

Отчёт по лабораторной работе №5

Дисциплина: Информационная безопасность

Выполнил: Танрибергенов Эльдар

Содержание

1	Цель работы	5
2	Задания	6
3	Указания к работе	7
4	Выполнение лабораторной работы	8
5	Выводы	19

Список иллюстраций

3.1	Подготовка лабораторного стенда	7
4.1	Программа, выводющая UID и GID	8
4.2	Компилирование программы	8
4.3	Выполнение программы	9
4.4	Системная программа id	9
4.5	Добавление в программу действительных идентификаторов	9
4.6	Компилирование программы	9
4.7	Запуск программы	10
4.8	Смена владельца файла и добавление SetUID-бита	10
4.9	Проверка добавления SUID-бита	10
4.10	Запуск программы simpleid2	10
4.11	Запуск системной программы id	11
4.12	Добавление программе SetGID-бит	11
4.13	Запуск программы simpleid2 после добавления SGID-бита	11
4.14	Программа, считывающая и выводющая в консоль содержимое файла	11
4.15	Компилирование программы	12
4.16	Смена владельца файла и разрешение только суперпользователю читать его	12
4.17	Проверка невозможности чтения файла пользователем guest- etanribergenov	12
4.18	Смена владельца у программы readfile и установка SetUID-бита . .	12
4.19	Проверка возможности чтения программой readfile другого файла	13
4.20	Проверка возможности чтения программой readfile файла /etc/shadow	13
4.21	Проверка наличия Sticky-бита на директории /tmp	14
4.22	Создание файла со словом «test» в директории /tmp от имени поль- зователя guest-etanribergenov	14
4.23	Просмотр атрибутов и разрешение на чтение и запись для категории пользователей «остальные»	14
4.24	Попытка чтения файла пользователем, не являющимся его владель- цем	15
4.25	Попытка записи в файл пользователем, не являющимся его вла- дельцем	15
4.26	Проверка содержимого файла	15
4.27	Попытка перезаписи файла пользователем, не являющимся его вла- дельцем	15
4.28	Проверка содержимого файла	16

4.29 Попытка удаления файла пользователем, не являющимся его владельцем	16
4.30 Снятие Sticky-бита с директории /tmp суперпользователем	16
4.31 Выход из режима суперпользователя	17
4.32 Проверка отсутствия Sticky-бита у директории /tmp	17
4.33 Попытка удаления файла пользователем, не являющимся его владельцем	17
4.34 Возвращение Sticky-бита на директорию /tmp	18

1 Цель работы

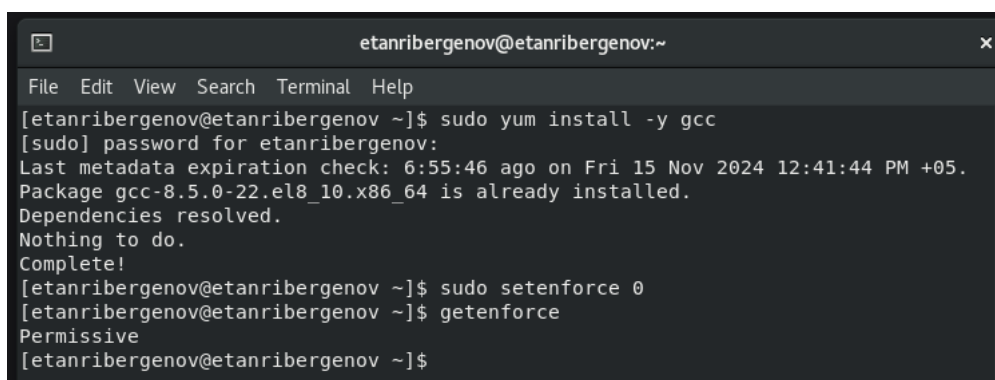
Изучение механизмов изменения идентификаторов, применения *SetUID*- и *Sticky*-битов. Получение практических навыков работы в консоли с дополнительными атрибутами. Рассмотрение работы механизма смены идентификатора процессов пользователей, а также влияние бита *Sticky* на запись и удаление файлов.

2 Задания

1. Исследовать *SetUID*- и *SetGID*-биты.
2. Исследовать *Sticky*-бит.

