,c++,objc,obj-c++,java,fortran,ada,go,lto --enable-plugin --enable-initfini-array --disable-libgcj --with-isl=/builddir/build/BUILD/gcc-4.8.5-20150702/obj-x86_64-redhat-linux/isl-install --with-cloog=/builddir/build/BUILD/gcc-4.8.5-20150702/obj-x86_64-redhat-linux/cloog-install --enable-gnu-indirect-function --with-tune=generic --with-arch_32=x86-64 --build=x86_64-redhat-linux  
Thread model: posix  
gcc version 4.8.5 20150623 (Red Hat 4.8.5-44) (GCC)  
[root@yarmetov ~]# getenforce  
Permissive  
[root@yarmetov ~]#
```

Figure 2.1: подготовка к работе

## 2.2 Изучение механики SetUID

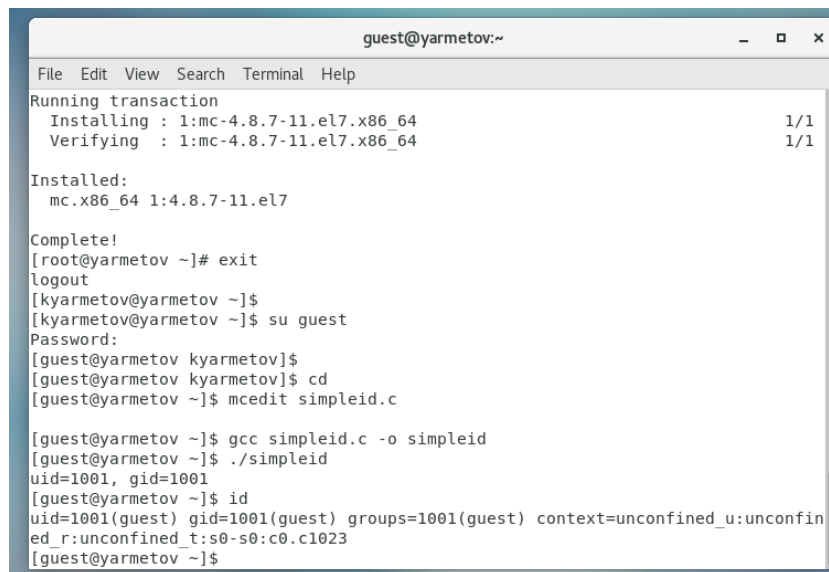
1. Вошли в систему от имени пользователя guest.
2. Написали программу simpleid.c.



```
guest@yarmetov:~  
File Edit View Search Terminal Help  
simpleid.c [----] 1 L: [ 1+10 11/ 11] *(187 / 187b) <EOF> [*][X]  
#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 2.2: программа simpleid

