

Отчёт по лабораторной работе №5

Основы информационной безопасности

Ищенко Ирина НПИбд-02-22

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	15
	Список литературы	16

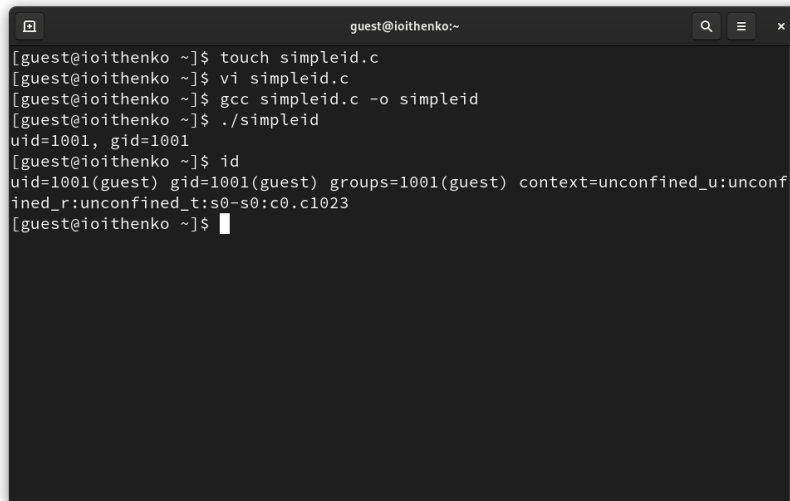
Список иллюстраций

2.1	simpleid.c	6
2.2	Выполнение simpleid	7
2.3	simpleid02	7
2.4	Выполнение simpleid02	8
2.5	Смена владельца и атрибут	8
2.6	readfile.c	9
2.7	Изменение владельца и прав	9
2.8	Отказ в чтении	10
2.9	Смена владельца и атрибута	10
2.10	Выполнение проверки	11
2.11	Выполнение проверки	11
2.12	Права на файл	12
2.13	Проверка атрибута	13
2.14	Проверка снятия атрибута	14

Список таблиц

1 Цель работы

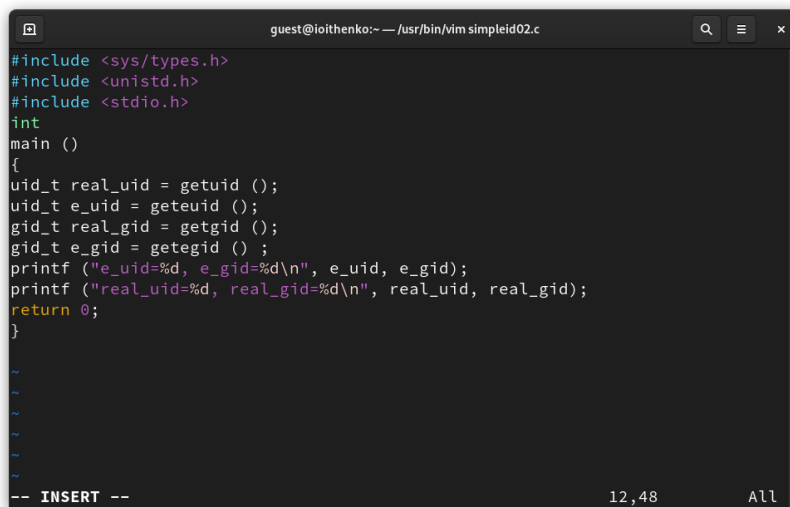
Изучение механизмов изменения идентификаторов, применения SetUID- и Sticky-битов. Получение практических навыков работы в консоли с дополнительными атрибутами. Рассмотрение работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов[1].



```
guest@ioithenko:~$ touch simpleid.c
guest@ioithenko:~$ vi simpleid.c
guest@ioithenko:~$ gcc simpleid.c -o simpleid
guest@ioithenko:~$ ./simpleid
uid=1001, gid=1001
guest@ioithenko:~$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
guest@ioithenko:~$
```

Рис. 2.2: Выполнение simpleid

Усложним программу, добавив вывод действительных идентификаторов (рис. 2.3):

