

Отчет по лабораторной работе №5

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

Каримов Зуфар НПИ-01-18

Содержание

1	Цель работы	3
2	Последовательность выполнения работы	4
2.1	Создание программы	4
2.2	Исследование Sticky-бита	12
3	Выводы	16

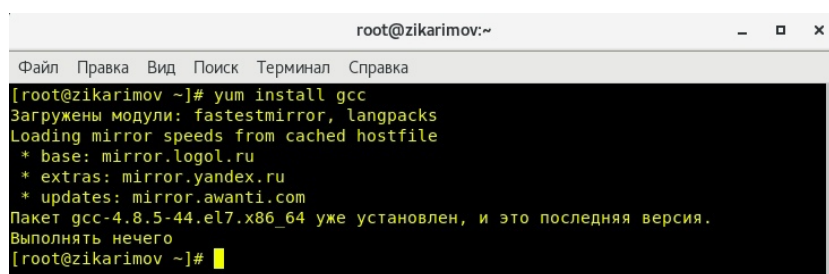
1 Цель работы

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 Последовательность выполнения работы

2.1 Создание программы

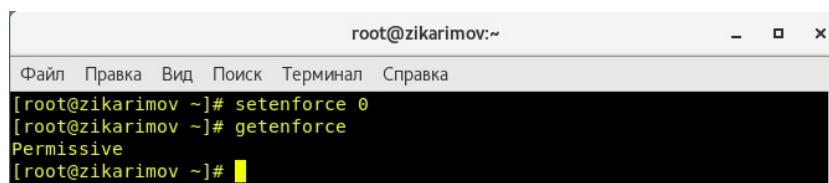
Для начала нам следовало установить компилятор gcc. (рис. 2.1)



```
root@zikarimov:~  
Файл Правка Вид Поиск Терминал Справка  
[root@zikarimov ~]# yum install gcc  
Загружены модули: fastestmirror, langpacks  
Loading mirror speeds from cached hostfile  
* base: mirror.logol.ru  
* extras: mirror.yandex.ru  
* updates: mirror.awanti.com  
Пакет gcc-4.8.5-44.el7.x86_64 уже установлен, и это последняя версия.  
Выполнять нечего  
[root@zikarimov ~]#
```

Figure 2.1: Компилятор gcc

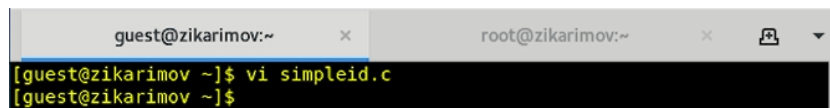
Чтобы защита SELinux не мешала выполнению заданий работы, мы отключили ее. (рис. 2.2)



```
root@zikarimov:~  
Файл Правка Вид Поиск Терминал Справка  
[root@zikarimov ~]# setenforce 0  
[root@zikarimov ~]# getenforce  
Permissive  
[root@zikarimov ~]#
```

