

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ
ФАКУЛЬТЕТ ФИЗИКО-МАТЕМАТИЧЕСКИХ И ЕСТЕСТВЕННЫХ НАУК
КАФЕДРА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИИ

ОТЧЕТ

По дисциплине: Информационная безопасность.

Лабораторная работа № 5.

Дискреционное разграничение прав в Linux. Исследование влияния
дополнительных атрибутов.

Нгуен Чау Ки Ань

1032185287

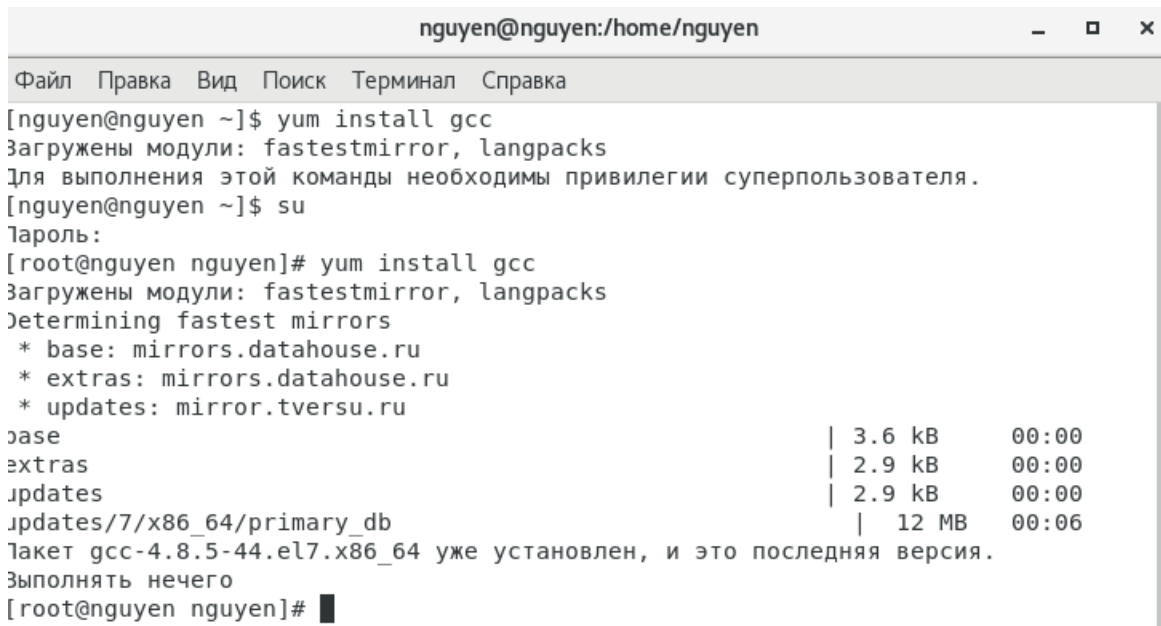
НБИбд-01-18

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. Подготовка к работе

1/ Средства разработки приложений



```
nguyen@nguyen:/home/nguyen
Файл Правка Вид Поиск Терминал Справка
[nguyen@nguyen ~]$ yum install gcc
Загружены модули: fastestmirror, langpacks
Для выполнения этой команды необходимы привилегии суперпользователя.
[nguyen@nguyen ~]$ su
Пароль:
[root@nguyen nguyen]# yum install gcc
Загружены модули: fastestmirror, langpacks
Determining fastest mirrors
 * base: mirrors.datahouse.ru
 * extras: mirrors.datahouse.ru
 * updates: mirror.tversu.ru
base | 3.6 kB 00:00
extras | 2.9 kB 00:00
updates | 2.9 kB 00:00
updates/7/x86_64/primary_db | 12 MB 00:06
Пакет gcc-4.8.5-44.el7.x86_64 уже установлен, и это последняя версия.
Выполнять нечего
[root@nguyen nguyen]#
```

Figure 1: yum install gcc

2/ Отключить систему запретов до очередной перезагрузки системы командой `setenforce 0`. После этого команда `getenforce` должна выводить `Permissive`. В этой работе система SELinux рассматриваться не будет.

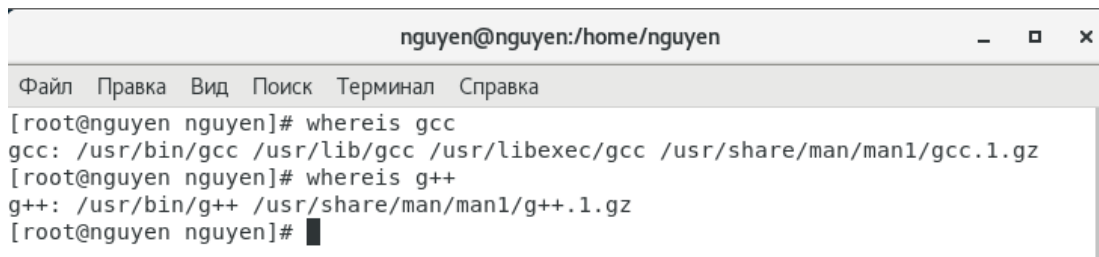
```
Выполнять нечего
[root@nguyen nguyen]# setenforce 0
[root@nguyen nguyen]# getenforce
Permissive
[root@nguyen nguyen]#
```

Figure 2: отключение SELinux

2. Компилирование программ

1. Для выполнения четвертой части задания вам потребуются навыки программирования, а именно, умение компилировать простые программы, написанные на языке C (C++), используя интерфейс CLI.

