

# **Отчёт по лабораторной работе №5**

**Дисциплина: Основы информационной безопасности**

**Балакирева Дарья Сергеевна, НПМбд-01-196**

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Теоретическое введение</b>	<b>6</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>7</b>
3.1	Создание программы . . . . .	7
3.2	Исследование Sticky-бита . . . . .	11
<b>4</b>	<b>Выводы</b>	<b>14</b>
	<b>Список литературы</b>	<b>15</b>

## Список иллюстраций

3.1	Текст программы simpleid.c . . . . .	7
3.2	Компиляция и запуск simpleid . . . . .	8
3.3	Текст программы simpleid2.c . . . . .	8
3.4	Компиляция и запуск simpleid2 . . . . .	8
3.5	Смена владельца и установка SetUID . . . . .	9
3.6	Запуск simpleid2 . . . . .	9
3.7	SetGID-бит . . . . .	9
3.8	Текст программы readfile.c . . . . .	10
3.9	Компиляция readfile.c . . . . .	10
3.10	Запуск readfile 1 . . . . .	11
3.11	Запуск readfile 2 . . . . .	11
3.12	Создание файла file01.txt . . . . .	11
3.13	Действия над file01.txt от лица guest2 . . . . .	12
3.14	Удаление Sticky-бита . . . . .	12
3.15	Повтор действий . . . . .	13
3.16	Возвращение Sticky-бита . . . . .	13

## **Список таблиц**

# 1 Цель работы

Изучить особенности работы с дополнительными атрибутами SetUID, SetGID и Sticky битами и их влияние на работу с файлами при их наличии и отсутствии.

## 2 Теоретическое введение

SetUID, SetGID и Sticky — это специальные типы разрешений, которые позволяют задавать расширенные права доступа на файлы и каталоги.

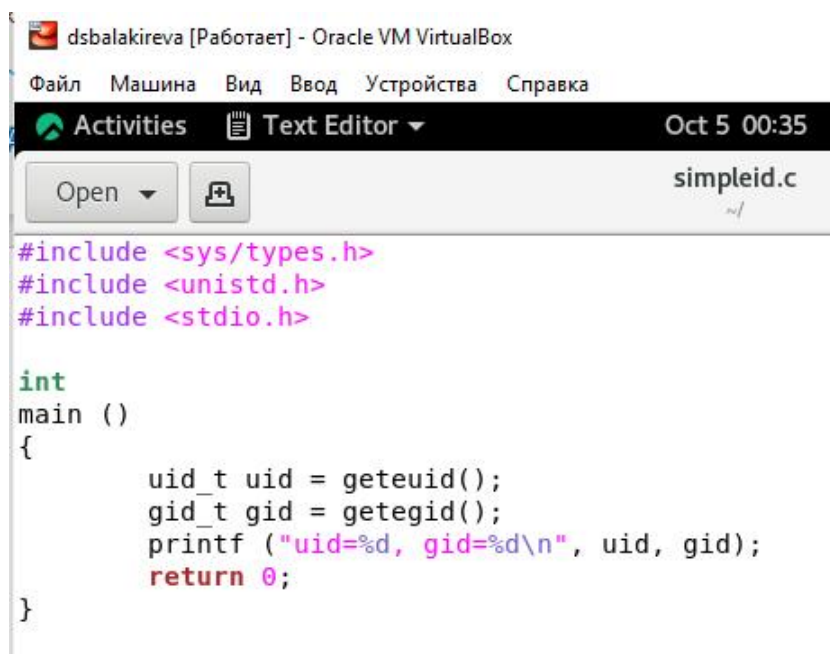
- SetUID — это бит разрешения, который позволяет пользователю запускать исполняемый файл с правами владельца этого файла. Другими словами, использование этого бита позволят поднять привилегии пользователя в случае, если это необходимо. Наличие SetUID бита выражается в том, что на месте классического бита x выставлен специальный бит s: `-rwsr-xr-x`
- SetGID — очень похож на SetUID с отличием, что файл будет запускаться от имени группы, который владеет файлом: `-rwxr-sr-x`
- Sticky — в случае, если этот бит установлен для папки, то файлы в этой папке могут быть удалены только их владельцем. Наличие этого бита показывается через букву t в конце всех прав: `drwxrwxrwx t`

Более подробно см. в [1].