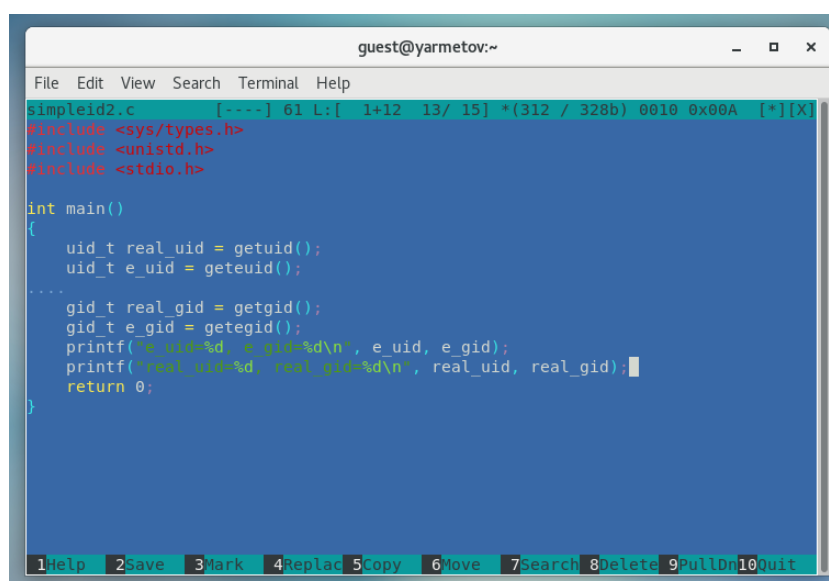
3. Скомпилировали программу и убедились, что файл программы создан: `gcc simpleid.c -o simpleid`
4. Выполнили программу simpleid командой `./simpleid`
5. Выполнили системную программу id с помощью команды `id`. uid и gid совпадает в обеих программах



```
guest@yarmetov:~  
File Edit View Search Terminal Help  
Running transaction  
  Installing : 1:mc-4.8.7-11.el7.x86_64 1/1  
  Verifying  : 1:mc-4.8.7-11.el7.x86_64 1/1  
  
Installed:  
mc.x86_64 1:4.8.7-11.el7  
  
Complete!  
[root@yarmetov ~]# exit  
logout  
[kyarmetov@yarmetov ~]$  
[kyarmetov@yarmetov ~]$ su guest  
Password:  
[guest@yarmetov kyarmetov]$  
[guest@yarmetov kyarmetov]$ cd  
[guest@yarmetov ~]$ mcedit simpleid.c  
  
[guest@yarmetov ~]$ gcc simpleid.c -o simpleid  
[guest@yarmetov ~]$ ./simpleid  
uid=1001, gid=1001  
[guest@yarmetov ~]$ id  
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023  
[guest@yarmetov ~]$
```

Figure 2.3: результат программы simpleid

6. Усложнили программу, добавив вывод действительных идентификаторов.



```
guest@yarmetov:~  
File Edit View Search Terminal Help  
simpleid2.c [1:12 13/15] *(312 / 328b) 0010 0x00A [*][X]  
#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 2.4: программа simpleid2

7. Скомпилировали и запустили simpleid2.c:

```
gcc simpleid2.c -o simpleid2
```

```
./simpleid2
```

8. От имени суперпользователя выполнили команды:

```
chown root:guest /home/guest/simpleid2
```

```
chmod u+s /home/guest/simpleid2
```

9. Использовали su для повышения прав до суперпользователя

10. Выполнили проверку правильности установки новых атрибутов и смены владельца файла simpleid2:

```
ls -l simpleid2
```

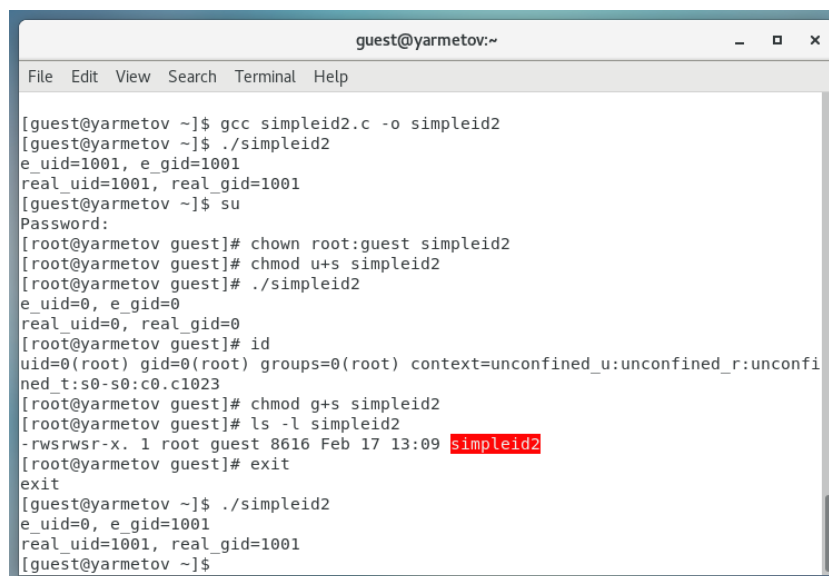
11. Запустили simpleid2 и id:

```
./simpleid2
```

```
id
```

Результат выполнения программ теперь немного отличается

12. Прodelали тоже самое относительно SetGID-бита.

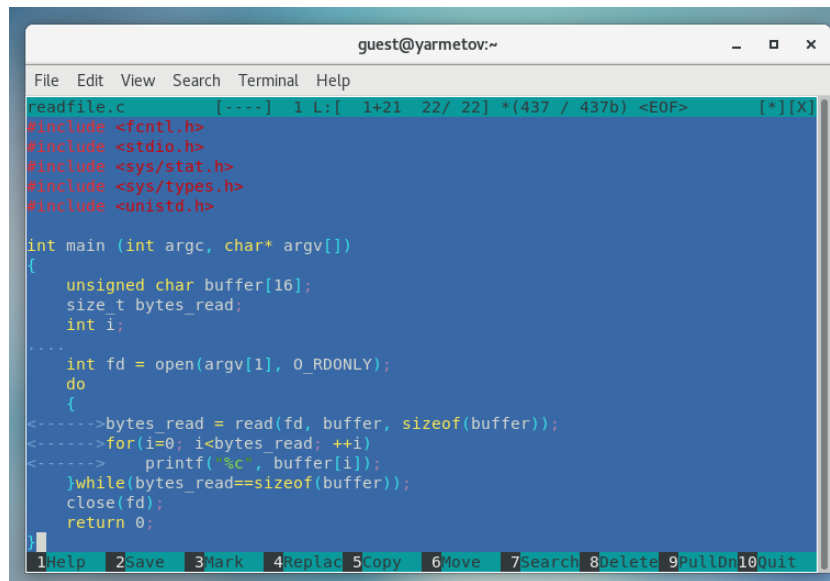


```
guest@yarmetov:~  
File Edit View Search Terminal Help  
[guest@yarmetov ~]$ gcc simpleid2.c -o simpleid2  
[guest@yarmetov ~]$ ./simpleid2  
e_uid=1001, e_gid=1001  
real_uid=1001, real_gid=1001  
[guest@yarmetov ~]$ su  
Password:  
[root@yarmetov guest]# chown root:guest simpleid2  
[root@yarmetov guest]# chmod u+s simpleid2  
[root@yarmetov guest]# ./simpleid2  
e_uid=0, e_gid=0  
real_uid=0, real_gid=0  
[root@yarmetov guest]# id  
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfi  
ned t:s0-s0:c0.c1023  
[root@yarmetov guest]# chmod g+s simpleid2  
[root@yarmetov guest]# ls -l simpleid2  
-rwsrwsr-x. 1 root guest 8616 Feb 17 13:09 simpleid2  
[root@yarmetov guest]# exit  
exit  
[guest@yarmetov ~]$ ./simpleid2  
e_uid=0, e_gid=1001  
real_uid=1001, real_gid=1001  
[guest@yarmetov ~]$
```

Figure 2.5: результат программы simpleid2



### 13. Написали программу readfile.c



```
guest@yarmetov:~  
File Edit View Search Terminal Help  
readfile.c [----] 1 L: [ 1+21 22/ 22] *(437 / 437b) <EOF> [X]  
#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 2.6: программа readfile

### 14. Откомпилировали её.

```
gcc readfile.c -o readfile
```

### 15. Сменили владельца у файла readfile.c и изменили права так, чтобы только суперпользователь (root) мог прочитать его, а guest не мог.

```
chown root:guest /home/guest/readfile.c
```

```
chmod 700 /home/guest/readfile.c
```