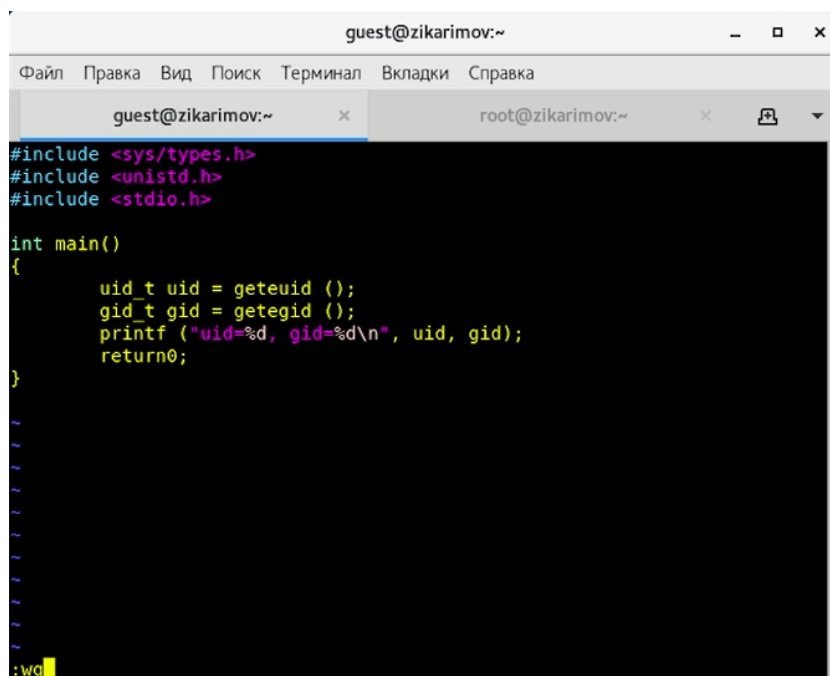
Figure 2.2: Отключение защиты

1. Войдите в систему от имени пользователя guest.
2. Создайте программу simpleid.c: (рис. 2.3) (рис. 2.4)



```
guest@zikarimov:~ x root@zikarimov:~ x
[guest@zikarimov ~]$ vi simpleid.c
[guest@zikarimov ~]$
```

Figure 2.3: Программа simpleid.c



```
guest@zikarimov:~
Файл Правка Вид Поиск Терминал Вкладки Справка
guest@zikarimov:~ x root@zikarimov:~ 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;
}

:wq
```

Figure 2.4: Программа simpleid.c

3. Скомпилируйте программу и убедитесь, что файл программы создан:

```
gcc simpleid.c -o simpleid
```

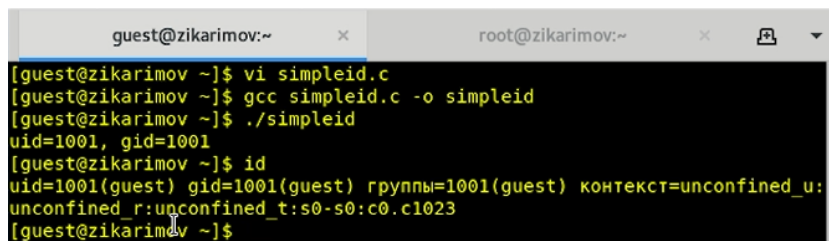
4. Выполните программу simpleid:

```
./simpleid
```

5. Выполните системную программу id:

```
id
```

и сравните полученный вами результат с данными предыдущего пункта задания. (рис. 2.5)



```
guest@zikarimov:~  
[guest@zikarimov ~]$ vi simpleid.c  
[guest@zikarimov ~]$ gcc simpleid.c -o simpleid  
[guest@zikarimov ~]$ ./simpleid  
uid=1001, gid=1001  
[guest@zikarimov ~]$ id  
uid=1001(guest) gid=1001(guest) rгруппы=1001(guest) контекст=unconfined_u:  
unconfined_r:unconfined_t:s0-s0:c0.c1023  
[guest@zikarimov ~]$
```

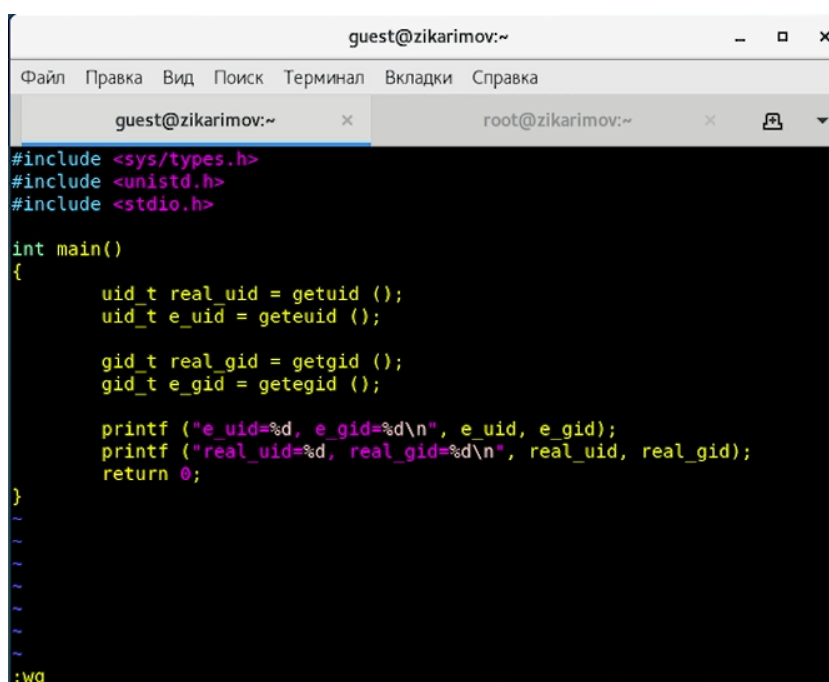
Figure 2.5: Компиляция и выполнения программы