Компилятор языка C называется `gcc`. Компилятор языка C++ называется `g++` и запускается с параметрами почти так же, как `gcc`.



```
nguyen@nguyen:/home/nguyen
Файл Правка Вид Поиск Терминал Справка
[root@nguyen nguyen]# whereis gcc
gcc: /usr/bin/gcc /usr/lib/gcc /usr/libexec/gcc /usr/share/man/man1/gcc.1.gz
[root@nguyen nguyen]# whereis g++
g++: /usr/bin/g++ /usr/share/man/man1/g++.1.gz
[root@nguyen nguyen]#
```

Figure 3: проверка командами whereis gcc, whereis g++

3. Порядок выполнения работы

1. Создание программы

1. Войти в систему от имени пользователя guest.

2. Создать программу simpleid.c:



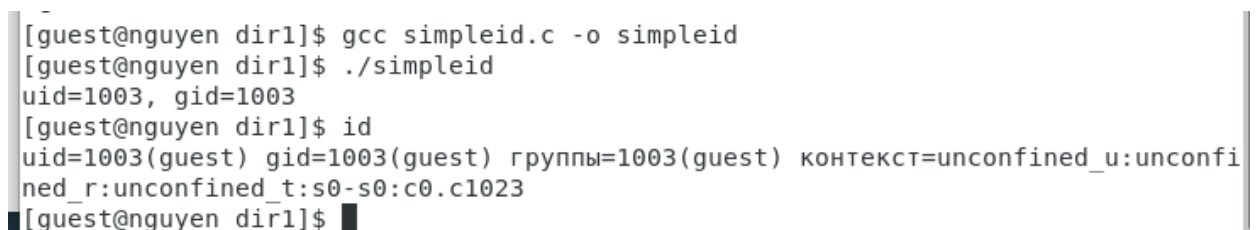
```
guest@nguyen:~/dir1
Файл Правка Вид Поиск Терминал Справка
[guest@nguyen dir1]$ touch simpleid.c
[guest@nguyen dir1]$ ls
simpleid.c
[guest@nguyen dir1]$ makefile simpleid.c
bash: makefile: команда не найдена...
[guest@nguyen dir1]$ cat > simpleid.c
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>
int
main ()
{
uid_t uid = geteuid ();
gid_t gid = getegid ();
printf ("uid=%d, gid=%d\n", uid, gid);
return 0;
}
^C
[guest@nguyen dir1]$
```

Figure 4: simpleid.c

3. Скомпилировать программу --- файл программы создан.

4. Выполнить программу simpleid.


5. Выполнить системную программу id.



```
[guest@nguyen dir1]$ gcc simpleid.c -o simpleid
[guest@nguyen dir1]$ ./simpleid
uid=1003, gid=1003
[guest@nguyen dir1]$ id
uid=1003(guest) gid=1003(guest) группы=1003(guest) контекст=unconfined_u:unconfi
ned_r:unconfined_t:s0-s0:c0.c1023
[guest@nguyen dir1]$
```