### 16. Проверили, что пользователь guest не может прочитать файл readfile.c.

### 17. Сменили у программы readfile владельца и установили SetU'D-бит.

### 18. Проверили, может ли программа readfile прочитать файл readfile.c

### 19. Проверили, может ли программа readfile прочитать файл /etc/shadow

```
guest@yarmetov:~  
File Edit View Search Terminal Help  
[guest@yarmetov ~]$ gcc readfile.c -o readfile  
[guest@yarmetov ~]$ su  
Password:  
[root@yarmetov guest]# chown root:root readfile  
[root@yarmetov guest]# chown root:root readfile.c  
[root@yarmetov guest]# chmod 700 readfile.c  
[root@yarmetov guest]# chmod u+s readfile  
[root@yarmetov guest]# exit  
exit  
[guest@yarmetov ~]$ cat readfile.c  
cat: readfile.c: Permission denied  
[guest@yarmetov ~]$ rea  
read      readcd      readlink      readom      readprofile      realpath  
readarray  readelf      readmult      readonly      realm  
[guest@yarmetov ~]$ ./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  
    {
```

Figure 2.7: результат программы readfile

```
guest@yarmetov:~  
File Edit View Search Terminal Help  
rtkit:!!:19038:.....  
pulse:!!:19038:.....  
radvd:!!:19038:.....  
chrony:!!:19038:.....  
unbound:!!:19038:.....  
qemu:!!:19038:.....  
tss:!!:19038:.....  
usbmuxd:!!:19038:.....  
geoclue:!!:19038:.....  
gluster:!!:19038:.....  
gdm:!!:19038:.....  
rpcuser:!!:19038:.....  
nfsnobody:!!:19038:.....  
gnome-initial-setup:!!:19038:.....  
sshd:!!:19038:.....  
avahi:!!:19038:.....  
postfix:!!:19038:.....  
ntp:!!:19038:.....  
tcpdump:!!:19038:.....  
kyarmetov:$6$n0G1lN76TDeMKIxh$sUTH3iWYCzppnEafkDJWJBipWdANJOMKIE.1yDLNjb3tdgvZc3  
SWI4vimnDs6Z9vaZQpu/P027k1EjKgs/BF11::0:99999:7:::  
vboxadd:!!:19038:.....  
guest:$6$4A4VMui$JeiQarygdKIsbz0pKQubuz2dxMm23BBoA1QAN.eqkijPTdAnj5VPPKpfjnCCn  
80wEgSt54/.JbN4VkjIG2q7/:19038:0:99999:7:::  
guest2:$6$3KLkwjuM$FZmo5XaB12Koz9wQWFvYIMy7CkCbo.1hI3vjJwR09A85EzC2anSzBhE3hu5P9  
Pbi1T5gm1GNLC34lQ60Ae9jJ1:19039:0:99999:7:::  
[guest@yarmetov ~]$
```

Figure 2.8: результат программы readfile

## 2.3 Исследование Sticky-бита

1. Выяснили, установлен ли атрибут Sticky на директории /tmp:

```
ls -l / | grep tmp
```

2. От имени пользователя guest создали файл file01.txt в директории /tmp со словом test:

```
echo "test" > /tmp/file01.txt
```

3. Просмотрели атрибуты у только что созданного файла и разрешили чтение и запись для категории пользователей «все остальные»:

```
ls -l /tmp/file01.txt  
chmod o+rw /tmp/file01.txt  
ls -l /tmp/file01.txt
```

Первоначально все группы имели право на чтение, а запись могли осуществлять все, кроме «остальных пользователей».

4. От пользователя (не являющегося владельцем) попробовали прочитать файл /file01.txt:

```
cat /file01.txt
```

5. От пользователя попробовали дозаписать в файл /file01.txt слово test3 командой:

```
echo "test2" >> /file01.txt
```

6. Проверили содержимое файла командой:

```
cat /file01.txt
```

В файле теперь записано:

Test

Test2

7. От пользователя попробовали записать в файл /tmp/file01.txt слово test4, стерев при этом всю имеющуюся в файле информацию командой. Для этого воспользовалась командой `echo "test3" > /tmp/file01.txt`

8. Проверили содержимое файла командой

```
cat /tmp/file01.txt
```

9. От пользователя попробовали удалить файл /tmp/file01.txt командой `rm /tmp/file01.txt`, однако получила отказ.
10. От суперпользователя командой выполнили команду, снимающую атрибут `t` (Sticky-бит) с директории /tmp:

```
chmod -t /tmp
```

Покинули режим суперпользователя командой `exit`.