6. Усложните программу, добавив вывод действительных идентификаторов:
(рис. 2.6) (рис. 2.7)



```
[guest@zikarimov ~]$ vi simpleid2.c  
[guest@zikarimov ~]$
```

Figure 2.6: Программа simpleid2.c



```
guest@zikarimov:~  
Файл Правка Вид Поиск Терминал Вкладки Справка  
guest@zikarimov:~ root@zikarimov:~  
#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;  
}  
~  
~  
~  
~  
~  
:wq
```

Figure 2.7: Программа simpleid2.c

7. Скомпилируйте и запустите simpleid2.c:

```
gcc simpleid2.c -o simpleid2  
./simpleid2 (рис. 2.8)
```

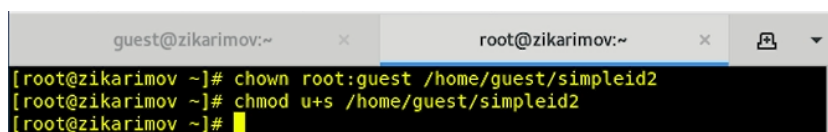
```
[guest@zikarimov ~]$ gcc simpleid2.c -o simpleid2
[guest@zikarimov ~]$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
[guest@zikarimov ~]$
```

Figure 2.8: Компиляция и выполнения программы

8. От имени суперпользователя выполните команды:

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

`chmod u+s /home/guest/simpleid2` (рис. 2.9)



```
guest@zikarimov:~ x root@zikarimov:~ x
[root@zikarimov ~]# chown root:guest /home/guest/simpleid2
[root@zikarimov ~]# chmod u+s /home/guest/simpleid2
[root@zikarimov ~]#
```

Figure 2.9: Смена пользователя и установка SetU'D-бита

9. Используйте `sudo` или повысьте временно свои права с помощью `su`. Поясните, что делают эти команды.

Команда `sudo` позволяет пользователям выполнять указанные программы с административными привилегиями без ввода пароля суперпользователя `root`.

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

`ls -l simpleid2`

11. Запустите `simpleid2` и `id`:

`./simpleid2`

`id`

Сравните результаты. (рис. 2.10)

```
[guest@zikarimov ~]$ ls -l simpleid2
-rwsrwxr-x. 1 root guest 8576 ноя 13 12:59 simpleid2
[guest@zikarimov ~]$ ./simpleid2
e_uid=0, e_gid=1001
real_uid=1001, real_gid=1001
[guest@zikarimov ~]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:
unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@zikarimov ~]$
```

Figure 2.10: Проверка правильности установки новых атрибутов

12. Прodelайте тоже самое относительно SetGID-бита. (рис. 2.11) (рис. 2.12)

