

# **Отчет по лабораторной работе №5**

**Дисциплина: Информационная безопасность**

**Выполнила Дяченко Злата Константиновна, НФИбд-03-18**

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Задание</b>	<b>5</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
3.1	Шаг 1 . . . . .	6
3.2	Шаг 2 . . . . .	6
3.3	Шаг 3 . . . . .	7
3.4	Шаг 4 . . . . .	7
3.5	Шаг 5 . . . . .	8
3.6	Шаг 6 . . . . .	8
3.7	Шаг 7 . . . . .	9
3.8	Шаг 8 . . . . .	9
3.9	Шаг 9 . . . . .	10
3.10	Шаг 10 . . . . .	10
3.11	Шаг 11 . . . . .	11
3.12	Шаг 12 . . . . .	11
3.13	Шаг 13 . . . . .	11
3.14	Шаг 14 . . . . .	12
3.15	Шаг 15 . . . . .	13
3.16	Шаг 16 . . . . .	13
3.17	Шаг 17 . . . . .	13
3.18	Шаг 18 . . . . .	14
<b>4</b>	<b>Выводы</b>	<b>15</b>

# List of Figures

3.1	Программа simpleid.c . . . . .	6
3.2	Компиляция программы . . . . .	7
3.3	Результат выполнения программы simpleid и системной программы id . . . . .	7
3.4	Программа simpleid.c . . . . .	8
3.5	Результат выполнения программы simpleid2 . . . . .	8
3.6	Выполнение команд . . . . .	9
3.7	Выполнение программы simpleid2 после смены владельца файла .	9
3.8	Установка SetGID-бита и выполнение программы simpleid2 . . . .	9
3.9	Программа readfile.c . . . . .	10
3.10	Новый владелец readfile.c . . . . .	10
3.11	Смена владельца и установка SetUID-бита . . . . .	11
3.12	Попытка прочитать readfile.c . . . . .	11
3.13	Попытка прочитать /etc/shadow . . . . .	12
3.14	Создание файла и установка атрибутов . . . . .	12
3.15	Изменение файла /tmp/file01.txt от имени пользователя guest2 . .	13
3.16	Снятие Sticky-бита . . . . .	13
3.17	Удаление файла от имени guest2 . . . . .	14
3.18	Возвращение атрибута t . . . . .	14

# 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 Задание

Создать программы и исследовать SetUID- и Sticky-биты.

## 3 Выполнение лабораторной работы

### 3.1 Шаг 1

Вошла в систему от имени пользователя guest и создала программу simpleid.c, код которой представлен на Рисунке 1 (рис. 3.1)



```
#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;
}
```

Figure 3.1: Программа simpleid.c

### 3.2 Шаг 2

Скомпилировала программу и убедилась, что файл программы создан (рис. 3.2).



Figure 3.2: Компиляция программы

### 3.3 Шаг 3

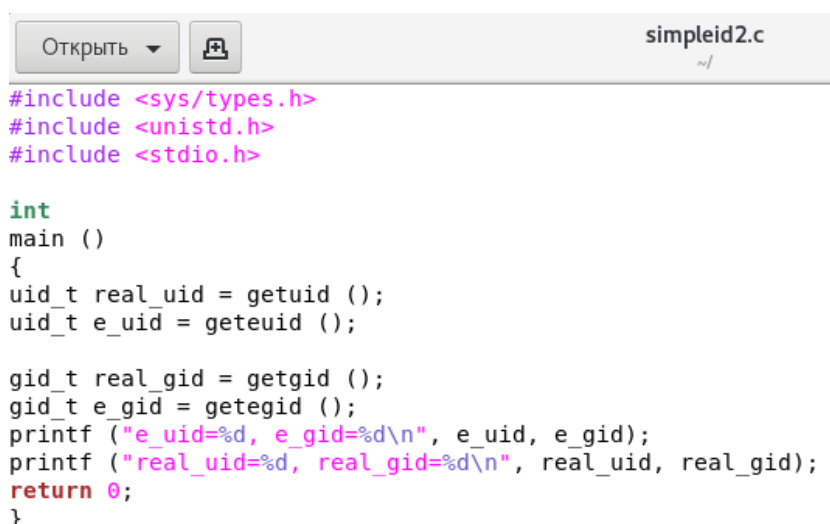
Выполнила программу simpleid и системную программу id - получила одинаковые результаты (рис. 3.3).

```
[guest@zkdyachenko ~]$ ./simpleid
uid=1001, gid=1001
[guest@zkdyachenko ~]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest),10(wheel) контекст=unconfined
_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@zkdyachenko ~]$
```