11. От пользователя проверили, что атрибута `t` у директории /tmp нет:

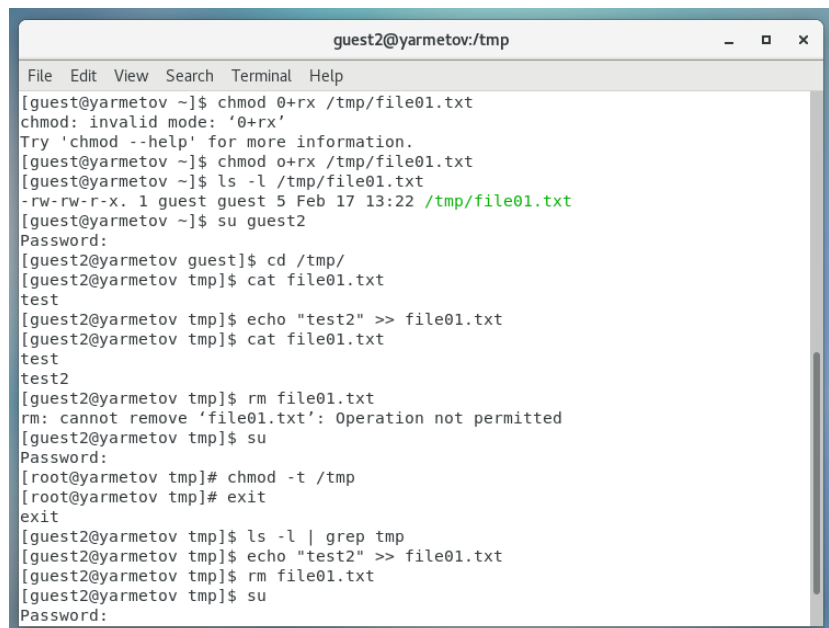
```
ls -l / | grep tmp
```

12. Повторили предыдущие шаги. Получилось удалить файл
13. Удалось удалить файл от имени пользователя, не являющегося его владельцем.
14. Повысили свои права до суперпользователя и вернули атрибут `t` на директорию /tmp :

```
su
```

```
chmod +t /tmp
```

```
exit
```



```
guest2@yarmetov:~/tmp
File Edit View Search Terminal Help
[guest@yarmetov ~]$ chmod 0+rx /tmp/file01.txt
chmod: invalid mode: '0+rx'
Try 'chmod --help' for more information.
[guest@yarmetov ~]$ chmod o+rx /tmp/file01.txt
[guest@yarmetov ~]$ ls -l /tmp/file01.txt
-rw-rw-r-x. 1 guest guest 5 Feb 17 13:22 /tmp/file01.txt
[guest@yarmetov ~]$ su guest2
Password:
[guest2@yarmetov guest]$ cd /tmp/
[guest2@yarmetov tmp]$ cat file01.txt
test
[guest2@yarmetov tmp]$ echo "test2" >> file01.txt
[guest2@yarmetov tmp]$ cat file01.txt
test
test2
[guest2@yarmetov tmp]$ rm file01.txt
rm: cannot remove 'file01.txt': Operation not permitted
[guest2@yarmetov tmp]$ su
Password:
[root@yarmetov tmp]# chmod -t /tmp
[root@yarmetov tmp]# exit
exit
[guest2@yarmetov tmp]$ ls -l | grep tmp
[guest2@yarmetov tmp]$ echo "test2" >> file01.txt
[guest2@yarmetov tmp]$ rm file01.txt
[guest2@yarmetov tmp]$ su
Password:
```

Figure 2.9: исследование Sticky-бита

## 3 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.

# Список литературы

1. КОМАНДА CHATTR В LINUX
2. chattr