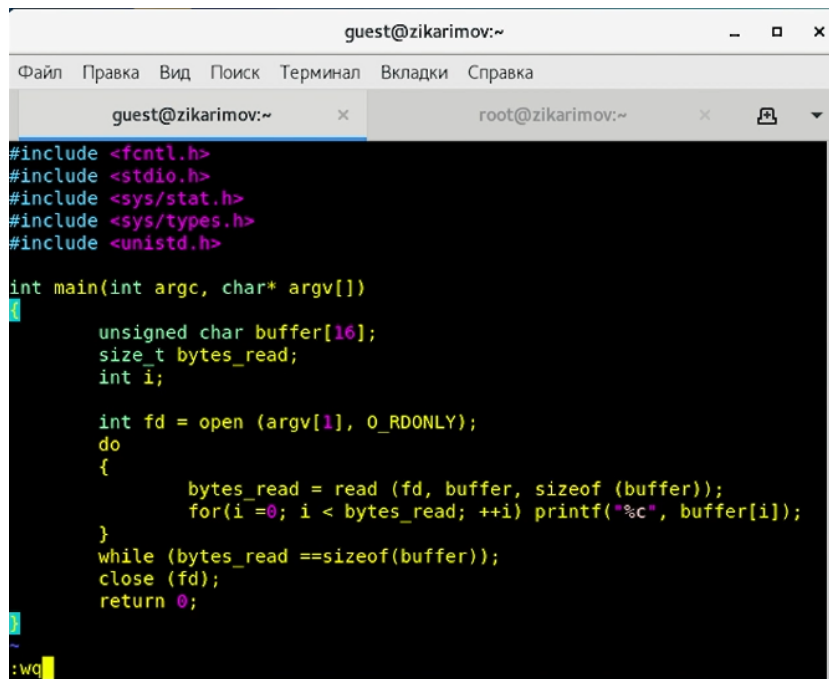
```
guest@zikarimov:~ root@zikarimov:~
[root@zikarimov ~]# chown root:guest /home/guest/simpleid2
[root@zikarimov ~]# chmod u+s /home/guest/simpleid2
[root@zikarimov ~]# chmod g+s /home/guest/simpleid2
[root@zikarimov ~]#
```

Figure 2.11: Проверка правильности установки новых атрибутов

```
guest@zikarimov:~ root@zikarimov:~
[guest@zikarimov ~]$ ls -l simpleid2
-rwsrwsr-x. 1 root guest 8576 ноя 13 12:59 simpleid2
[guest@zikarimov ~]$ ./simpleid2
e_uid=0, e_gid=1001
real_uid=1001, real_gid=1001
[guest@zikarimov ~]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:
unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@zikarimov ~]$
```

Figure 2.12: Проверка правильности установки новых атрибутов

13. . Создайте программу readfile.c: (рис. 2.13)

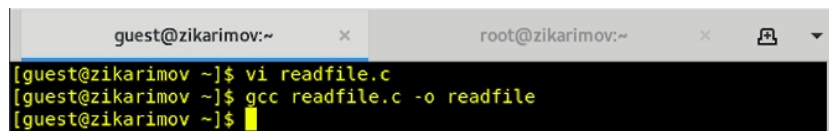


```
guest@zikarimov:~  
Файл Правка Вид Поиск Терминал Вкладки Справка  
guest@zikarimov:~ x root@zikarimov:~ x  
#include <fcntl.h>  
#include <stdio.h>  
#include <sys/stat.h>  
#include <sys/types.h>  
#include <unistd.h>  
  
int main(int argc, char* argv[])  
{  
    unsigned char buffer[16];  
    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 2.13: Программа readfile.c

14. Откомпилируйте её.

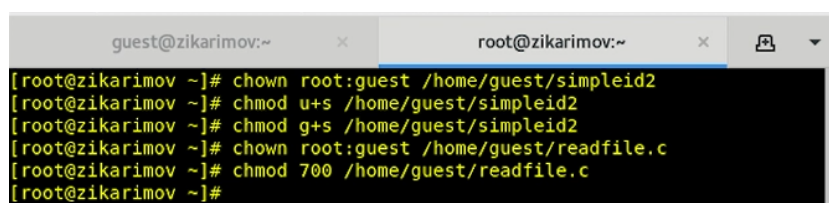
gcc readfile.c -o readfile (рис. 2.14)



```
guest@zikarimov:~ x root@zikarimov:~ x  
[guest@zikarimov ~]$ vi readfile.c  
[guest@zikarimov ~]$ gcc readfile.c -o readfile  
[guest@zikarimov ~]$
```

Figure 2.14: Программа readfile

15. Смените владельца у файла readfile.c (или любого другого текстового файла в системе) и измените права так, чтобы только суперпользователь (root) мог прочитать его, а guest не мог (рис. 2.15)



```
guest@zikarimov:~ x root@zikarimov:~ x  
[root@zikarimov ~]# chown root:guest /home/guest/simpleid2  
[root@zikarimov ~]# chmod u+s /home/guest/simpleid2  
[root@zikarimov ~]# chmod g+s /home/guest/simpleid2  
[root@zikarimov ~]# chown root:guest /home/guest/readfile.c  
[root@zikarimov ~]# chmod 700 /home/guest/readfile.c  
[root@zikarimov ~]#
```

Figure 2.15: Смена владельца и изменения прав

16. Проверьте, что пользователь guest не может прочитать файл readfile.c. (рис. 2.16)