3 Указания к работе

Подготовка лабораторного стенда. Помимо прав администратора для выполнения части заданий потребуются средства разработки приложений. В частности, при подготовке стенда следует убедиться, что в системе установлен компилятор *gcc*. Так как программы с установленным битом *SetUID* могут представлять большую брешь в системе безопасности, в современных системах используются дополнительные механизмы защиты. Проследите, чтобы система защиты *SELinux* не мешала выполнению заданий работы. Отключите систему запретов до очередной перезагрузки системы командой *setenforce 0*. После этого команда *getenforce* должна выводить *Permissive*.



```
etanribergenov@etanribergenov:~  
File Edit View Search Terminal Help  
[etanribergenov@etanribergenov ~]$ sudo yum install -y gcc  
[sudo] password for etanribergenov:  
Last metadata expiration check: 6:55:46 ago on Fri 15 Nov 2024 12:41:44 PM +05.  
Package gcc-8.5.0-22.el8_10.x86_64 is already installed.  
Dependencies resolved.  
Nothing to do.  
Complete!  
[etanribergenov@etanribergenov ~]$ sudo setenforce 0  
[etanribergenov@etanribergenov ~]$ getenforce  
Permissive  
[etanribergenov@etanribergenov ~]$
```

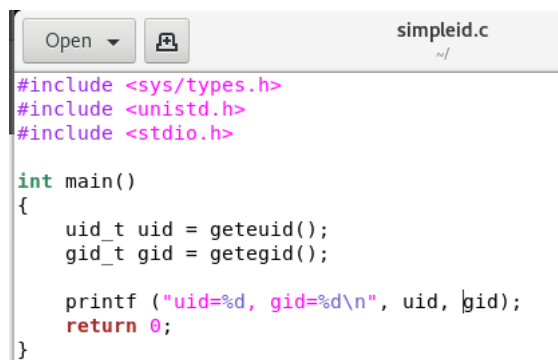
Рис. 3.1: Подготовка лабораторного стенда

4 Выполнение лабораторной работы

1. Исследование *SetUID* и *SetGID*-битов

1.1. Вошёл в систему от имени пользователя *guest-etanribergenov*.

1.2. Создал программу *simpleid.c*



```
Open simpleid.c
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>

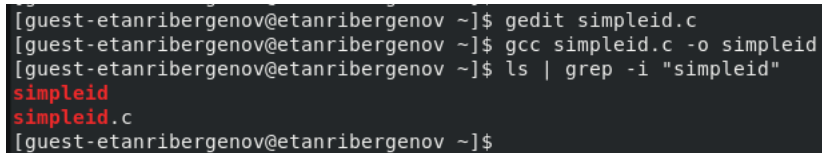
int main()
{
    uid_t uid = geteuid();
    gid_t gid = getegid();

    printf ("uid=%d, gid=%d\n", uid, gid);
    return 0;
}
```

Рис. 4.1: Программа, выводящая UID и GID

1.3. Скомпилировал программу и убедился, что файл программы создан

`gcc simpleid.c -o simpleid`



```
[guest-etanribergenov@etanribergenov ~]$ gedit simpleid.c
[guest-etanribergenov@etanribergenov ~]$ gcc simpleid.c -o simpleid
[guest-etanribergenov@etanribergenov ~]$ ls | grep -i "simpleid"
simpleid
simpleid.c
[guest-etanribergenov@etanribergenov ~]$
```

Рис. 4.2: Компилирование программы

1.4. Выполнил программу *simpleid*:

`./simpleid`


```
[guest-etanribergenov@etanribergenov ~]$ ./simpleid
uid=1001, gid=1001
[guest-etanribergenov@etanribergenov ~]$
```