Figure 3.3: Результат выполнения программы simpleid и системной программы id

### 3.4 Шаг 4

Усложнила программу, добавив вывод действительных идентификаторов, назвала новую программу simpleid2.c, а ее код предоставлен на Рисунке 4 (рис. 3.4).



```
#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;
}
```

Figure 3.4: Программа simpleid.c

## 3.5 Шаг 5

Скомпилировала и запустила simpleid2.c (рис. 3.5). Теперь мы получаем не только указание на «настоящих» пользователя и группу, «управляющих» процессом, но и на владельца файла.

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

Figure 3.5: Результат выполнения программы simpleid2

## 3.6 Шаг 6

От имени суперпользователя выполнила команду `chown`, чтобы изменить владельца (пользователя и группу) файла `simpleid2` и `chmod u+s`, чтобы установить SetUID (рис. 3.6).



```
[guest@zkdychenko ~]$ sudo chown root:guest /home/guest/simpleid2
[sudo] пароль для guest:
[guest@zkdychenko ~]$ sudo u+s /home/guest/simpleid2
sudo: u+s: command not found
[guest@zkdychenko ~]$ sudo chmod u+s /home/guest/simpleid2
[guest@zkdychenko ~]$
```

Figure 3.6: Выполнение команд

### 3.7 Шаг 7

Выполнила проверку правильности установки новых атрибутов и смены владельца файла simpleid2. А затем запустила simpleid2 и id (рис. 3.7). Команда id показывает uid и gid текущего пользователя и группы, а программа simpleid2 показывает также uid и gid владельца файла - uid=0 так как файлом владеет пользователь root, группа совпадает с текущей.

```
[guest@zkdychenko ~]$ ls -l simpleid2
-rwsrwxr-x. 1 root guest 8576 ноя 11 18:25 simpleid2
[guest@zkdychenko ~]$ ./simpleid2
e_uid=0, e_gid=1001
real_uid=1001, real_gid=1001
[guest@zkdychenko ~]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest),10(wheel) контекст=unconfined
_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@zkdychenko ~]$
```