## 3 Выполнение лабораторной работы

### 3.1 Создание программы

Создадим программу simpleid.c (рис. 3.1).



The screenshot shows a text editor window titled 'dsbalakireva [Работает] - Oracle VM VirtualBox'. The menu bar includes 'Файл', 'Машина', 'Вид', 'Ввод', 'Устройства', and 'Справка'. The toolbar has 'Activities', 'Text Editor', and a date/time display 'Oct 5 00:35'. The file name 'simpleid.c' is shown in the top right. The code content is as follows:

```
#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. 3.1: Текст программы simpleid.c

Скомпилируем программу с помощью команды `gcc` и убеждаемся, что файл действительно создан. Далее запускаем исполняемый файл через `./`. Вывод написанной программы совпадает с выводом команды `id` (рис 3.2).

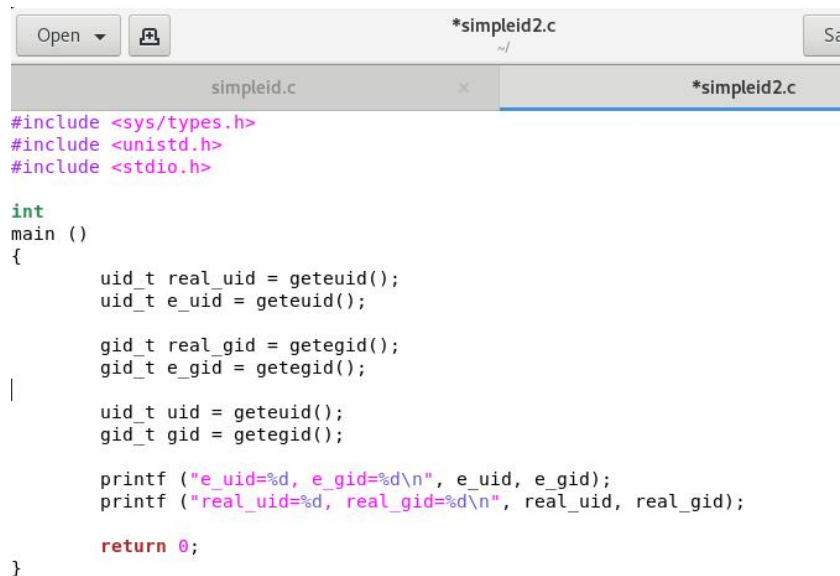
```

[guest@dsbalakireva ~]$ touch simpleid.c
[guest@dsbalakireva ~]$ gcc simpleid.c -o simpleid
[guest@dsbalakireva ~]$ ls
Desktop  Documents  Music      Public      simpleid.c  Templates
dir1     Downloads  Pictures   simpleid    simplified   Videos
[guest@dsbalakireva ~]$ ./simpleid
uid=1001, gid=1001
[guest@dsbalakireva ~]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@dsbalakireva ~]$

```

Рис. 3.2: Компиляция и запуск simpleid

Усложним программу и назовём её simpleid2.c (рис. 3.3).