Рис. 4.3: Выполнение программы

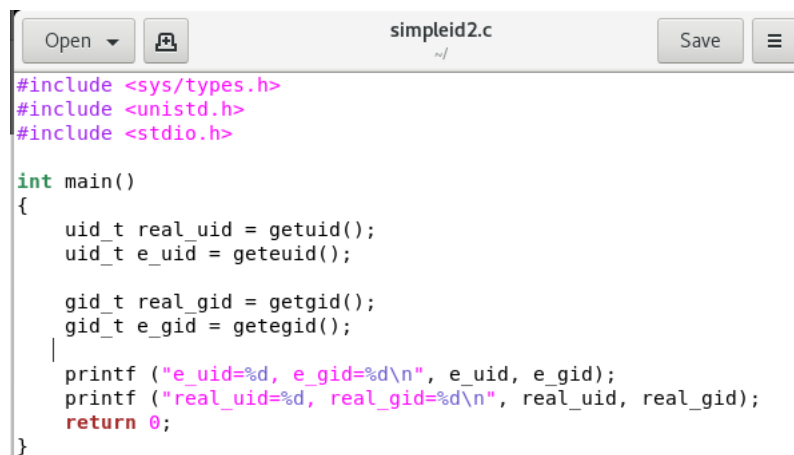
1.5. Выполнил системную программу *id*:

```
[guest-etanribergenov@etanribergenov ~]$ id
uid=1001(guest-etanribergenov) gid=1001(guest-etanribergenov) groups=1001(guest-etanribergenov)
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest-etanribergenov@etanribergenov ~]$
```

Рис. 4.4: Системная программа *id*

Выведенные данные совпадают.

1.6. Усложнил программу, добавив вывод действительных идентификаторов, получившуюся программу назвал *simpleid2.c* :



```

Open  simpleid2.c  Save
#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;
}

```

Рис. 4.5: Добавление в программу действительных идентификаторов

1.7. Скомпилируйте и запустите *simpleid2.c*:

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

```
[guest-etanribergenov@etanribergenov ~]$ gcc simpleid2.c -o simpleid2
```

Рис. 4.6: Компилирование программы

```
[guest-etanribergenov@etanribergenov ~]$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
[guest-etanribergenov@etanribergenov ~]$
```

Рис. 4.7: Запуск программы

1.8. От имени суперпользователя выполните команды:

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

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

```
[guest-etanribergenov@etanribergenov ~]$ su etanribergenov
Password:
[etanribergenov@etanribergenov guest-etanribergenov]$ sudo su
[sudo] password for etanribergenov:
[root@etanribergenov guest-etanribergenov]# chown root:guest-etanribergenov
/home/guest-etanribergenov/simpleid2
[root@etanribergenov guest-etanribergenov]# chmod u+s /home/guest-etanriber
genov/simpleid2
[root@etanribergenov guest-etanribergenov]#
```

Рис. 4.8: Смена владельца файла и добавление SetUID-бита

1.9. Использовал *sudo su*, чтобы повысить права пользователя до прав супер-пользователя.

1.10. Выполнил проверку правильности установки новых атрибутов и смены владельца файла *simpleid2*:

```
ls -l simpleid2
```

```
[guest-etanribergenov@etanribergenov ~]$ ls -l simpleid2
-rwsrwxr-x. 1 root guest-etanribergenov 18312 Nov 15 20:19 simpleid2
[guest-etanribergenov@etanribergenov ~]$
```

Рис. 4.9: Проверка добавления SUID-бита

1.11. Запустил *simpleid2* и *id*:

```
./simpleid2
```

```
id
```

```
[guest-etanribergenov@etanribergenov ~]$ ./simpleid2
e_uid=0, e_gid=1001
real_uid=1001, real_gid=1001
[guest-etanribergenov@etanribergenov ~]$
```

Рис. 4.10: Запуск программы simpleid2

```
[guest-etanribergenov@etanribergenov ~]$ id
uid=1001(guest-etanribergenov) gid=1001(guest-etanribergenov) groups=1001(guest-etanribergenov)
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest-etanribergenov@etanribergenov ~]$
```

Рис. 4.11: Запуск системной программы `id`

Значение `e_uid` стало 0, в то время, как настоящий идентификатор `uid` и `uid` из программы `id` вывели 1001.

1.12. Прodelал тоже самое относительно `SetGID`-бита.