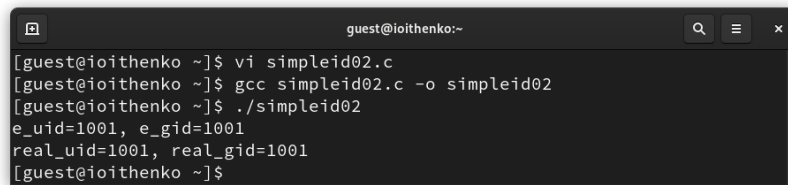


```
guest@ioithenko:~ -- /usr/bin/vim simpleid02.c
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>
int
main ()
{
    uid_t real_uid = getuid ();
    uid_t e_uid = geteuid ();
    gid_t real_gid = getgid ();
    gid_t e_gid = getegid ();
    printf ("e_uid=%d, e_gid=%d\n", e_uid, e_gid);
    printf ("real_uid=%d, real_gid=%d\n", real_uid, real_gid);
    return 0;
}

~
~
~
~
~
-- INSERT -- 12,48 All
```

Рис. 2.3: simpleid02

Скомпилируем и запустим simpleid02.c: gcc simpleid02.c -o simpleid02 ./simpleid2 (рис. 2.4).

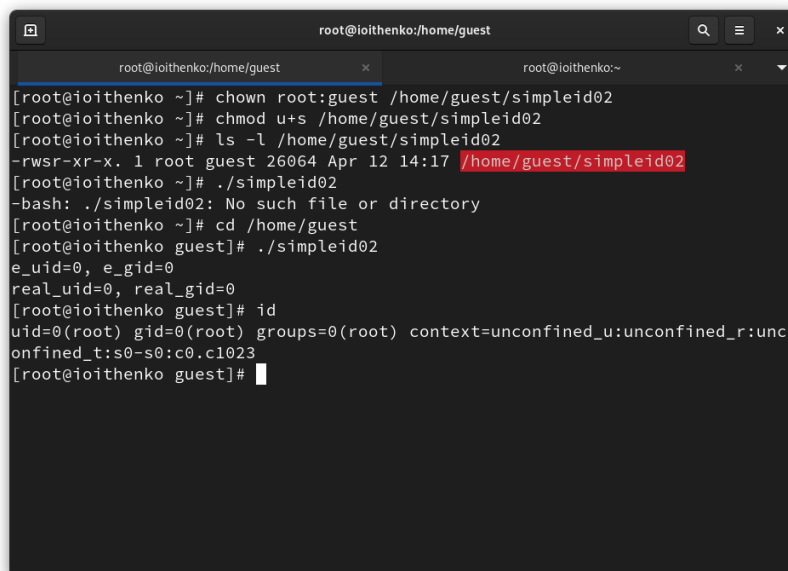


```
guest@ioithenko:~  
[guest@ioithenko ~]$ vi simpleid02.c  
[guest@ioithenko ~]$ gcc simpleid02.c -o simpleid02  
[guest@ioithenko ~]$ ./simpleid02  
e_uid=1001, e_gid=1001  
real_uid=1001, real_gid=1001  
[guest@ioithenko ~]$
```

Рис. 2.4: Выполнение simpleid02

От имени суперпользователя выполнив команды: `chown root:guest /home/guest/simpleid2` `chmod u+s /home/guest/simpleid2` Т.е. изменяем владельца файла и добавляем атрибут.