```

#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>

int
main ()
{
    uid_t real_uid = geteuid();
    uid_t e_uid = geteuid();

    gid_t real_gid = getegid();
    gid_t e_gid = getegid();

    uid_t uid = geteuid();
    gid_t gid = getegid();

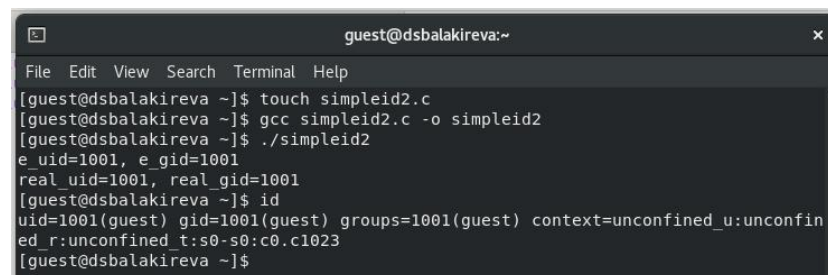
    printf ("e_uid=%d, e_gid=%d\n", e_uid, e_gid);
    printf ("real_uid=%d, real_gid=%d\n", real_uid, real_gid);

    return 0;
}

```

Рис. 3.3: Текст программы simpleid2.c

Скомпилируем и запустим файл simpleid2 (рис. 3.4).



```

guest@dsbalakireva:~$ touch simpleid2.c
guest@dsbalakireva:~$ gcc simpleid2.c -o simpleid2
guest@dsbalakireva:~$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
guest@dsbalakireva:~$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
guest@dsbalakireva:~$

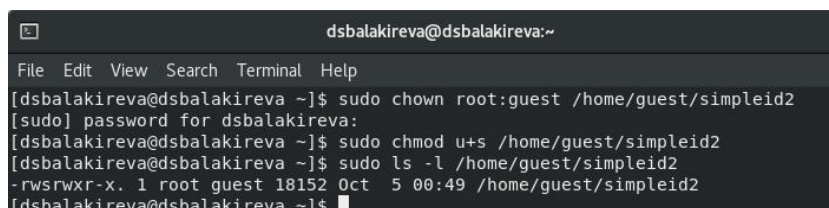
```

Рис. 3.4: Компиляция и запуск simpleid2

От имени суперпользователя сменим владельца файла simpleid2 на root и



установим SetUID-бит. Далее через команду `ls -l` видим, что бит установлен корректно (рис. 3.5)



```
dsbalakireva@dsbalakireva:~  
File Edit View Search Terminal Help  
[dsbalakireva@dsbalakireva ~]$ sudo chown root:guest /home/guest/simpleid2  
[sudo] password for dsbalakireva:  
[dsbalakireva@dsbalakireva ~]$ sudo chmod u+s /home/guest/simpleid2  
[dsbalakireva@dsbalakireva ~]$ sudo ls -l /home/guest/simpleid2  
-rwsrwxr-x. 1 root guest 18152 Oct  5 00:49 /home/guest/simpleid2  
[dsbalakireva@dsbalakireva ~]$
```

Рис. 3.5: Смена владельца и установка SetUID


Запускаем программу `simpleid2` и команду `id`. Теперь видим, что появились отличия в `uid` строках (рис. 3.6).



```
guest@dsbalakireva:~  
File Edit View Search Terminal Help  
[guest@dsbalakireva ~]$ ./simpleid2  
e_uid=0, e_gid=1001  
real_uid=0, real_gid=1001  
[guest@dsbalakireva ~]$ id  
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023  
[guest@dsbalakireva ~]$
```

Рис. 3.6: Запуск `simpleid2`

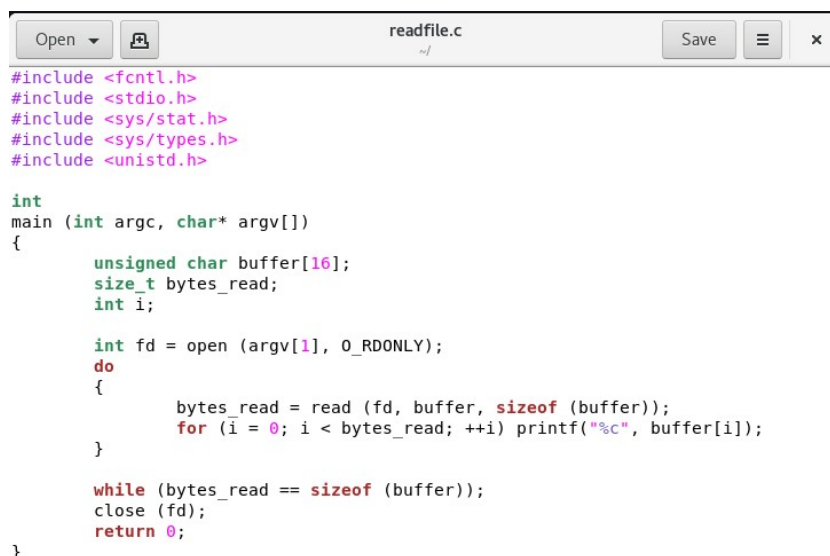
Продолываем выше описанные действия для SetGID-бита. Теперь после запуска `simpleid2` можем увидеть отличие и в `gid` строках (рис. 3.7).