```
[guest@zikarimov ~]$ ls -l readfile.c
-rwx-----. 1 root guest 415 ноя 13 13:05 readfile.c
[guest@zikarimov ~]$ cat readfile.c
cat: readfile.c: Отказано в доступе
[guest@zikarimov ~]$
```

Figure 2.16: Проверка на правильность

17. Смените у программы readfile владельца и установите SetU'D-бит. (рис. 2.17)

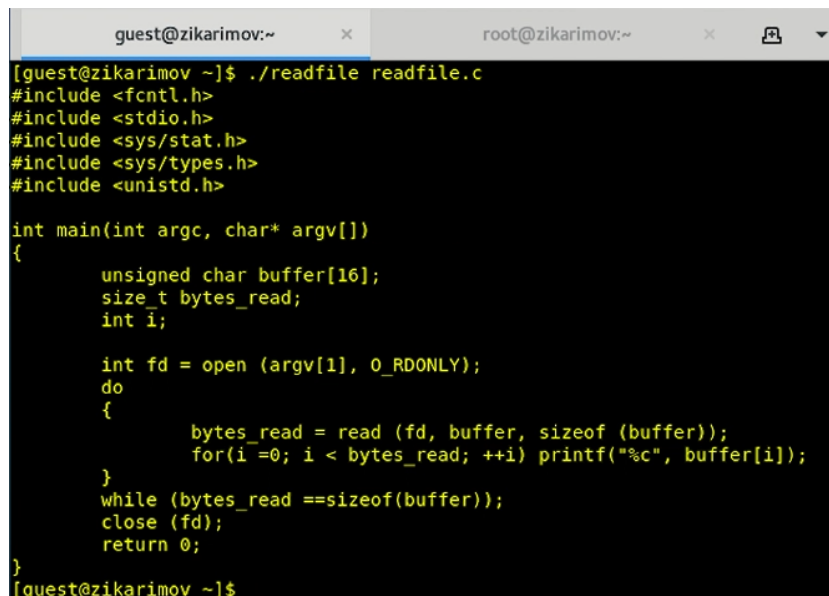
```
guest@zikarimov:~ x root@zikarimov:~ x
[root@zikarimov ~]# chown root:guest /home/guest/simpleid2
[root@zikarimov ~]# chmod u+s /home/guest/simpleid2
[root@zikarimov ~]# chmod g+s /home/guest/simpleid2
[root@zikarimov ~]# chown root:guest /home/guest/readfile.c
[root@zikarimov ~]# chmod 700 /home/guest/readfile.c
[root@zikarimov ~]# chown root:guest /home/guest/readfile
[root@zikarimov ~]# chmod u+s /home/guest/readfile
[root@zikarimov ~]#
```

Figure 2.17: Смена пользователя и установка SetU'D-бита

18. Проверьте, может ли программа readfile прочитать файл readfile.c? (рис. 2.18) (рис. 2.19)

```
[guest@zikarimov ~]$ ls -l readfile
-rwsrwxr-x. 1 root guest 8512 ноя 13 13:05 readfile
```

Figure 2.18: Проверка на правильность



```
guest@zikarimov:~ x root@zikarimov:~ x
[guest@zikarimov ~]$ ./readfile readfile.c
#include <fcntl.h>
#include <stdio.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>

