

# **Отчёт по лабораторной работе №05**

**Дисциплина: Основы информационной безопасности**

**Бансимба Клодели Дьегра НПИбд-02-22**

# Содержание

<b>1</b>	<b>5.1 Цель работы</b>	<b>4</b>
<b>2</b>	<b>5.2 Порядок выполнения работы</b>	<b>5</b>
2.1	5.2.1 Создание программы . . . . .	5
<b>3</b>	<b>Выводы</b>	<b>13</b>

## Список иллюстраций

2.1	программа simpleid . . . . .	5
2.2	программа simpleid . . . . .	6
2.3	программа simpleid2 . . . . .	6
2.4	запустили simpleid2 . . . . .	6
2.5	запустили simpleid2 и id . . . . .	7
2.6	программа readfile . . . . .	7
2.7	программа readfile . . . . .	8
2.8	результат программы readfile . . . . .	8
2.9	исследование Sticky-бита . . . . .	9
2.10	исследование Sticky-бита . . . . .	11
2.11	исследование Sticky-бита . . . . .	12

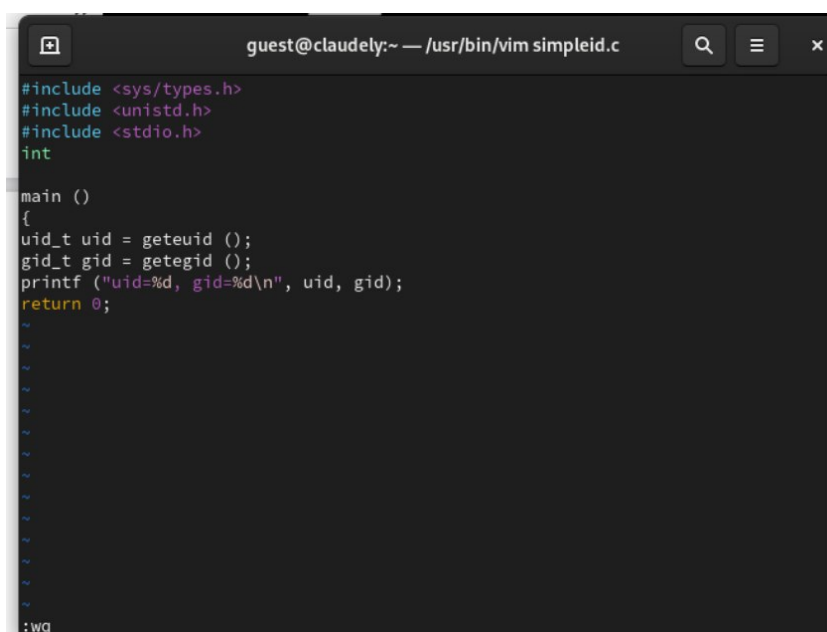
## **1 5.1 Цель работы**

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 5.2 Порядок выполнения работы

### 2.1 5.2.1 Создание программы

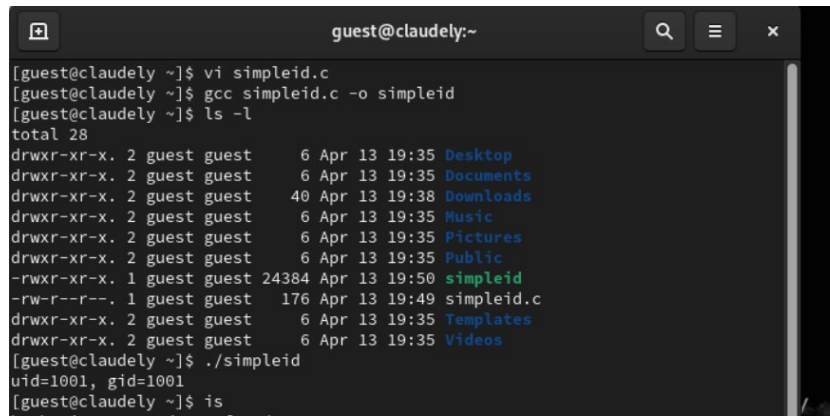
1. Вошли в систему от имени пользователя guest.



```
guest@claudely:~ — /usr/bin/vim 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;
}
:wq
```

