

# Лабораторная работа №5

## Основы информационной безопасности

---

Иванов Сергей Владимирович, НПИбд-01-23

3 апреля 2025

Российский университет дружбы народов, Москва, Россия

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.

## **Выполнение работы**

---

Создаем программу simpleid.c (рис. 1)

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

**Рис. 1:** Создание программы

# Содержимое 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;
}
~
~
```

**Рис. 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. 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. 4:** Системный id

## Усложнение программы

Усложним программу и назовем ее 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;
}
```

Рис. 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. 6:** Компиляция и запуск

## Смена владельца директории и прав доступа файла

От имени суперпользователя выполним команды: (рис. 7).

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

**Рис. 7:** Смена владельца директории и прав доступа файла

Выполним проверку правильности установки новых атрибутов и смены владельца файла `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. 8:** Проверка смены атрибутов

Запустим simpleid2 и 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. 9:** Запуск программы и id

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

```
[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
dir1     Downloads  Music  Public   readfile.c  simpleid2  Templates  Videos
[root@svivanov guest]#
```

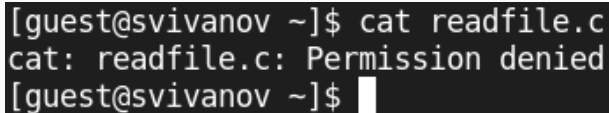
**Рис. 10:** Создание и компиляция программы

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. 11:** Смена прав и владельца readfile.c

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

A terminal window with a dark background and light gray text. The text shows a user named 'guest' at a host named 'svivanov' in the home directory '~'. They enter the command 'cat readfile.c'. The system responds with 'cat: readfile.c: Permission denied'. The prompt returns, and a white cursor is visible.

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

**Рис. 12:** Проверка чтения файла

## Проверка чтения файла программой

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

[illegible]

**Рис. 13:** Проверка чтения файла программой



## Проверка атрибута Sticky

Выясним, установлен ли атрибут Sticky на директории /tmp, для чего выполним команду (рис. 14)

```
[guest@svivanov ~]$ ls -l / | grep tmp  
drwxrwxrwt. 17 root root 4096 Apr  3 14:59 tmp  
[guest@svivanov ~]$
```

**Рис. 14:** Проверка атрибута Sticky

## Создание файла, изменение прав доступа

От имени пользователя guest создадим файл file01.txt в директории /tmp со словом test. Просмотрим атрибуты у только что созданного файла и разрешим чтение и запись для категории пользователей «все остальные» (рис. 15)

```
[guest@svivanov ~]$ ls -l / | grep tmp
drwxrwxrwt. 17 root root 4096 Apr  3 14:59 tmp
[guest@svivanov ~]$ echo "test" > /tmp/file01.txt
[guest@svivanov ~]$ ls -l /tmp/file01.txt
-rw-r--r--. 1 guest guest 5 Apr  3 15:01 /tmp/file01.txt
[guest@svivanov ~]$ chmod o+rw /tmp/file01.txt
[guest@svivanov ~]$ ls -l /tmp/file01.txt
-rw-r--rw-. 1 guest guest 5 Apr  3 15:01 /tmp/file01.txt
[guest@svivanov ~]$
```

**Рис. 15:** Создание файла, изменение прав доступа

От пользователя guest2 попробуем прочитать файл /tmp/file01.txt, дозаписать слово test2. Прочитать удалось, а записать нет (рис. 16)

```
[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. 16:** Попытка чтения и записи

От пользователя guest2 попробуем удалить файл /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. 17:** Попытка удаления

## Снимаем атрибут Sticky

Повысим свои права до суперпользователя и выполним команду, снимающую атрибут `t` с директории. 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. 18:** Снимаем атрибут Sticky

## Проверим снятие атрибута

От пользователя guest2 проверим, что атрибута t у директории /tmp нет (рис. 20)

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

**Рис. 19:** Проверим снятие атрибута

## Проверка предыдущих шагов

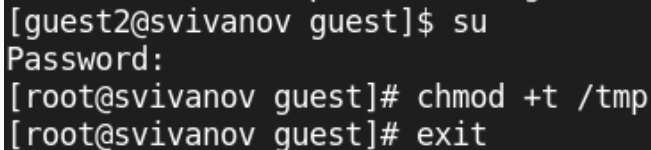
Повторите предыдущие шаги. Записать в файл не получилось, но теперь стало доступно удаление. (рис. 21)

```
[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. 20:** Проверка предыдущих шагов

## Возвращение атрибута t

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

A terminal window with a black background and white text. The prompt is [guest2@svivanov guest]\$. The user enters 'su'. The prompt changes to [root@svivanov guest]#. The user enters 'chmod +t /tmp'. The prompt changes to [root@svivanov guest]#. The user enters 'exit'.

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

Рис. 21: Возвращение атрибута t



## 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.