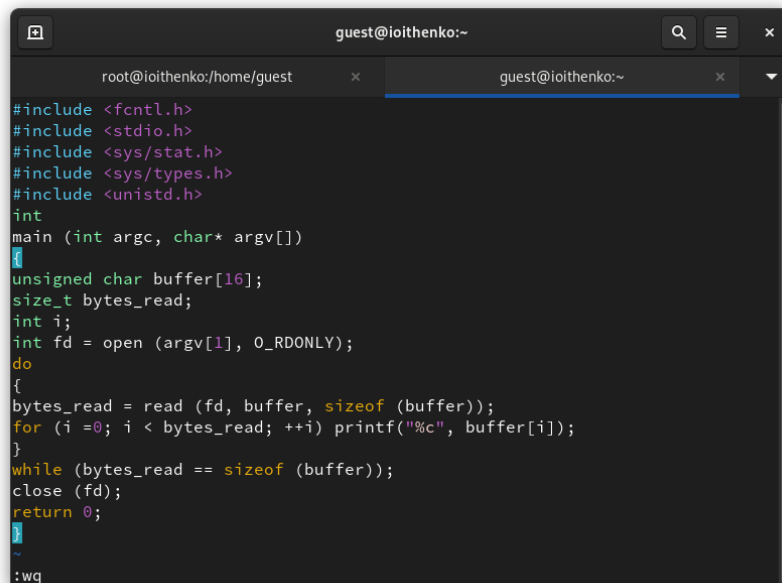
Выполним проверку правильности установки новых атрибутов и смены владельца файла `simpleid02`: `ls -l simpleid02` Запустим `simpleid02` и `id: ./simpleid02` Сравним результаты. Данные совпадают (рис. 2.5).



```
root@ioithenko:/home/guest  
[root@ioithenko ~]# chown root:guest /home/guest/simpleid02  
[root@ioithenko ~]# chmod u+s /home/guest/simpleid02  
[root@ioithenko ~]# ls -l /home/guest/simpleid02  
-rwsr-xr-x. 1 root guest 26064 Apr 12 14:17 /home/guest/simpleid02  
[root@ioithenko ~]# ./simpleid02  
-bash: ./simpleid02: No such file or directory  
[root@ioithenko ~]# cd /home/guest  
[root@ioithenko guest]# ./simpleid02  
e_uid=0, e_gid=0  
real_uid=0, real_gid=0  
[root@ioithenko guest]# id  
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unc  
onfined_t:s0-s0:c0.c1023  
[root@ioithenko guest]#
```

Рис. 2.5: Смена владельца и атрибут

Создадим программу `readfile.c` (рис. 2.6):

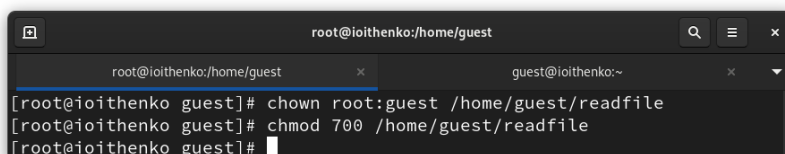


```
#include <fcntl.h>
#include <stdio.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>
int
main (int argc, char* argv[])
{
    unsigned char buffer[16];
    size_t bytes_read;
    int i;
    int fd = open (argv[1], O_RDONLY);
    do
    {
        bytes_read = read (fd, buffer, sizeof (buffer));
        for (i = 0; i < bytes_read; ++i) printf("%c", buffer[i]);
    }
    while (bytes_read == sizeof (buffer));
    close (fd);
    return 0;
}
```

Рис. 2.6: readfile.c

Откомпилируем её. gcc readfile.c -o readfile

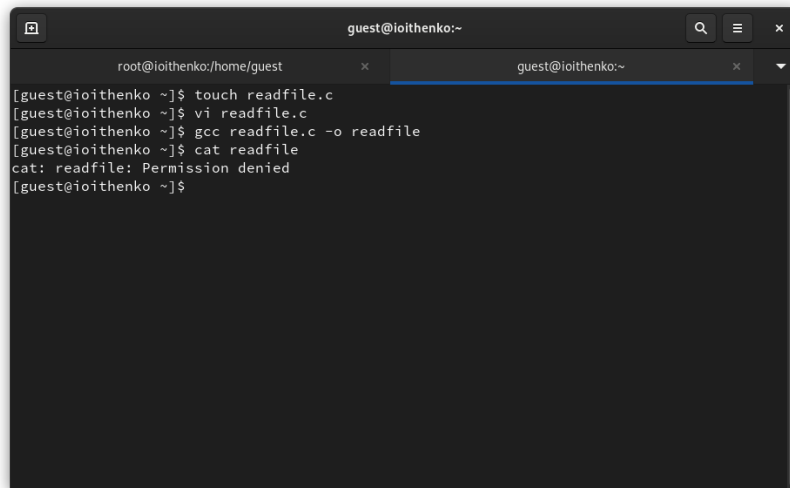
Сменим владельца у файла readfile.c (или любого другого текстового файла в системе) и изменим права так, чтобы только суперпользователь (root) мог прочитать его, а guest не мог (рис. 2.7).



```
[root@ioithenko guest]# chown root:guest /home/guest/readfile
[root@ioithenko guest]# chmod 700 /home/guest/readfile
[root@ioithenko guest]#
```