```
dsbalakireva@dsbalakireva:~  
File Edit View Search Terminal Help  
[dsbalakireva@dsbalakireva ~]$ sudo chmod g+s /home/guest/simpleid2  
[sudo] password for dsbalakireva:  
Sorry, try again.  
[sudo] password for dsbalakireva:  
[dsbalakireva@dsbalakireva ~]$  
[guest@dsbalakireva ~]$ ./simpleid2  
e_uid=0, e_gid=1001  
real_uid=0, real_gid=1001  
[guest@dsbalakireva ~]$ id  
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023  
[guest@dsbalakireva ~]$
```

Рис. 3.7: SetGID-бит

Создадим программу `readfile.c` (рис. 3.8).



```

#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;

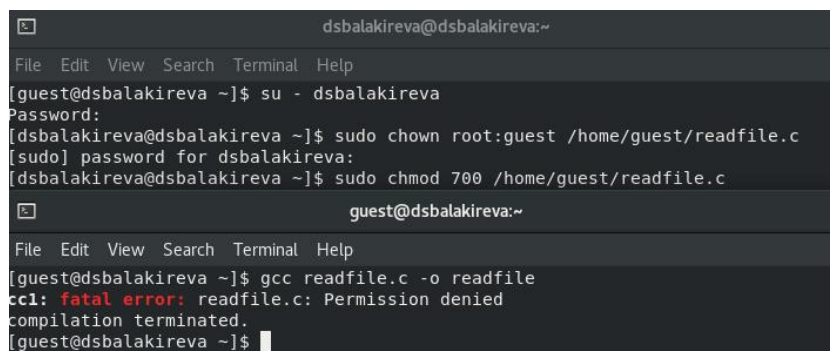
    int fd = open (argv[1], O_RDONLY);
    do
    {
        bytes_read = read (fd, buffer, sizeof (buffer));
        for (i = 0; i < bytes_read; ++i) printf ("%c", buffer[i]);
    }

    while (bytes_read == sizeof (buffer));
    close (fd);
    return 0;
}

```

Рис. 3.8: Текст программы readfile.c

Откомпилируем эту программу командой gcc. Далее меняем владельца файла readfile.c и отнимаем у пользователя guest право на чтение. При попытке прочесть файл от имени пользователя guest возникает ошибка (рис. 3.9)