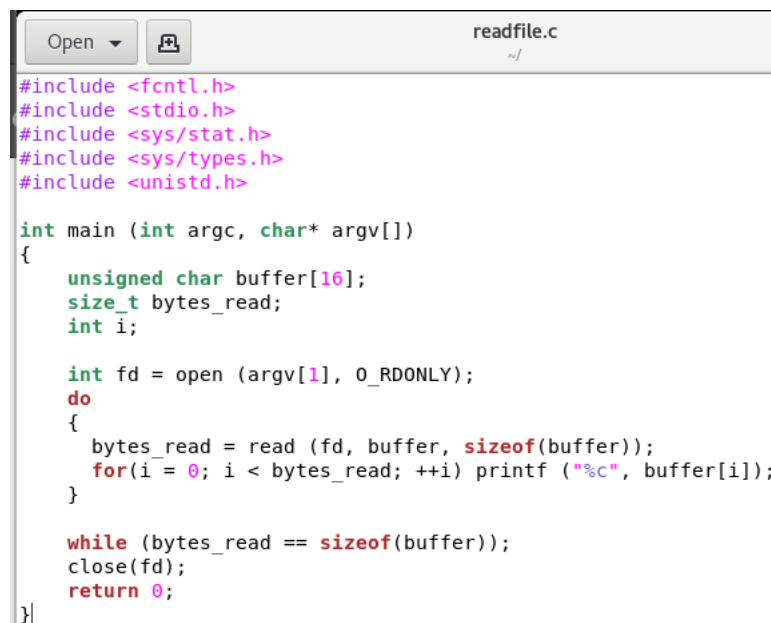
```
[root@etanribergenov guest-etanribergenov]# chmod ug+s simpleid2
[root@etanribergenov guest-etanribergenov]# ls -l simpleid2
-rwsrwsr-x. 1 root guest-etanribergenov 18312 Nov 15 20:19 simpleid2
[root@etanribergenov guest-etanribergenov]#
```

Рис. 4.12: Добавление программе `SetGID`-бит

```
[guest-etanribergenov@etanribergenov ~]$ ./simpleid2
e_uid=0, e_gid=1001
real_uid=1001, real_gid=1001
[guest-etanribergenov@etanribergenov ~]$
```

Рис. 4.13: Запуск программы `simpleid2` после добавления `SGID`-бита

1.13. Создал программу `readfile.c`



```
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;
}
```

Рис. 4.14: Программа, считывающая и выводорящая в консоль содержимое файла

1.14. Откомпилировал её.

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

```
[guest-etanribergenov@etanribergenov ~]$ gcc readfile.c -o readfile  
[guest-etanribergenov@etanribergenov ~]$
```

Рис. 4.15: Компилирование программы

1.15. Сменил владельца у файла *readfile.c* и изменил права так, чтобы только суперпользователь (*root*) мог прочитать его, а *guest-etanribergenov* не мог.

```
[root@etanribergenov guest-etanribergenov]# chown root:guest-etanribergenov  
readfile.c  
[root@etanribergenov guest-etanribergenov]# chmod 700 readfile.c  
[root@etanribergenov guest-etanribergenov]#
```

Рис. 4.16: Смена владельца файла и разрешение только суперпользователю читать его

1.16. Проверил, что пользователь *guest-etanribergenov* не может прочитать файл *readfile.c*.

```
[guest-etanribergenov@etanribergenov ~]$ cat readfile.c  
cat: readfile.c: Permission denied
```

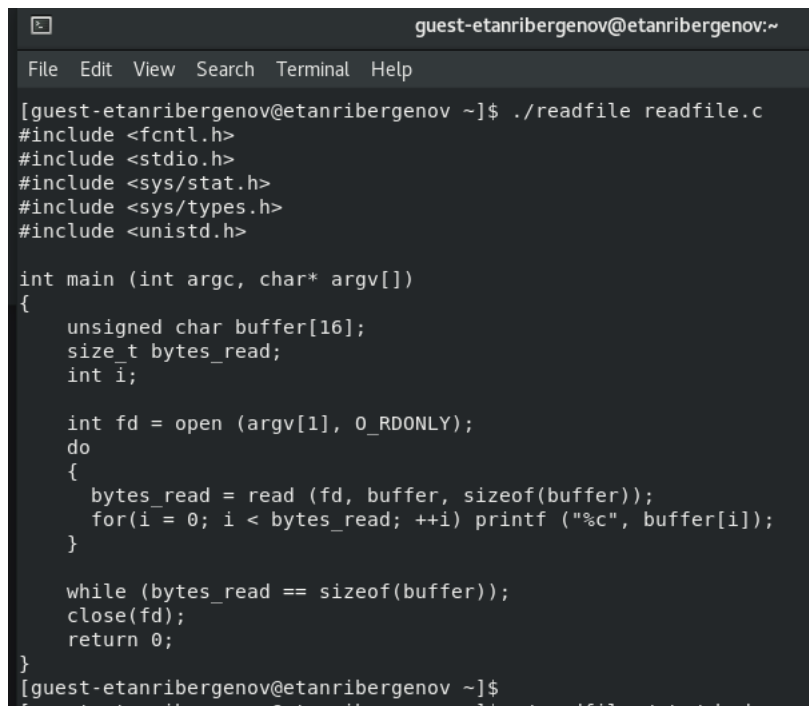
Рис. 4.17: Проверка невозможности чтения файла пользователем *guest-etanribergenov*