Рис. 2.7: Изменение владельца и прав

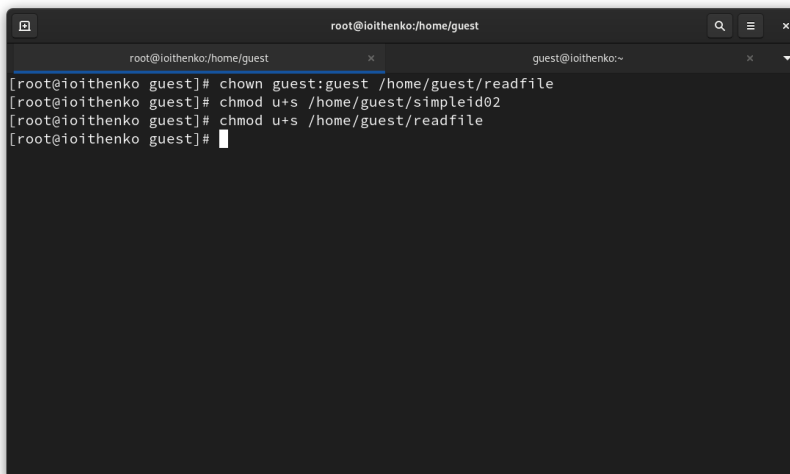
Проверим, что пользователь guest не может прочитать файл readfile.c (рис. 2.8).

A terminal window with two tabs. The first tab is titled 'root@ioithenko:/home/guest' and the second is 'guest@ioithenko:~'. The first tab is active, showing the following commands and output:

```
[guest@ioithenko ~]$ touch readfile.c
[guest@ioithenko ~]$ vi readfile.c
[guest@ioithenko ~]$ gcc readfile.c -o readfile
[guest@ioithenko ~]$ cat readfile
cat: readfile: Permission denied
[guest@ioithenko ~]$
```

Рис. 2.8: Отказ в чтении

Сменим у программы readfile владельца и установим SetU'D-бит (рис. 2.9).

A terminal window with two tabs. The first tab is titled 'root@ioithenko:/home/guest' and the second is 'guest@ioithenko:~'. The first tab is active, showing the following commands and output:

```
[root@ioithenko guest]# chown guest:guest /home/guest/readfile
[root@ioithenko guest]# chmod u+s /home/guest/simpleid02
[root@ioithenko guest]# chmod u+s /home/guest/readfile
[root@ioithenko guest]#
```

Рис. 2.9: Смена владельца и атрибута

Проверим, может ли программа readfile прочитать файл readfile.c? Может (рис. 2.10).

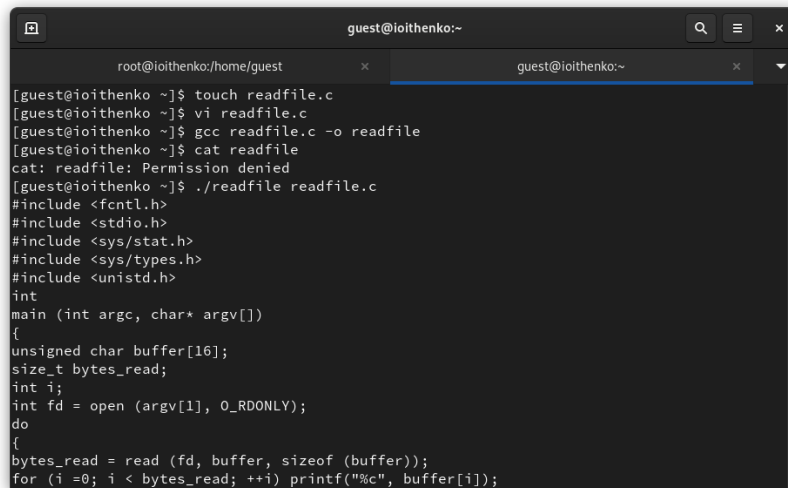


Рис. 2.10: Выполнение проверки

Проверим, может ли программа `readfile` прочитать файл `/etc/shadow`? Может (рис. 2.11).