Figure 3.7: Выполнение программы simpleid2 после смены владельца файла

### 3.8 Шаг 8

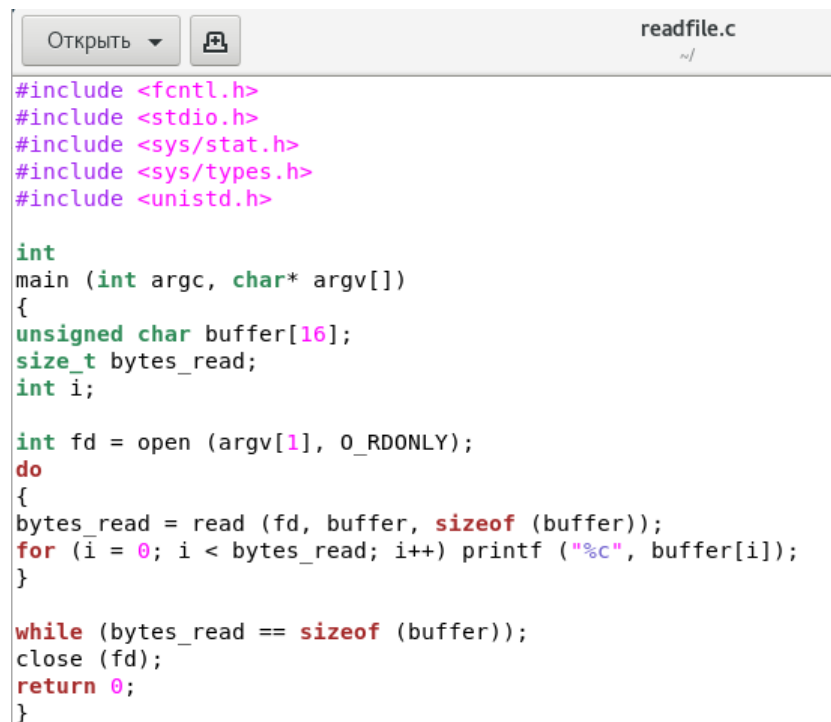
Проделала тоже самое относительно SetGID-бита (рис. 3.8).

```
[guest@zkdychenko ~]$ sudo chown guest:root /home/guest/simpleid2
[guest@zkdychenko ~]$ sudo chmod g+s /home/guest/simpleid2
[guest@zkdychenko ~]$ ls -l simpleid2
-rwxrwsr-x. 1 guest root 8576 ноя 11 18:25 simpleid2
[guest@zkdychenko ~]$ ./simpleid2
e_uid=1001, e_gid=0
real_uid=1001, real_gid=1001
[guest@zkdychenko ~]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest),10(wheel) контекст=unconfined
_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@zkdychenko ~]$
```

Figure 3.8: Установка SetGID-бита и выполнение программы simpleid2

## 3.9 Шаг 9

Создала программу readfile.c (рис. 3.9).



```
#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;
}
```

Figure 3.9: Программа readfile.c

## 3.10 Шаг 10

Откомпилировала программу. Сменила владельца у файла readfile.c и изменила права так, чтобы только суперпользователь (root) мог прочитать его, а guest не мог. Проверила, что пользователь guest не может прочитать файл readfile.c (рис. 3.10).

```
[guest@zkdyachenko ~]$ gcc readfile.c -o readfile
[guest@zkdyachenko ~]$ sudo chown guest2:guest /home/guest/readfile.c
[sudo] пароль для guest:
[guest@zkdyachenko ~]$ sudo chmod ug-r /home/guest/readfile.c
[guest@zkdyachenko ~]$ ls -l readfile.c
--w--w-r--. 1 guest2 guest 407 ноя 11 18:41 readfile.c
[guest@zkdyachenko ~]$ cat readfile.c
cat: readfile.c: Отказано в доступе
[guest@zkdyachenko ~]$
```

Figure 3.10: Новый владелец readfile.c

## 3.11 Шаг 11

Сменила у программы readfile владельца и установила SetUID-бит (рис. 3.11).

```
[guest@zkdyachenko ~]$ sudo chown guest2:guest /home/guest/readfile
[sudo] пароль для guest:
[guest@zkdyachenko ~]$ sudo chmod u+s /home/guest/readfile
[guest@zkdyachenko ~]$ █
```