1.17. Сменил у программы *readfile* владельца и установил *SetUID*-бит.

```
root@etanribergenov:/home/guest-etanribergenov  
File Edit View Search Terminal Help  
[root@etanribergenov guest-etanribergenov]# chown root:guest-etanribergenov  
readfile  
[root@etanribergenov guest-etanribergenov]# chmod u+s readfile  
[root@etanribergenov guest-etanribergenov]#
```

Рис. 4.18: Смена владельца у программы *readfile* и установка *SetUID*-бита

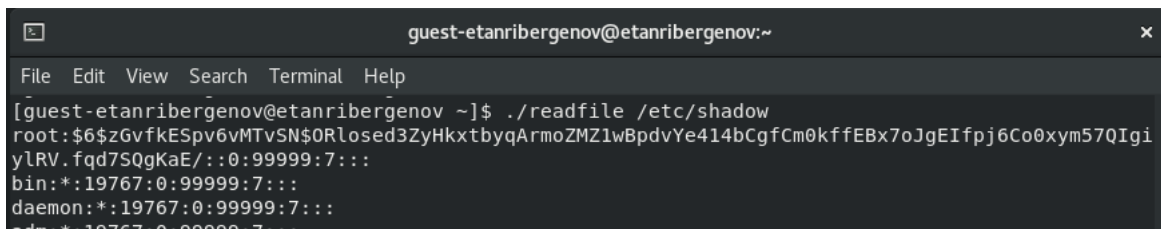
1.18. Проверил, может ли программа *readfile* прочитать файл *readfile.c* - может.



```
guest-etanribergenov@etanribergenov:~  
File Edit View Search Terminal Help  
[guest-etanribergenov@etanribergenov ~]$ ./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;  
}
```

Рис. 4.19: Проверка возможности чтения программой readfile другого файла

1.19. Проверил, может ли программа *readfile* прочитать файл */etc/shadow* - может. Это возможно потому, что пользователем программы стал суперпользователь (*root*).



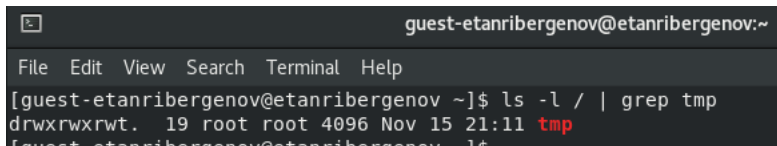
```
guest-etanribergenov@etanribergenov:~  
File Edit View Search Terminal Help  
[guest-etanribergenov@etanribergenov ~]$ ./readfile /etc/shadow  
root:$6$zGvfkESpv6vMTvSN$0Rlosed3ZyHkxtbyqArmoZMZ1wBpdvYe414bCgfCm0kffEBx7oJgEIfpj6Co0xym570Igi  
yLRV.fqd7S0gKaE/::0:99999:7:::  
bin*:19767:0:99999:7:::  
daemon*:19767:0:99999:7:::  
adm*:19767:0:99999:7:::
```

Рис. 4.20: Проверка возможности чтения программой readfile файла */etc/shadow*

2. Исследование Sticky-бита

2.1. Выяснил, установлен ли атрибут *Sticky* на директории */tmp*, для чего выполнил команду