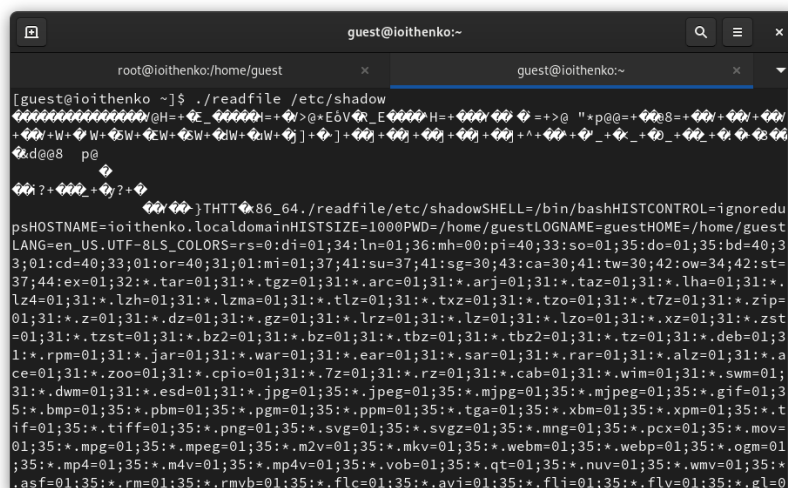
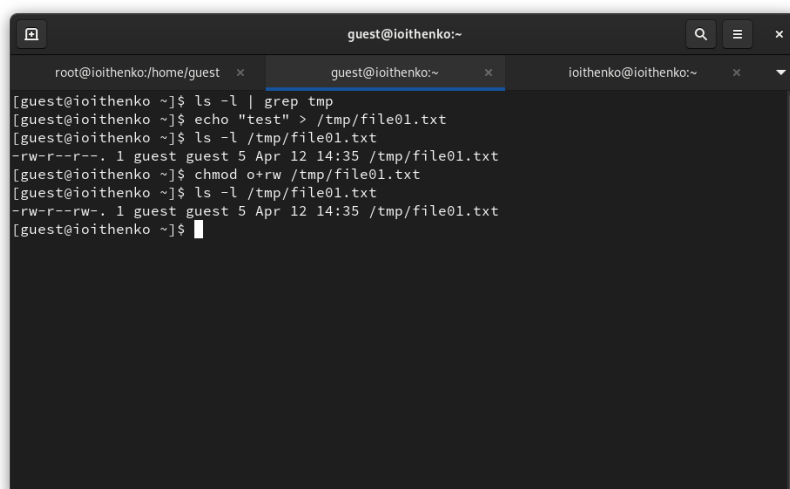


Рис. 2.11: Выполнение проверки

Выясним, установлен ли атрибут Sticky на директории /tmp, для чего выполните команду `ls -l | grep tmp` От имени пользователя guest создадим файл file01.txt в директории /tmp со словом test: `echo "test" > /tmp/file01.txt` Просмотрим атрибуты у только что созданного файла и разрешим чтение и запись для категории

