

# **Отчет по лабораторной работе №5**

**Дисциплина: Основы информационной безопасности**

Иванов Сергей Владимирович

# Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
3	Вывод	13

## Список иллюстраций

2.1	Создание программы . . . . .	5
2.2	Содержимое программы . . . . .	6
2.3	Компиляция и выполнение . . . . .	6
2.4	Системный id . . . . .	6
2.5	Усложнение программы . . . . .	7
2.6	Компиляция и запуск . . . . .	7
2.7	Смена владельца директории и прав доступа файла . . . . .	8
2.8	Проверка смены атрибутов . . . . .	8
2.9	Запуск программы и id . . . . .	8
2.10	Создание и компиляция программы . . . . .	9
2.11	Смена прав и владельца readfile.c . . . . .	9
2.12	Проверка чтения файла . . . . .	9
2.13	Проверка чтения файла программой . . . . .	10
2.14	Проверка атрибута Sticky . . . . .	10
2.15	Создание файла, изменение прав доступа . . . . .	10
2.16	Попытка чтения и записи . . . . .	11
2.17	Попытка записи . . . . .	11
2.18	Попытка удаления . . . . .	11
2.19	Снимаем атрибут Sticky . . . . .	12
2.20	Проверим снятие атрибута . . . . .	12
2.21	Проверка предыдущих шагов . . . . .	12
2.22	Возвращение атрибута t . . . . .	12

# 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

Создаем программу simpleid.c (рис. 1)

```
[guest@svivanov ~]$ touch simpleid.c  
[guest@svivanov ~]$ vim simpleid.c  
[guest@svivanov ~]$
```

Рис. 2.1: Создание программы

Содержимое программы (рис. 2).

```
#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;  
}
```

```

#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. 2.2: Содержимое программы

Скомпилируем программу и выполним её (рис. 3).

```

[guest@svivanov ~]$ gcc simpleid.c -o simpleid
[guest@svivanov ~]$ ./simpleid
uid=1001, gid=1001

```

Рис. 2.3: Компиляция и выполнение

Выполним системную программу id. Сравним полученный результат с данными предыдущего пункта и видим что они совпадают (рис. 4).

```

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

```

Рис. 2.4: Системный id

Усложним программу и назовем её simpleid2.c (рис. 5).

```

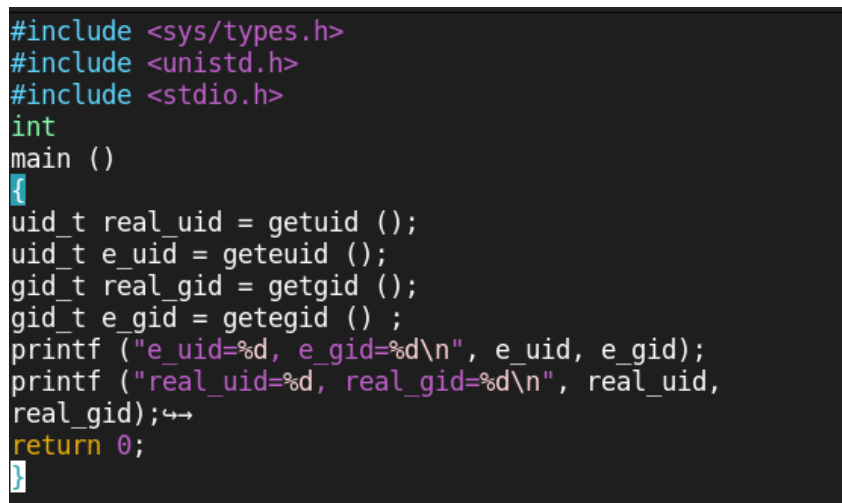
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>
int
main ()
{

```

```

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;
}

```



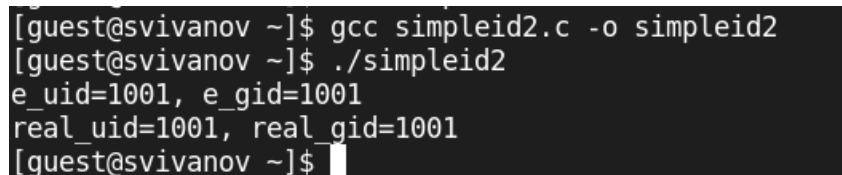
```

#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. 2.5: Усложнение программы

Скомпилируем и запустим программу (рис. 6).



```

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

```

Рис. 2.6: Компиляция и запуск

От имени суперпользователя выполним команды: `chown root:guest /home/guest/simpleid2` `chmod u+s /home/guest/simpleid2`. (рис. 7).

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

Рис. 2.7: Смена владельца директории и прав доступа файла

Выполним проверку правильности установки новых атрибутов и смены владельца файла simpleid2: `ls -l simpleid2`. (рис. 8).

```
[root@svivanov guest]# ls -l simpleid2
-rwsr-xr-x. 1 root guest 17704 Apr  3 14:48 simpleid2
[root@svivanov guest]#
```

Рис. 2.8: Проверка смены атрибутов

Запустим simpleid2 и id: `./simpleid2 id` Видим что вывод id более подробный (рис. 9).

```
[root@svivanov guest]# ./simpleid2
e_uid=0, e_gid=0
real uid=0, real gid=0
[root@svivanov guest]# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfi
ned_t:s0-s0:c0.c1023
[root@svivanov guest]#
```