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

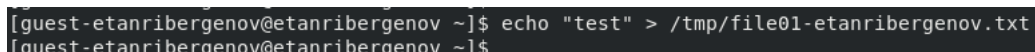


```
guest-etanribergenov@etanribergenov:~  
File Edit View Search Terminal Help  
[guest-etanribergenov@etanribergenov ~]$ ls -l / | grep tmp  
drwxrwxrwt. 19 root root 4096 Nov 15 21:11 tmp  
[guest-etanribergenov@etanribergenov ~]$
```

Рис. 4.21: Проверка наличия Sticky-бита на директории /tmp

2.2. От имени пользователя *guest-etanribergenov* создал файл *file01-etanribergenov.txt* в директории */tmp* со словом *test*:

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

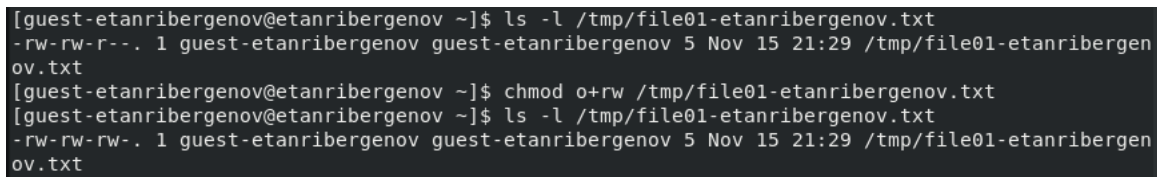


```
[guest-etanribergenov@etanribergenov ~]$ echo "test" > /tmp/file01-etanribergenov.txt  
[guest-etanribergenov@etanribergenov ~]$
```

Рис. 4.22: Создание файла со словом «test» в директории /tmp от имени пользователя *guest-etanribergenov*

2.3. Просмотрел атрибуты у только что созданного файла и разрешил чтение и запись для категории пользователей «все остальные»:

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

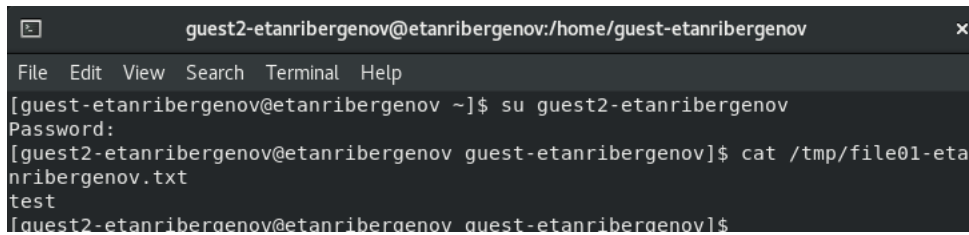


```
[guest-etanribergenov@etanribergenov ~]$ ls -l /tmp/file01-etanribergenov.txt  
-rw-rw-r--. 1 guest-etanribergenov guest-etanribergenov 5 Nov 15 21:29 /tmp/file01-etanribergenov.txt  
[guest-etanribergenov@etanribergenov ~]$ chmod o+rw /tmp/file01-etanribergenov.txt  
[guest-etanribergenov@etanribergenov ~]$ ls -l /tmp/file01-etanribergenov.txt  
-rw-rw-rw-. 1 guest-etanribergenov guest-etanribergenov 5 Nov 15 21:29 /tmp/file01-etanribergenov.txt  
[guest-etanribergenov@etanribergenov ~]$
```

Рис. 4.23: Просмотр атрибутов и разрешение на чтение и запись для категории пользователей «остальные»

2.4. От пользователя *guest2-etanribergenov* (не являющегося владельцем) попробуйте прочитать файл */tmp/file01-etanribergenov.txt*:

```
cat /tmp/file01-etanribergenov.txt
```

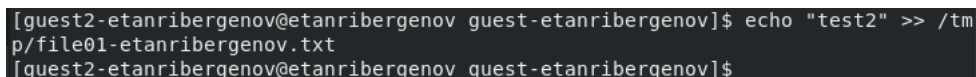
A terminal window titled "guest2-etanribergenov@etanribergenov:/home/guest-etanribergenov" with a menu bar (File, Edit, View, Search, Terminal, Help). The user switches to the "guest2-etanribergenov" user and attempts to read the file "/tmp/file01-etanribergenov.txt". The output shows the file's content, "test", indicating successful access despite not being the owner.

```
guest2-etanribergenov@etanribergenov:/home/guest-etanribergenov
File Edit View Search Terminal Help
[guest-etanribergenov@etanribergenov ~]$ su guest2-etanribergenov
Password:
[guest2-etanribergenov@etanribergenov guest-etanribergenov]$ cat /tmp/file01-etanribergenov.txt
test
[guest2-etanribergenov@etanribergenov guest-etanribergenov]$
```