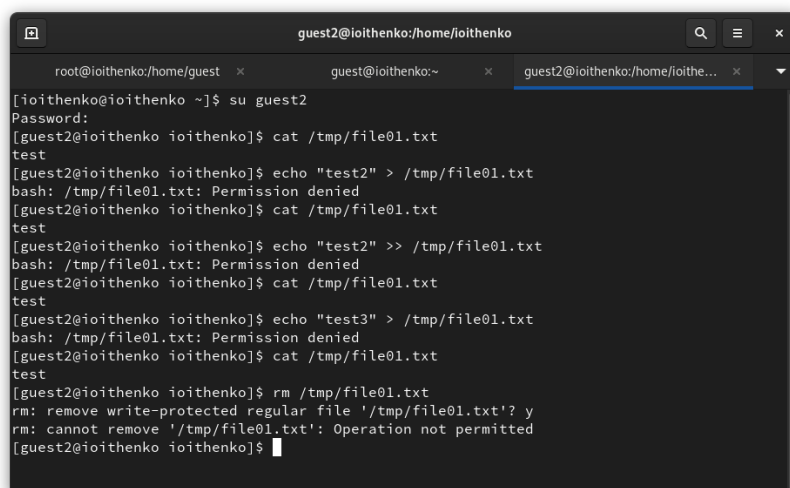
пользователей «все остальные»: `ls -l /tmp/file01.txt chmod o+rw /tmp/file01.txt ls -l /tmp/file01.txt` (рис. 2.12).



```
guest@ioithenko:~  
root@ioithenko/home/guest x guest@ioithenko:~ x ioithenko@ioithenko:~ x  
[guest@ioithenko ~]$ ls -l | grep tmp  
[guest@ioithenko ~]$ echo "test" > /tmp/file01.txt  
[guest@ioithenko ~]$ ls -l /tmp/file01.txt  
-rw-r--r--. 1 guest guest 5 Apr 12 14:35 /tmp/file01.txt  
[guest@ioithenko ~]$ chmod o+rw /tmp/file01.txt  
[guest@ioithenko ~]$ ls -l /tmp/file01.txt  
-rw-r--rw-. 1 guest guest 5 Apr 12 14:35 /tmp/file01.txt  
[guest@ioithenko ~]$
```

Рис. 2.12: Права на файл

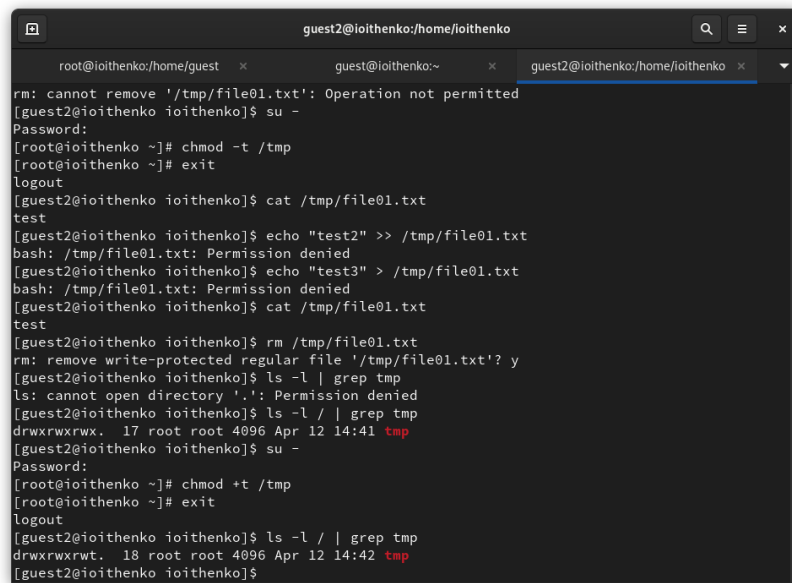
От пользователя `guest2` (не являющегося владельцем) попробуем прочитать файл `/tmp/file01.txt`: `cat /tmp/file01.txt` От пользователя `guest2` попробуем дозаписать в файл `/tmp/file01.txt` слово `test2` командой `echo "test2" > /tmp/file01.txt` Удалось ли вам выполнить операцию? Не удалось. Проверим содержимое файла командой `cat /tmp/file01.txt` От пользователя `guest2` попробуем 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? Не удалось. Проверим содержимое файла командой `cat /tmp/file01.txt` От пользователя `guest2` попробуем удалить файл `/tmp/file01.txt` командой `rm /tmp/file01.txt` Удалось ли вам удалить файл? Не удалось (рис. 2.13).



```
guest2@ioithenko:/home/ioithenko
root@ioithenko:/home/guest x guest@ioithenko:~ x guest2@ioithenko:/home/ioithe... x
[ioithenko@ioithenko ~]$ su guest2
Password:
[guest2@ioithenko ioithenko]$ cat /tmp/file01.txt
test
[guest2@ioithenko ioithenko]$ echo "test2" > /tmp/file01.txt
bash: /tmp/file01.txt: Permission denied
[guest2@ioithenko ioithenko]$ cat /tmp/file01.txt
test
[guest2@ioithenko ioithenko]$ echo "test2" >> /tmp/file01.txt
bash: /tmp/file01.txt: Permission denied
[guest2@ioithenko ioithenko]$ cat /tmp/file01.txt
test
[guest2@ioithenko ioithenko]$ echo "test3" > /tmp/file01.txt
bash: /tmp/file01.txt: Permission denied
[guest2@ioithenko ioithenko]$ cat /tmp/file01.txt
test
[guest2@ioithenko ioithenko]$ rm /tmp/file01.txt
rm: remove write-protected regular file '/tmp/file01.txt'? y
rm: cannot remove '/tmp/file01.txt': Operation not permitted
[guest2@ioithenko ioithenko]$
```