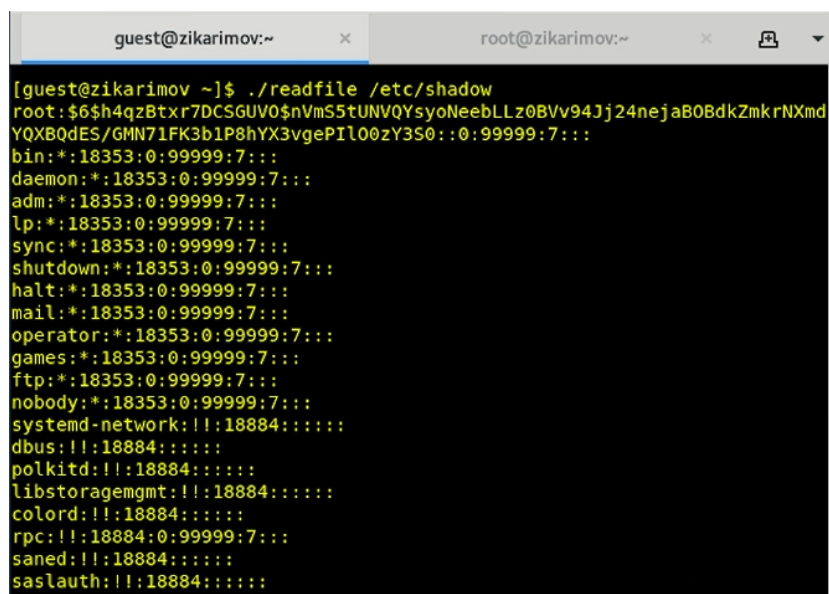
int main(int argc, char* argv[])
{
    unsigned char buffer[16];
    size_t bytes_read;
    int i;

    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@zikarimov ~]$
```

Figure 2.19: Чтения файла

19. Проверьте, может ли программа readfile прочитать файл /etc/shadow?

Отразите полученный результат и ваши объяснения в отчёте. (рис. 2.20)



```
guest@zikarimov:~ x root@zikarimov:~ x
[guest@zikarimov ~]$ ./readfile /etc/shadow
root:$6$h4qzBtxr7DCSGUV0$nmVmS5tUNVQYsy0NeebLLz0BVv94Jj24nejaB0BdkZmkrNXmd
YQXBqdES/GMN71FK3b1P8hYX3vgePIl00zy3S0::0:99999:7:::
bin:!:18353:0:99999:7:::
daemon:!:18353:0:99999:7:::
adm:!:18353:0:99999:7:::
lp:!:18353:0:99999:7:::
sync:!:18353:0:99999:7:::
shutdown:!:18353:0:99999:7:::
halt:!:18353:0:99999:7:::
mail:!:18353:0:99999:7:::
operator:!:18353:0:99999:7:::
games:!:18353:0:99999:7:::
ftp:!:18353:0:99999:7:::
nobody:!:18353:0:99999:7:::
systemd-network:!!:18884:::::
dbus:!!:18884:::::
polkitd:!!:18884:::::
libstoragemgmt:!!:18884:::::
colord:!!:18884:::::
rpc:!!:18884:0:99999:7:::
saned:!!:18884:::::
ssslauth:!!:18884:::::

```

Figure 2.20: Чтения файла

2.2 Исследование Sticky-бита

1. Выясните, установлен ли атрибут Sticky на директории /tmp, для чего выполните команду

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

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

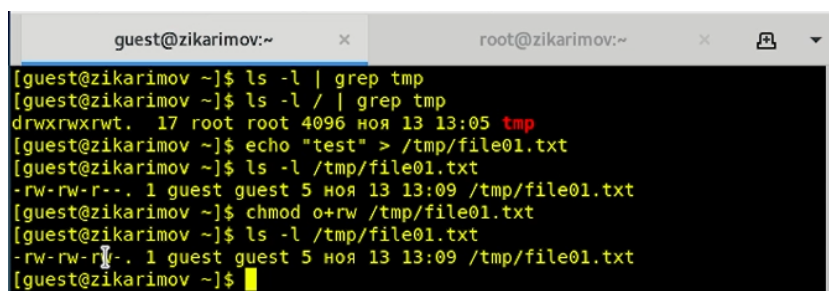
```
echo "test" > /tmp/file01.txt
```

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