Рис. 2.1: программа simpleid

2. Написали программу simpleid.c. Скомпилировали программу и убедились, что файл программы создан: `gcc simpleid.c -o simpleid`. Выполнили программу simpleid командой `./simpleid`. Выполнили системную программу id с помощью команды `id`. uid и gid совпадает в обеих программах



```
guest@claudely:~  
[guest@claudely ~]$ vi simpleid.c  
[guest@claudely ~]$ gcc simpleid.c -o simpleid  
[guest@claudely ~]$ ls -l  
total 28  
drwxr-xr-x. 2 guest guest 6 Apr 13 19:35 Desktop  
drwxr-xr-x. 2 guest guest 6 Apr 13 19:35 Documents  
drwxr-xr-x. 2 guest guest 40 Apr 13 19:38 Downloads  
drwxr-xr-x. 2 guest guest 6 Apr 13 19:35 Music  
drwxr-xr-x. 2 guest guest 6 Apr 13 19:35 Pictures  
drwxr-xr-x. 2 guest guest 6 Apr 13 19:35 Public  
-rw-r--r--. 1 guest guest 24384 Apr 13 19:50 simpleid  
-rw-r--r--. 1 guest guest 176 Apr 13 19:49 simpleid.c  
drwxr-xr-x. 2 guest guest 6 Apr 13 19:35 Templates  
drwxr-xr-x. 2 guest guest 6 Apr 13 19:35 Videos  
[guest@claudely ~]$ ./simpleid  
uid=1001, gid=1001  
[guest@claudely ~]$ is
```

Рис. 2.2: программа simpleid

3. Усложнили программу, добавив вывод действительных идентификаторов.



```
guest@claudely:~ — /usr/bin/vim simpleid2.c  
#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. 2.3: программа simpleid2

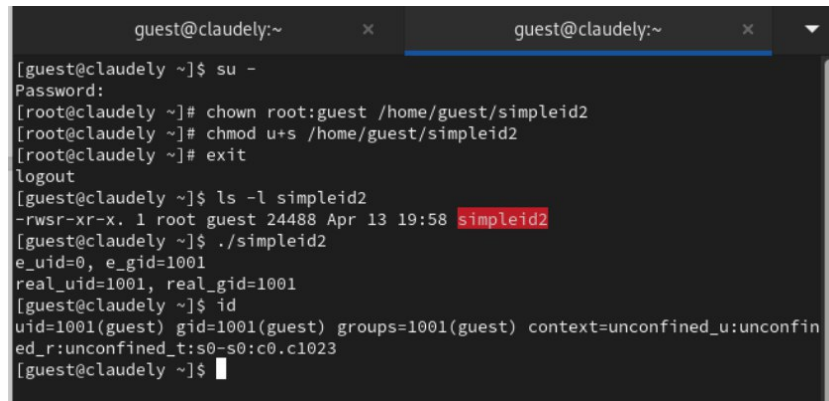
4. Скомпилировали и запустили simpleid2.c:



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

Рис. 2.4: запустили simpleid2

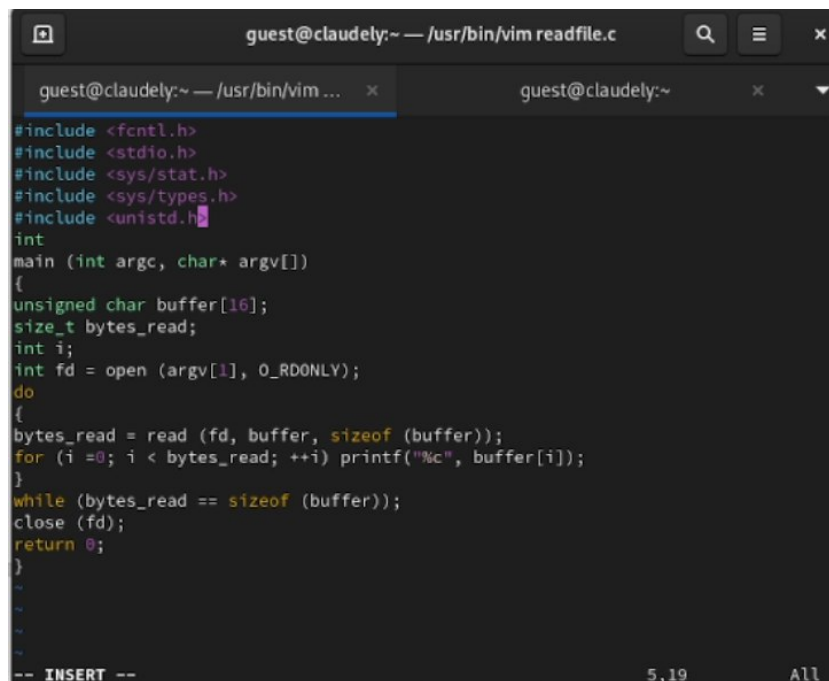
5. От имени суперпользователя выполнили команды, Использовали su для повышения прав до суперпользователя.Выполнили проверку правильности установки новых атрибутов и смены владельца файла simpleid2: Запустили simpleid2 и id:



```
guest@claudely:~  
[guest@claudely ~]$ su -  
Password:  
[root@claudely ~]# chown root:guest /home/guest/simpleid2  
[root@claudely ~]# chmod u+s /home/guest/simpleid2  
[root@claudely ~]# exit  
logout  
[guest@claudely ~]$ ls -l simpleid2  
-rwsr-xr-x. 1 root guest 24488 Apr 13 19:58 simpleid2  
[guest@claudely ~]$ ./simpleid2  
e_uid=0, e_gid=1001  
real_uid=1001, real_gid=1001  
[guest@claudely ~]$ id  
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023  
[guest@claudely ~]$
```

Рис. 2.5: запустили simpleid2 и id

6. Проделали тоже самое относительно SetGID-бита.Написали программу readfile.c



```
guest@claudely:~ — /usr/bin/vim 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. 2.6: программа readfile

7. Откомпилировали её.

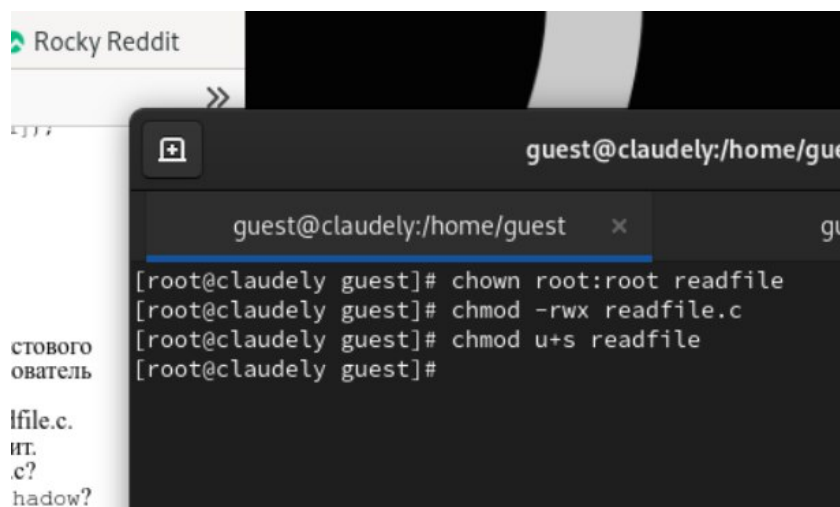
```
gcc readfile.c -o readfile
```



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

Рис. 2.7: программа readfile

8. Сменили владельца у файла readfile.c и изменили права так, чтобы только суперпользователь (root) мог прочитать его, а guest не мог. Проверили, что пользователь guest не может прочитать файл readfile.c. Сменили у программы readfile владельца и установили SetU'D-бит. Проверили, может ли программа readfile прочитать файл readfile.c. Проверили, может ли программа readfile прочитать файл /etc/shadow



```
Rocky Reddit
>>
guest@claudely:/home/guest
guest@claudely:/home/guest x
[root@claudely guest]# chown root:root readfile
[root@claudely guest]# chmod -rwx readfile.c
[root@claudely guest]# chmod u+s readfile
[root@claudely guest]#
```