Рис. 2.13: Проверка атрибута

Повысим свои права до суперпользователя следующей командой `su` - и выполним после этого команду, снимающую атрибут `t` (Sticky-бит) с директории `/tmp`: `chmod -t /tmp` Покинем режим суперпользователя командой `exit` От пользователя `guest2` проверим, что атрибута `t` у директории `/tmp` нет: `ls -l / | grep tmp` Повторим предыдущие шаги. Удалось ли вам удалить файл от имени пользователя, не являющегося его владельцем? Удалось выполнить только удаление файла, записать и дозаписать не получилось. Повысим свои права до суперпользователя и вернем атрибут `t` на директорию `/tmp` (рис. 2.14): `su - chmod +t /tmp exit`



```
guest2@ioithenko:/home/ioithenko
root@ioithenko/home/guest x guest@ioithenko:~ x guest2@ioithenko:/home/ioithenko x
rm: cannot remove '/tmp/file01.txt': Operation not permitted
[guest2@ioithenko ioithenko]$ su -
Password:
[root@ioithenko ~]# chmod -t /tmp
[root@ioithenko ~]# exit
logout
[guest2@ioithenko ioithenko]$ cat /tmp/file01.txt
test
[guest2@ioithenko ioithenko]$ echo "test2" >> /tmp/file01.txt
bash: /tmp/file01.txt: Permission denied
[guest2@ioithenko ioithenko]$ echo "test3" > /tmp/file01.txt
bash: /tmp/file01.txt: Permission denied
[guest2@ioithenko ioithenko]$ cat /tmp/file01.txt
test
[guest2@ioithenko ioithenko]$ rm /tmp/file01.txt
rm: remove write-protected regular file '/tmp/file01.txt'? y
[guest2@ioithenko ioithenko]$ ls -l | grep tmp
ls: cannot open directory '.': Permission denied
[guest2@ioithenko ioithenko]$ ls -l / | grep tmp
drwxrwxrwx. 17 root root 4096 Apr 12 14:41 tmp
[guest2@ioithenko ioithenko]$ su -
Password:
[root@ioithenko ~]# chmod +t /tmp
[root@ioithenko ~]# exit
logout
[guest2@ioithenko ioithenko]$ ls -l / | grep tmp
drwxrwxrwt. 18 root root 4096 Apr 12 14:42 tmp
[guest2@ioithenko ioithenko]$
```

Рис. 2.14: Проверка снятия атрибута

3 Выводы

В ходе лабораторной работы я изучила механизмы изменения идентификаторов, применения SetUID- и Sticky-битов. Получила практические навыки работы в консоли с дополнительными атрибутами. Рассмотрела работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

Список литературы

1. Кулябов Д.С., Королькова А.В., Геворкян М.Н. Информационная безопасность компьютерных сетей. Лабораторные работы, учебное пособие. Москва: РУДН, 2015. 64 с.