```
ls -l /tmp/file01.txt
```

```
chmod o+rw /tmp/file01.txt
```

```
ls -l /tmp/file01.txt (рис. 2.21)
```



```
guest@zikarimov:~ root@zikarimov:~
[guest@zikarimov ~]$ ls -l | grep tmp
[guest@zikarimov ~]$ ls -l / | grep tmp
drwxrwxrwt. 17 root root 4096 ноя 13 13:05 tmp
[guest@zikarimov ~]$ echo "test" > /tmp/file01.txt
[guest@zikarimov ~]$ ls -l /tmp/file01.txt
-rw-rw-r--. 1 guest guest 5 ноя 13 13:09 /tmp/file01.txt
[guest@zikarimov ~]$ chmod o+rw /tmp/file01.txt
[guest@zikarimov ~]$ ls -l /tmp/file01.txt
-rw-rw-rw-. 1 guest guest 5 ноя 13 13:09 /tmp/file01.txt
[guest@zikarimov ~]$
```

Figure 2.21: Просмотр атрибутов и разрешения прав

4. От пользователя guest2 (не являющегося владельцем) попробуйте прочитать файл /tmp/file01.txt:

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

5. От пользователя guest2 попробуйте дозаписать в файл /tmp/file01.txt слово test2 командой

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

Удалось ли вам выполнить операцию? Да, удалось.

6. Проверьте содержимое файла командой

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

7. От пользователя guest2 попробуйте записать в файл /tmp/file01.txt слово test3, стерев при этом всю имеющуюся в файле информацию командой

```
echo "test3" > /tmp/file01.txt
```

Удалось ли вам выполнить операцию? Да, удалось выполнить операцию.

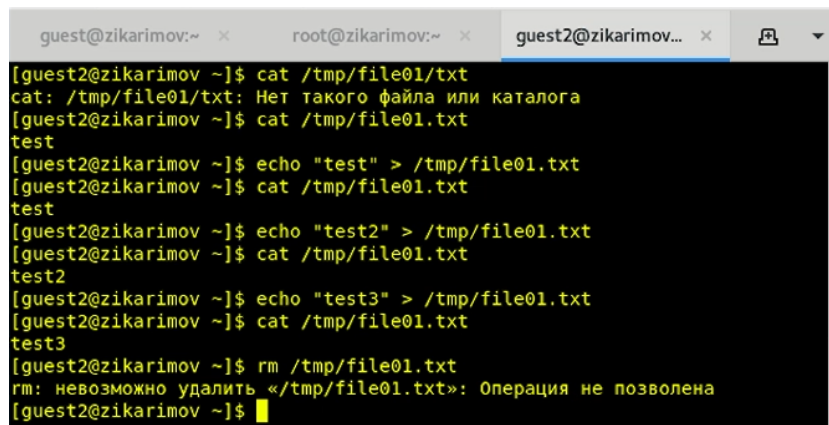
8. Проверьте содержимое файла командой

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

9. От пользователя guest2 попробуйте удалить файл /tmp/file01.txt командой

```
rm /tmp/file01.txt
```

Удалось ли вам удалить файл? Нет, не удалось удалить файл. (рис. 2.22)



```
guest@zikarimov:~ x root@zikarimov:~ x guest2@zikarimov... x
[guest2@zikarimov ~]$ cat /tmp/file01/txt
cat: /tmp/file01/txt: Нет такого файла или каталога
[guest2@zikarimov ~]$ cat /tmp/file01.txt
test
[guest2@zikarimov ~]$ echo "test" > /tmp/file01.txt
[guest2@zikarimov ~]$ cat /tmp/file01.txt
test
[guest2@zikarimov ~]$ echo "test2" > /tmp/file01.txt
[guest2@zikarimov ~]$ cat /tmp/file01.txt
test2
[guest2@zikarimov ~]$ echo "test3" > /tmp/file01.txt
[guest2@zikarimov ~]$ cat /tmp/file01.txt
test3
[guest2@zikarimov ~]$ rm /tmp/file01.txt
rm: невозможно удалить «/tmp/file01.txt»: Операция не позволена
[guest2@zikarimov ~]$
```

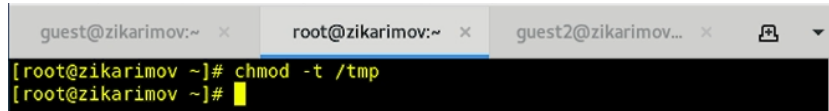
Figure 2.22: Просмотр атрибутов и разрешения прав

10. Повысьте свои права до суперпользователя следующей командой

su -

и выполните после этого команду, снимающую атрибут t (Sticky-бит) с директории /tmp:

chmod -t /tmp (рис. 2.23)