Figure 3.11: Смена владельца и установка SetUID-бита

## 3.12 Шаг 12

Проверила, что программа readfile не может прочитать файл readfile.c (рис. 3.12).

```
[guest@zkdyachenko ~]$ ./readfile readfile.c
700000x86_64./readfilereadfile.cXDG_VTNR=1XDG_SESSION_ID=1HOSTNAME=zkdyachenko.l
ocaldomainSHELL=/bin/bashTERM=xterm-256colorHISTSIZE=1000USER=guestLS_COLORS=rs=
0:di=38;5;27:ln=38;5;51:mh=44;38;5;15:pi=40;38;5;11:so=38;5;13:do=38;5;5:bd=48;5
;232;38;5;11:cd=48;5;232;38;5;3:or=48;5;232;38;5;9:mi=05;48;5;232;38;5;15:su=48;
5;196;38;5;15:sg=48;5;11;38;5;16:ca=48;5;196;38;5;226:tw=48;5;10;38;5;16:ow=48;5
;10;38;5;21:st=48;5;21;38;5;15:ex=38;5;34:*.tar=38;5;9:*.tgz=38;5;9:*.arc=38;5;9
:*.arj=38;5;9:*.taz=38;5;9:*.lha=38;5;9:*.lz4=38;5;9:*.lzh=38;5;9:*.lzma=38;5;9:
*.tlz=38;5;9:*.txz=38;5;9:*.tzo=38;5;9:*.t7z=38;5;9:*.zip=38;5;9:*.z=38;5;9:*.Z=
38;5;9:*.dz=38;5;9:*.gz=38;5;9:*.lrz=38;5;9:*.lzo=38;5;9:*.lzo=38;5;9:*.xz=38;5;9
:*.bz2=38;5;9:*.bz=38;5;9:*.tbz=38;5;9:*.tbz2=38;5;9:*.tz=38;5;9:*.deb=38;5;9:*.
rpm=38;5;9:*.jar=38;5;9:*.war=38;5;9:*.ear=38;5;9:*.sar=38;5;9:*.rar=38;5;9:*.al
z=38;5;9:*.ace=38;5;9:*.zoo=38;5;9:*.cpio=38;5;9:*.7z=38;5;9:*.rz=38;5;9:*.cab=3
8;5;9:*.jpg=38;5;13:*.jpeg=38;5;13:*.gif=38;5;13:*.bmp=38;5;13:*.pbm=38;5;13:*.p
gm=38;5;13:*.ppm=38;5;13:*.tga=38;5;13:*.xbm=38;5;13:*.xpm=38;5;13:*.tif=38;5;13
:*.tiff=38;5;13:*.png=38;5;13:*.svg=38;5;13:*.svgz=38;5;13:*.mng=38;5;13:*.pcx=3
8;5;13:*.mov=38;5;13:*.mpg=38;5;13:*.mpeg=38;5;13:*.m2v=38;5;13:*.mkv=38;5;13:*.
webm=38;5;13:*.ogm=38;5;13:*.mp4=38;5;13:*.m4v=38;5;13:*.mp4v=38;5;13:*.vob=38;5
;13:*.qt=38;5;13:*.nuv=38;5;13:*.wmv=38;5;13:*.asf=38;5;13:*.rm=38;5;13:*.rmvb=3
80ошибка сегментирования
```

Figure 3.12: Попытка прочитать readfile.c

## 3.13 Шаг 13

Проверила, что программа readfile не может прочитать файл /etc/shadow, потому что владелец файла программы - guest2 (рис. 3.13).

[illegible]

Figure 3.13: Попытка прочитать /etc/shadow

### 3.14 Шаг 14

Выяснила, что установлен атрибут Sticky на директории /tmp - в конце списка атрибутов видим t (рис. 3.14). От имени пользователя guest создала файл file01.txt в директории /tmp со словом test. Просмотрела атрибуты у только что созданного файла и разрешила чтение и запись для категории пользователей «все остальные».

```
[guest@zkdyachenko ~]$ ls -l / | grep tmp
drwxrwxrwt. 16 root root 4096 ноя 11 19:01 tmp
[guest@zkdyachenko ~]$ echo "test" > /tmp/file01.txt
[guest@zkdyachenko ~]$ ls -l /tmp/file01.txt
-rw-rw-r--. 1 guest guest 5 ноя 11 19:07 /tmp/file01.txt
[guest@zkdyachenko ~]$ chmod o+rw /tmp/file01.txt
[guest@zkdyachenko ~]$ ls -l /tmp/file01.txt
-rw-rw-rw-. 1 guest guest 5 ноя 11 19:07 /tmp/file01.txt
[guest@zkdyachenko ~]$
```