Рис. 2.8: результат программы readfile

##Исследование Sticky-бита 1. Выяснили, установлен ли атрибут Sticky на директории /tmp:

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



2. От имени пользователя guest создали файл file01.txt в директории /tmp со словом test:

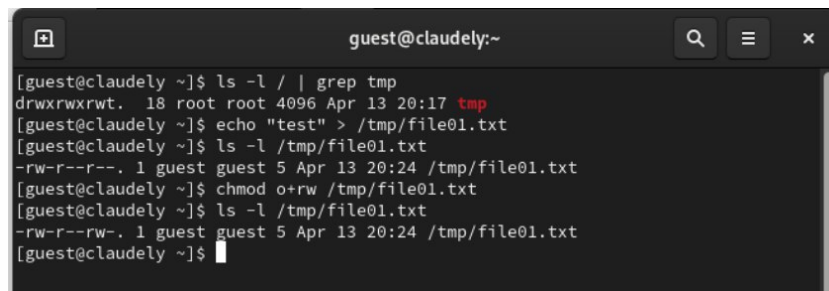
```
echo "test" > /tmp/file01.txt
```

3. Просмотрели атрибуты у только что созданного файла и разрешили чтение и запись для категории пользователей «все остальные»:

```
ls -l /tmp/file01.txt
```

```
chmod o+rw /tmp/file01.txt
```

```
ls -l /tmp/file01.txt
```



```
guest@claudely:~  
[guest@claudely ~]$ ls -l / | grep tmp  
drwxrwxrwt. 18 root root 4096 Apr 13 20:17 tmp  
[guest@claudely ~]$ echo "test" > /tmp/file01.txt  
[guest@claudely ~]$ ls -l /tmp/file01.txt  
-rw-r--r--. 1 guest guest 5 Apr 13 20:24 /tmp/file01.txt  
[guest@claudely ~]$ chmod o+rw /tmp/file01.txt  
[guest@claudely ~]$ ls -l /tmp/file01.txt  
-rw-r--rw-. 1 guest guest 5 Apr 13 20:24 /tmp/file01.txt  
[guest@claudely ~]$
```

Рис. 2.9: исследование Sticky-бита

4. От пользователя (не являющегося владельцем) попробовали прочитать файл /file01.txt:

```
cat /file01.txt
```

5. От пользователя попробовали дозаписать в файл /file01.txt слово test3 командой:

```
echo "test2" >> /file01.txt
```

6. Проверили содержимое файла командой:

```
cat /file01.txt
```

В файле теперь записано:

Test

Test2

7. От пользователя попробовали записать в файл /tmp/file01.txt слово test4, стерев при этом всю имеющуюся в файле информацию командой. Для этого воспользовалась командой `echo "test3" > /tmp/file01.txt`
8. Проверили содержимое файла командой

```
cat /tmp/file01.txt
```

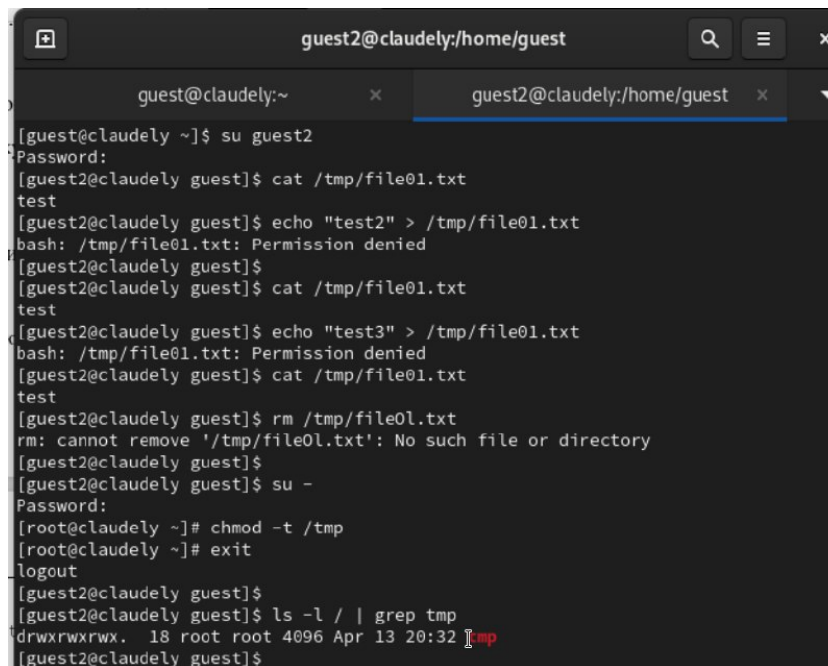
9. От пользователя попробовали удалить файл /tmp/file01.txt командой `rm /tmp/file01.txt`, однако получила отказ.
10. От суперпользователя командой выполнили команду, снимающую атрибут t (Sticky-бит) с директории /tmp:

```
chmod -t /tmp
```