```
guest@zikarimov:~ x root@zikarimov:~ x guest2@zikarimov... x
[root@zikarimov ~]# chmod -t /tmp
[root@zikarimov ~]#
```

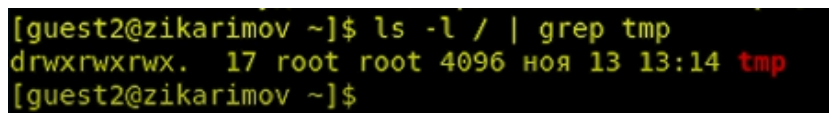
Figure 2.23: Просмотр атрибутов и разрешения прав

11. Покиньте режим суперпользователя командой

exit

12. От пользователя guest2 проверьте, что атрибута t у директории /tmp нет:

ls -l / | grep tmp (рис. 2.24)

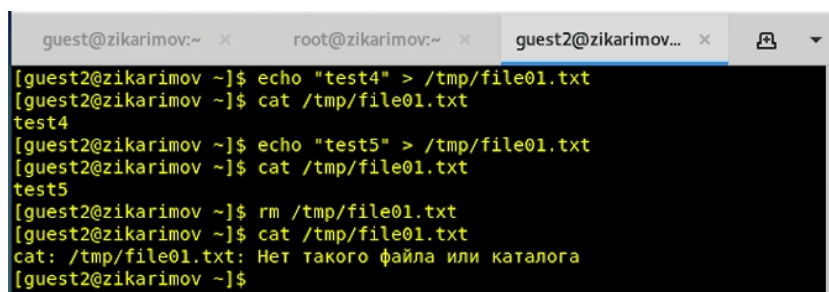


```
[guest2@zikarimov ~]$ ls -l / | grep tmp
drwxrwxrwx. 17 root root 4096 ноя 13 13:14 tmp
[guest2@zikarimov ~]$
```

Figure 2.24: Просмотр атрибутов и разрешения прав

13. Повторите предыдущие шаги. Какие наблюдаются изменения?

14. Удалось ли вам удалить файл от имени пользователя, не являющегося его владельцем? Ваши наблюдения занесите в отчёт. (рис. 2.25)



```
guest@zikarimov:~ x root@zikarimov:~ x guest2@zikarimov... x
[guest2@zikarimov ~]$ echo "test4" > /tmp/file01.txt
[guest2@zikarimov ~]$ cat /tmp/file01.txt
test4
[guest2@zikarimov ~]$ echo "test5" > /tmp/file01.txt
[guest2@zikarimov ~]$ cat /tmp/file01.txt
test5
[guest2@zikarimov ~]$ rm /tmp/file01.txt
[guest2@zikarimov ~]$ cat /tmp/file01.txt
cat: /tmp/file01.txt: Нет такого файла или каталога
[guest2@zikarimov ~]$
```

Figure 2.25: Просмотр атрибутов и разрешения прав

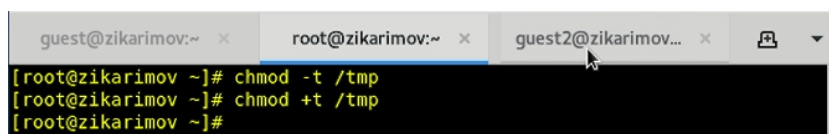
Да, удалось удалить файл от имени пользователя, не являющегося его владельцем.

15. Повысьте свои права до суперпользователя и верните атрибут `t` на директорию `/tmp`:

`su -`

`chmod +t /tmp`

`exit` (рис. 2.26)



```
guest@zikarimov:~ x root@zikarimov:~ x guest2@zikarimov... x
[root@zikarimov ~]# chmod -t /tmp
[root@zikarimov ~]# chmod +t /tmp
[root@zikarimov ~]#
```

Figure 2.26: Просмотр атрибутов и разрешения прав

3 Выводы

Изучил механизмы изменения идентификаторов, применив SetUID- и Sticky-биты. Получил практические навыки работы в консоли с дополнительными атрибутами. Рассмотрел работы механизмов смены идентификаторов процесса пользователей, а также влияние бита Sticky на запись и удаление файлов.