```

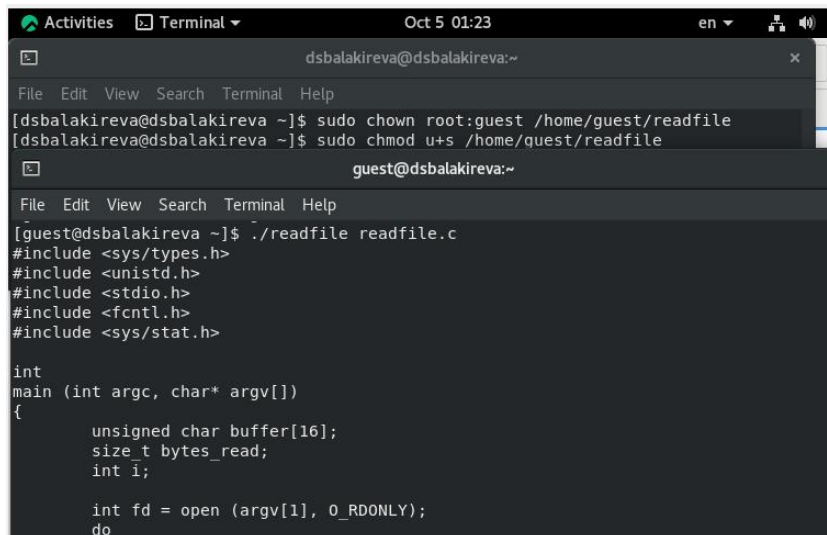
dsbalakireva@dsbalakireva:~
File Edit View Search Terminal Help
[guest@dsbalakireva ~]$ su - dsbalakireva
Password:
[dsbalakireva@dsbalakireva ~]$ sudo chown root:guest /home/guest/readfile.c
[sudo] password for dsbalakireva:
[dsbalakireva@dsbalakireva ~]$ sudo chmod 700 /home/guest/readfile.c

guest@dsbalakireva:~
File Edit View Search Terminal Help
[guest@dsbalakireva ~]$ gcc readfile.c -o readfile
cc1: fatal error: readfile.c: Permission denied
compilation terminated.
[guest@dsbalakireva ~]$

```

Рис. 3.9: Компиляция readfile.c

Меняем владельца файла readfile и устанавливаем на него SetUID-бит. Запускаем исполняемый файл и убеждаемся, что программа может прочитать файлы readfile.c и /etc/shadow (рис. 3.11) (рис. 3.10).




```
Activities Terminal Oct 5 01:23 en
dsbalakireva@dsbalakireva:~
File Edit View Search Terminal Help
[dsbalakireva@dsbalakireva ~]$ sudo chown root:guest /home/guest/readfile
[dsbalakireva@dsbalakireva ~]$ sudo chmod u+s /home/guest/readfile
guest@dsbalakireva:~
File Edit View Search Terminal Help
[guest@dsbalakireva ~]$ ./readfile readfile.c
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>
#include <fcntl.h>
#include <sys/stat.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
```

Рис. 3.10: Запуск readfile 1



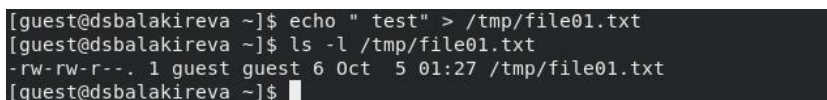
```
    {
        bytes_read = read (fd,buffer,sizeof (buffer));
        for (i = 0;i < bytes_read; ++i) printf("%c", buffer[i]);
    }

    while (bytes_read == sizeof (buffer));
    close (fd);
    return 0;
}
[guest@dsbalakireva ~]$ ./readfile /etc/shadow
root:$6$Ft6wabi9Lp3RGLGs$9yH66fdR6V9pV7rRZ0j rG.f0lXQ3HgBBZNHoeodwB6ZqMIRAqPjR0YB7jia
0o2u9rascRayUPX6mfqguInvxh.:0:99999:7:::
bin:*.18700:0:99999:7:::
daemon:*.18700:0:99999:7:::
adm:*.18700:0:99999:7:::
lp:*.18700:0:99999:7:::
sync:*.18700:0:99999:7:::
```

Рис. 3.11: Запуск readfile 2

## 3.2 Исследование Sticky-бита

Выполняя команду `ls -l` выявняем, что на каталоге `/tmp` установлен Sticky-бит. Это видно, т.к. в конце написана `t`. Далее от имени пользователя `guest` создаём файл `/tmp/file01.txt`. Потом просматриваем атрибуты только что созданного файла и даём всем пользователям право на чтение и запись (рис. 3.12).