Покинули режим суперпользователя командой `exit`.

11. От пользователя проверили, что атрибута t у директории /tmp нет:

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



```
guest2@claudely:/home/guest
[guest@claudely ~]$ su guest2
Password:
[guest2@claudely guest]$ cat /tmp/file01.txt
test
[guest2@claudely guest]$ echo "test2" > /tmp/file01.txt
bash: /tmp/file01.txt: Permission denied
[guest2@claudely guest]$ cat /tmp/file01.txt
test
[guest2@claudely guest]$ echo "test3" > /tmp/file01.txt
bash: /tmp/file01.txt: Permission denied
[guest2@claudely guest]$ cat /tmp/file01.txt
test
[guest2@claudely guest]$ rm /tmp/file01.txt
rm: cannot remove '/tmp/file01.txt': No such file or directory
[guest2@claudely guest]$ su -
Password:
[root@claudely ~]# chmod -t /tmp
[root@claudely ~]# exit
logout
[guest2@claudely guest]$
[guest2@claudely guest]$ ls -l / | grep tmp
drwxrwxrwx. 18 root root 4096 Apr 13 20:32 tmp
[guest2@claudely guest]$
```

Рис. 2.10: исследование Sticky-бита

12. Повторили предыдущие шаги. Получилось удалить файл
13. Удалось удалить файл от имени пользователя, не являющегося его владельцем.
14. Повысили свои права до суперпользователя и вернули атрибут t на директорию /tmp :

su

chmod +t /tmp

exit

```
guest2@claudely:/home/guest
drwxrwxrwx. 18 root root 4096 Apr 13 20:32 tmp
[guest2@claudely guest]$
[guest2@claudely guest]$
[guest2@claudely guest]$
[guest2@claudely guest]$
[guest2@claudely guest]$ cat /tmp/file01.txt
test
[guest2@claudely guest]$ echo "test2" > /tmp/file01.txt
bash: /tmp/file01.txt: Permission denied
[guest2@claudely guest]$ cat /tmp/file01.txt
test
[guest2@claudely guest]$ echo "test3" > /tmp/file01.txt
bash: /tmp/file01.txt: Permission denied
[guest2@claudely guest]$
[guest2@claudely guest]$ su -
Password:
[root@claudely ~]# chmod +t /tmp
[root@claudely ~]# exit
logout
[guest2@claudely guest]$ ls -l /tmp
total 4
-rw-r--rw-. 1 guest      guest      5 Apr 13 20:24 file01.txt
drwx-----. 3 root      root        17 Apr 13 19:34 systemd-private-
f3a7b5bb49c543e6bbacb84483f56cb5-chrond.service-jFhop4
drwx-----. 3 root      root        17 Apr 13 19:35 systemd-private-
f3a7b5bb49c543e6bbacb84483f56cb5-colord.service-09AhnG
```

Рис. 2.11: исследование Sticky-бита

## 3 Выводы

Изучили механизмы изменения идентификаторов, применения SetUID- и Sticky-битов. Получили практические навыки работы в консоли с дополнительными атрибутами.