Figure 3.14: Создание файла и установка атрибутов

### 3.15 Шаг 15

От пользователя guest2 прочитала файл /tmp/file01.txt. Кроме того, удалось дозаписать в файл слово test2, записать слово test3, стерев при этом всю имеющуюся в файле информацию (рис. 3.15).

```
[guest2@zkdyachenko ~]$ cat /tmp/file01.txt
test
[guest2@zkdyachenko ~]$ echo "test2" >> /tmp/file01.txt
[guest2@zkdyachenko ~]$ cat /tmp/file01.txt
test
test2
[guest2@zkdyachenko ~]$ echo "test3" > /tmp/file01.txt
[guest2@zkdyachenko ~]$ cat /tmp/file01.txt
test3
[guest2@zkdyachenko ~]$ █
```

---

Figure 3.15: Изменение файла /tmp/file01.txt от имени пользователя guest2

### 3.16 Шаг 16

От пользователя guest2 попробовала удалить файл /tmp/file01.txt, но сделать это не удалось. Повысила свои права до суперпользователя и выполнила после этого команду, снимающую атрибут t (Sticky-бит) с директории /tmp. Затем покинула режим суперпользователя (рис. 3.16)

```
[guest2@zkdyachenko ~]$ rm /tmp/file01.txt
rm: невозможно удалить «/tmp/file01.txt»: Операция не позволена
[guest2@zkdyachenko ~]$ su -
Пароль:
Последний вход в систему: Чт ноя 11 17:52:44 MSK 2021 на pts/0
[root@zkdyachenko ~]# chmod -t /tmp
[root@zkdyachenko ~]# exit
logout
[guest2@zkdyachenko ~]$
```

---

Figure 3.16: Снятие Sticky-бита

### 3.17 Шаг 17

От пользователя guest2 проверила, что атрибута t у директории /tmp нет. От пользователя guest2 прочитала файл /tmp/file01.txt, дозаписала в файл слово test2,

записала слово test3, стерев при этом всю имеющуюся в файле информацию. Кроме того, удалось удалить файл от имени guest2, хотя он не был владельцем файла (рис. 3.17).

```
[guest2@zkdyachenko ~]$ ls -l / | grep tmp
drwxrwxrwx. 16 root root 4096 ноя 11 19:15 tmp
[guest2@zkdyachenko ~]$ cat /tmp/file01.txt
test3
[guest2@zkdyachenko ~]$ echo "test2" >> /tmp/file01.txt
[guest2@zkdyachenko ~]$ cat /tmp/file01.txt
test3
test2
[guest2@zkdyachenko ~]$ echo "test3" > /tmp/file01.txt
[guest2@zkdyachenko ~]$ cat /tmp/file01.txt
test3
[guest2@zkdyachenko ~]$ rm /tmp/file01.txt
[guest2@zkdyachenko ~]$ █
```

---

Figure 3.17: Удаление файла от имени guest2

## 3.18 Шаг 18

Повысила свои права до суперпользователя и вернула атрибут t на директорию /tmp (рис. 3.18).

```
[guest2@zkdyachenko ~]$ su -
Пароль:
Последний вход в систему: 4т ноя 11 19:15:02 MSK 2021 на pts/0
[root@zkdyachenko ~]# chmod +t /tmp
[root@zkdyachenko ~]# exit
logout
[quest2@zkdyachenko ~]$ █
```

Figure 3.18: Возвращение атрибута t

## 4 Выводы

В результате работы я изучила механизмы изменения идентификаторов, применение SetUID- и Sticky-битов. Были получены практические навыки работы в консоли с дополнительными атрибутами и рассмотрена работа механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов. Результаты работы находятся в репозитории на GitHub, а также есть скринкаст выполнения лабораторной работы.