```
[guest@dsbalakireva ~]$ echo " test" > /tmp/file01.txt
[guest@dsbalakireva ~]$ ls -l /tmp/file01.txt
-rw-rw-r--. 1 guest guest 6 Oct  5 01:27 /tmp/file01.txt
[guest@dsbalakireva ~]$
```

Рис. 3.12: Создание файла file01.txt

От имени пользователя guest2 читаем файл file01.txt командой cat. Далее успешно дозаписываем в конец файла строку “test2”, а затем успешно перезаписываем содержимое, меняя его на строку “test3”. Однако при попытке удалить файл возникла ошибка (рис. 3.13).

```
[guest@dsbalakireva ~]$ su - guest2
Password:
[guest2@dsbalakireva ~]$ cat /tmp/file01.txt
test
[guest2@dsbalakireva ~]$ echo "test2" >> /tmp/file01.txt
[guest2@dsbalakireva ~]$ cat /tmp/file01.txt
test
test2
[guest2@dsbalakireva ~]$ echo "test3" > /tmp/file01.txt
[guest2@dsbalakireva ~]$ cat /tmp/file01.txt
test3
[guest2@dsbalakireva ~]$ rm /tmp/file01.txt
rm: cannot remove '/tmp/file01.txt': Operation not permitted
[guest2@dsbalakireva ~]$
```

Рис. 3.13: Действия над file01.txt от лица guest2

Временно повышаем права до суперпользователя и снимаем с директории /tmp Sticky-бит. Покидаем режим суперпользователя командой exit (рис. 3.14).

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

Рис. 3.14: Удаление Sticky-бита

Убеждаемся через команду ls -l, что Sticky-бит действительно отсутствует. Далее повторяем действия от имени пользователя guest2, описанные выше. В этот раз удалось удалить файл file01.txt даже при условии, что guest2 не является его владельцем (рис. 3.15).

```
[guest2@dsbalakireva ~]$ ls -l / | grep tmp
drwxrwxrwx. 16 root root 4096 Oct  5 01:33 tmp
[guest2@dsbalakireva ~]$ cat /tmp/file01.txt
test3
[guest2@dsbalakireva ~]$ echo "test2" >> /tmp/file01.txt
[guest2@dsbalakireva ~]$ cat /tmp/file01.txt
test3
test2
[guest2@dsbalakireva ~]$ echo "test3" > /tmp/file01.txt
[guest2@dsbalakireva ~]$ cat /tmp/file01.txt
test3
[guest2@dsbalakireva ~]$ rm /tmp/file01.txt
[guest2@dsbalakireva ~]$ ls /tmp | grep *.txt
[guest2@dsbalakireva ~]$ ls /tmp | grep file01.txt
[guest2@dsbalakireva ~]$ S
```

Рис. 3.15: Повтор действий

Временно повышаем права до суперпользователя и возвращает Sticky-бит на каталог /tmp (рис. 3.16).

```
[dsbalakireva@dsbalakireva ~]$ su -
Password:
[root@dsbalakireva ~]# chmod +t /tmp
[root@dsbalakireva ~]# exit
logout
[dsbalakireva@dsbalakireva ~]$ ls -l / | grep tmp
drwxrwxrwt. 16 root root 4096 Oct  5 01:38 tmp
[dsbalakireva@dsbalakireva ~]$
```

Рис. 3.16: Возращение Sticky-бита

## 4 Выводы

Изучила механизмы изменения идентификаторов и получила практические навыки по работе с SetUID, SetGID и Sticky битами и узнала об их особенностях и влиянии на файлы и директории.

## Список литературы

1. GNU Bash Manual [Электронный ресурс]. Free Software Foundation, 2016.  
URL: <https://www.gnu.org/software/bash/manual/>.