Рис. 4.24: Попытка чтения файла пользователем, не являющимся его владельцем

2.5. От пользователя *guest2-etanribergenov* попробуйте дозаписать в файл */tmp/file01-etanribergenov.txt* слово «test2» командой

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

A terminal window showing the user "guest2-etanribergenov" appending the string "test2" to the file "/tmp/file01-etanribergenov.txt" using the "echo" command with the ">>" operator. The command is successful.

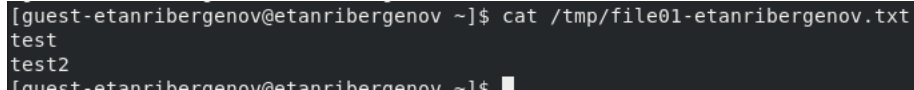
```
[guest2-etanribergenov@etanribergenov guest-etanribergenov]$ echo "test2" >> /tmp/file01-etanribergenov.txt
[guest2-etanribergenov@etanribergenov guest-etanribergenov]$
```

Рис. 4.25: Попытка записи в файл пользователем, не являющимся его владельцем

Удалось выполнить операцию.

2.6. Проверил содержимое файла командой

```
cat /tmp/file01-etanribergenov.txt
```

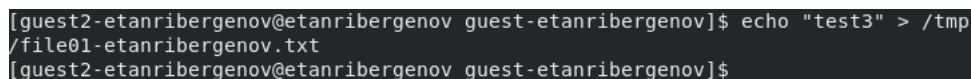
A terminal window showing the output of the "cat" command for the file "/tmp/file01-etanribergenov.txt". The output displays the words "test" and "test2" on separate lines, confirming the successful append operation.

```
[guest-etanribergenov@etanribergenov ~]$ cat /tmp/file01-etanribergenov.txt
test
test2
[guest-etanribergenov@etanribergenov ~]$
```

Рис. 4.26: Проверка содержимого файла

2.7. От пользователя *guest2-etanribergenov* попробуйте записать в файл */tmp/file01-etanribergenov.txt* слово «test3», стерев при этом всю имеющуюся в файле информацию командой

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

A terminal window showing the user "guest2-etanribergenov" overwriting the file "/tmp/file01-etanribergenov.txt" with the string "test3" using the "echo" command with the ">" operator. The command is successful.

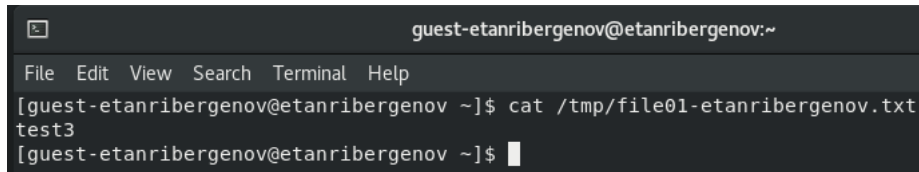
```
[guest2-etanribergenov@etanribergenov guest-etanribergenov]$ echo "test3" > /tmp/file01-etanribergenov.txt
[guest2-etanribergenov@etanribergenov guest-etanribergenov]$
```

Рис. 4.27: Попытка перезаписи файла пользователем, не являющимся его владельцем

Удалось выполнить операцию.