Рис. 2.9: Запуск программы и id

Создание и компиляция программы readfile.c (рис. 10).

```
#include <fcntl.h>
#include <stdio.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>

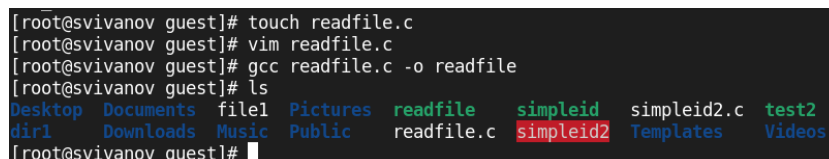
int
main (int argc, char* argv[])
{
    unsigned char buffer[16];
    size_t bytes_read;
    int i;
```



```

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;
}

```



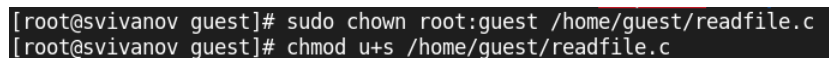
```

[root@svivanov guest]# touch readfile.c
[root@svivanov guest]# vim readfile.c
[root@svivanov guest]# gcc readfile.c -o readfile
[root@svivanov guest]# ls
Desktop  Documents  file1  Pictures  readfile  simpleid  simpleid2.c  test2
dirl     Downloads  Music  Public   readfile.c  simpleid2  Templates    Videos
[root@svivanov guest]#

```

Рис. 2.10: Создание и компиляция программы

Сменим владельца у файла readfile.c и изменим права так, чтобы только суперпользователь мог прочитать его, а guest не мог. (рис. 11).



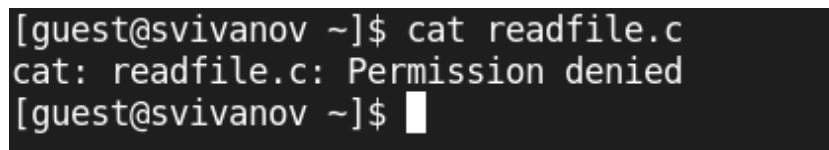
```

[root@svivanov guest]# sudo chown root:guest /home/guest/readfile.c
[root@svivanov guest]# chmod u+s /home/guest/readfile.c

```

Рис. 2.11: Смена прав и владельца readfile.c

Проверим, что пользователь guest не может прочитать файл readfile.c (рис. 12).



```

[guest@svivanov ~]$ cat readfile.c
cat: readfile.c: Permission denied
[guest@svivanov ~]$

```

Рис. 2.12: Проверка чтения файла

Проверим, может ли программа readfile прочитать файл /etc/shadow? Не может (рис. 13).



```
[guest@svivanov ~]$ su guest2
Password:
[guest2@svivanov guest]$ cat /tmp/file01.txt
test
[guest2@svivanov guest]$ echo "test2" > /tmp/file01.txt
bash: /tmp/file01.txt: Permission denied
[guest2@svivanov guest]$ cat /tmp/file01.txt
test
[guest2@svivanov guest]$ █
```

Рис. 2.16: Попытка чтения и записи

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. Операцию выполнить не удалось (рис. 17)

```
[guest2@svivanov guest]$ echo "test3" > /tmp/file01.txt
bash: /tmp/file01.txt: Permission denied
[guest2@svivanov guest]$ cat /tmp/file01.txt
test
[guest2@svivanov guest]$ █
```

Рис. 2.17: Попытка записи

От пользователя guest2 попробуем удалить файл /tmp/file01.txt командой rm /tmp/file01.txt. Не удалось удалить файл (рис. 18)

```
[guest2@svivanov guest]$ rm /tmp/file01.txt
rm: cannot remove '/tmp/file01.txt': No such file or directory
[guest2@svivanov guest]$ █
```

Рис. 2.18: Попытка удаления

Повысим свои права до суперпользователя и выполним команду, снимающую атрибут t с директории: chmod -t /tmp. Покинем режим суперпользователя командой exit (рис. 19)

```
[guest2@svivanov guest]$ su -
Password:
[root@svivanov ~]# chmod -t /tmp
[root@svivanov ~]# exit
logout
[guest2@svivanov guest]$
```

Рис. 2.19: Снимаем атрибут Sticky

От пользователя guest2 проверим, что атрибута t у директории /tmp нет: `ls -l / | grep tmp` (рис. 20)

```
[guest2@svivanov guest]$ ls -l / | grep tmp
drwxrwxrwx. 17 root root 4096 Apr  3 15:06 tmp
[guest2@svivanov guest]$
```

Рис. 2.20: Проверим снятие атрибута

Повторите предыдущие шаги. Записать в файл не получилось, но теперь стало доступно удаление. (рис. 21)

```
[guest2@svivanov guest]$ ls -l /tmp/file01
ls: cannot access '/tmp/file01': No such file or directory
[guest2@svivanov guest]$ ls -l /tmp/file01.txt
-rw-r--rw-. 1 guest guest 5 Apr  3 15:01 /tmp/file01.txt
[guest2@svivanov guest]$ cat /tmp/file01.txt
test
[guest2@svivanov guest]$ echo "test2" > /tmp/file01.txt
bash: /tmp/file01.txt: Permission denied
[guest2@svivanov guest]$ rm /tmp/file01.txt
rm: remove write-protected regular file '/tmp/file01.txt'? y
[guest2@svivanov guest]$
```

Рис. 2.21: Проверка предыдущих шагов

Повысим свои права до суперпользователя и вернём атрибут t на директорию /tmp (рис. 22)

```
[guest2@svivanov guest]$ su
Password:
[root@svivanov guest]# chmod +t /tmp
[root@svivanov guest]# exit
```

Рис. 2.22: Возвращение атрибута t

## 3 Вывод

В ходе работы были изучены механизмы изменения идентификаторов. Получены практических навыков работы с дополнительными атрибутами. Рассмотрены работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.