2.8. Проверил содержимое файла командой

```
cat /tmp/file01-etanribergenov.txt
```



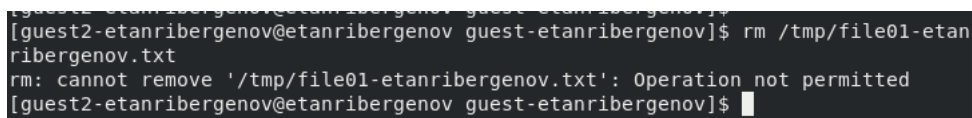
```
guest-etanribergenov@etanribergenov:~  
File Edit View Search Terminal Help  
[guest-etanribergenov@etanribergenov ~]$ cat /tmp/file01-etanribergenov.txt  
test3  
[guest-etanribergenov@etanribergenov ~]$
```

Рис. 4.28: Проверка содержимого файла

2.9. От пользователя *guest2-etanribergenov* попробуйте удалить файл */tmp/file01-etanribergenov.txt* командой

```
rm /tmp/file01-etanribergenov.txt
```

Не удалось удалить файл.



```
[guest2-etanribergenov@etanribergenov ~]$ rm /tmp/file01-etanribergenov.txt  
rm: cannot remove '/tmp/file01-etanribergenov.txt': Operation not permitted  
[guest2-etanribergenov@etanribergenov ~]$
```

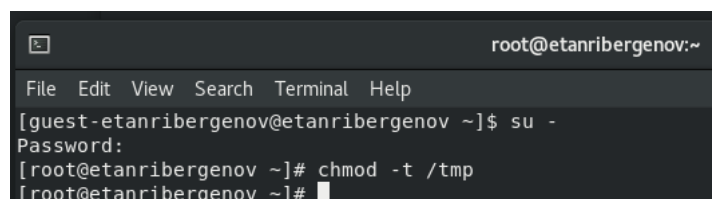
Рис. 4.29: Попытка удаления файла пользователем, не являющимся его владельцем

2.10. Повысил свои права до суперпользователя следующей командой

```
su -
```

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

```
chmod -t /tmp
```



```
root@etanribergenov:~  
File Edit View Search Terminal Help  
[guest-etanribergenov@etanribergenov ~]$ su -  
Password:  
[root@etanribergenov ~]# chmod -t /tmp  
[root@etanribergenov ~]#
```

Рис. 4.30: Снятие Sticky-бита с директории */tmp* суперпользователем

2.11. Покинул режим суперпользователя командой

```
exit
```



```
[root@etanribergenov ~]# exit
logout
[guest-etanribergenov@etanribergenov ~]$
```

Рис. 4.31: Выход из режима суперпользователя

2.12. От пользователя *guest2-etanribergenov* проверил, что атрибута *t* у директории */tmp* нет:

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

```
[guest2-etanribergenov@etanribergenov guest-etanribergenov]$ ls -l / | grep tmp
drwxrwxrwx. 19 root root 4096 Nov 15 21:46 tmp
[guest2-etanribergenov@etanribergenov guest-etanribergenov]$
```

Рис. 4.32: Проверка отсутствия Sticky-бита у директории */tmp*

2.13. Повторил предыдущие шаги. Изменений нет - файл всё также можно дополнить и переписать.

2.14. Удалось удалить файл от имени пользователя, не являющегося его владельцем. Это случилось из-за того, что ранее был снят *Sticky*-бит с директории, защищавший файлы от неразрешённых действий.

```
[guest2-etanribergenov@etanribergenov guest-etanribergenov]$
[guest2-etanribergenov@etanribergenov guest-etanribergenov]$ echo "test5" > /tmp
/file01-etanribergenov.txt
[guest2-etanribergenov@etanribergenov guest-etanribergenov]$
[guest2-etanribergenov@etanribergenov guest-etanribergenov]$ rm /tmp/file01-etan
ribergenov.txt
[guest2-etanribergenov@etanribergenov guest-etanribergenov]$
```

Рис. 4.33: Попытка удаления файла пользователем, не являющимся его владельцем

2.15. Повысил свои права до суперпользователя и вернул атрибут *t* на директорию */tmp*:

```
su -
chmod +t /tmp
exit
```

```
[guest-etanribergenov@etanribergenov ~]$ su -  
Password:  
[root@etanribergenov ~]# chmod +t /tmp  
[root@etanribergenov ~]# exit  
logout  
[guest-etanribergenov@etanribergenov ~]$
```

Рис. 4.34: Возвращение Sticky-бита на директорию /tmp

5 Выводы

В результате выполнения работы я изучил механизмы изменения идентификаторов, применения *SetUID*- и *Sticky*-битов. Получил практические навыки работы в консоли с дополнительными атрибутами. Рассмотрел работы механизма смены идентификатора процессов пользователей, а также влияние бита *Sticky* на запись и удаление файлов.