Figure 5: id

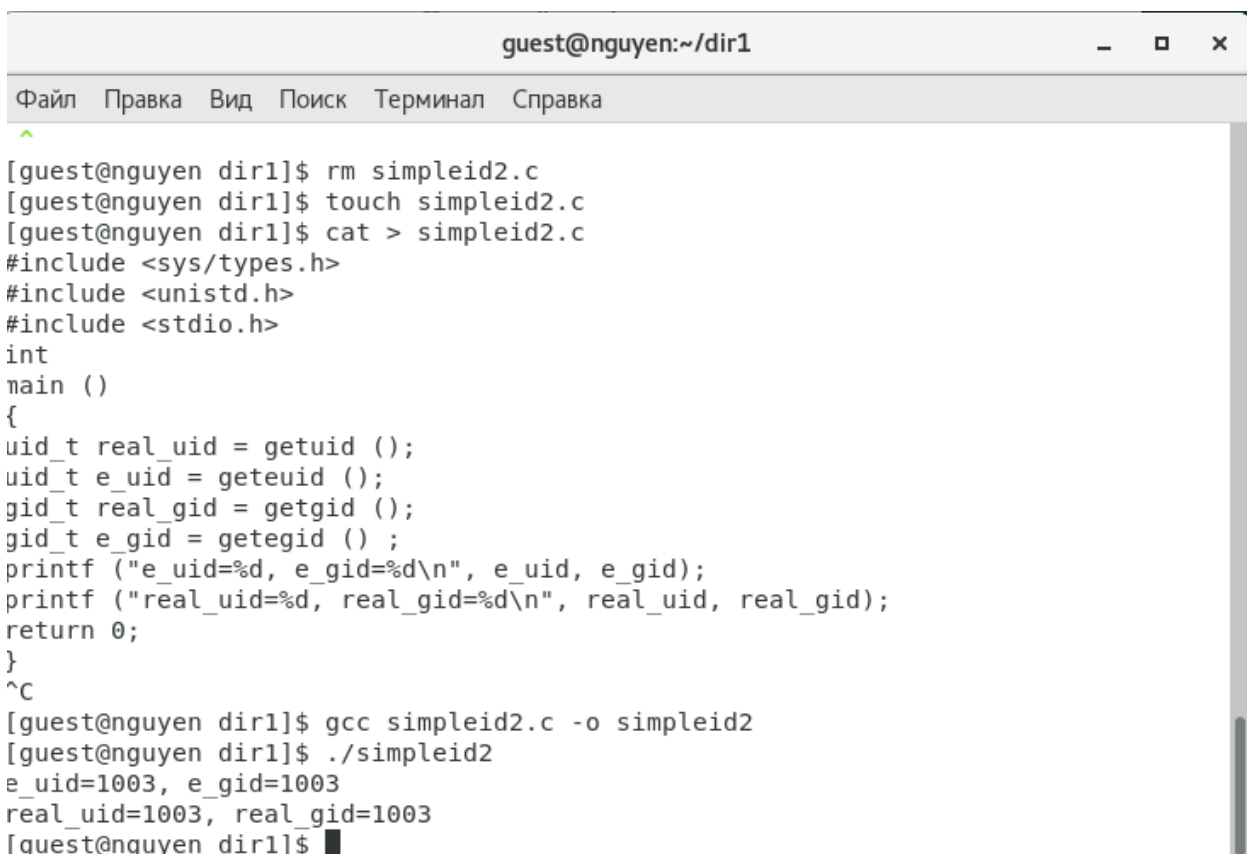
6. Усложнить программу, добавив вывод действительных идентификаторов.

A terminal window titled 'guest@nguyen:~/dir1' with a menu bar containing 'Файл', 'Правка', 'Вид', 'Поиск', 'Терминал', and 'Справка'. The terminal shows the following commands and output:

```
[guest@nguyen dir1]$ touch simpleid2.c
[guest@nguyen dir1]$ cat > 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;
}
^C
[guest@nguyen dir1]$
```

Figure 6: simpleid2.c

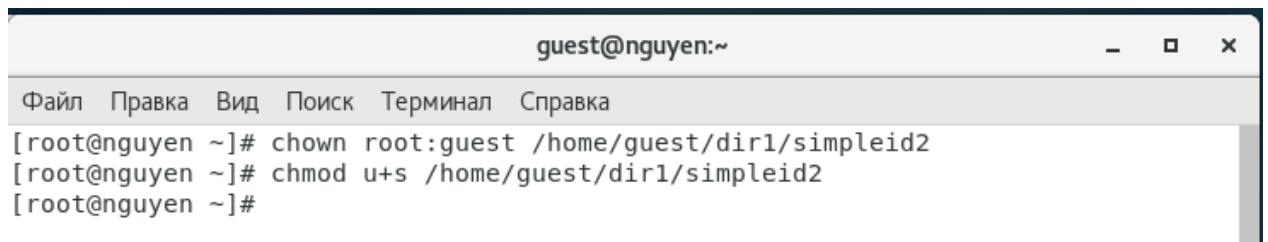
7. Скомпилировать и запустить simpleid2.c:

A terminal window titled 'guest@nguyen:~/dir1' with a menu bar containing 'Файл', 'Правка', 'Вид', 'Поиск', 'Терминал', and 'Справка'. The terminal shows the following commands and output:

```
[guest@nguyen dir1]$ rm simpleid2.c
[guest@nguyen dir1]$ touch simpleid2.c
[guest@nguyen dir1]$ cat > 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;
}
^C
[guest@nguyen dir1]$ gcc simpleid2.c -o simpleid2
[guest@nguyen dir1]$ ./simpleid2
e_uid=1003, e_gid=1003
real_uid=1003, real_gid=1003
[guest@nguyen dir1]$
```

Figure 7: программа ./simpleid2

8. Сменить владельца у файла readfile.c и изменили права так, чтобы только суперпользователь (root) мог прочитать его, а guest не мог.



```
guest@nguyen:~  
Файл  Правка  Вид  Поиск  Терминал  Справка  
[root@nguyen ~]# chown root:guest /home/guest/dir1/simpleid2  
[root@nguyen ~]# chmod u+s /home/guest/dir1/simpleid2  
[root@nguyen ~]#
```

Figure 8: chown root:guest /home/guest/dir1/readfile.c

9. Использовать sudo или повысьте временно свои права с помощью su.
10. Выполнить проверку правильности установки новых атрибутов и смены владельца файла simpleid2:

```
ls -l simpleid2
```

11. Запустить simpleid2 и id: ./simpleid2

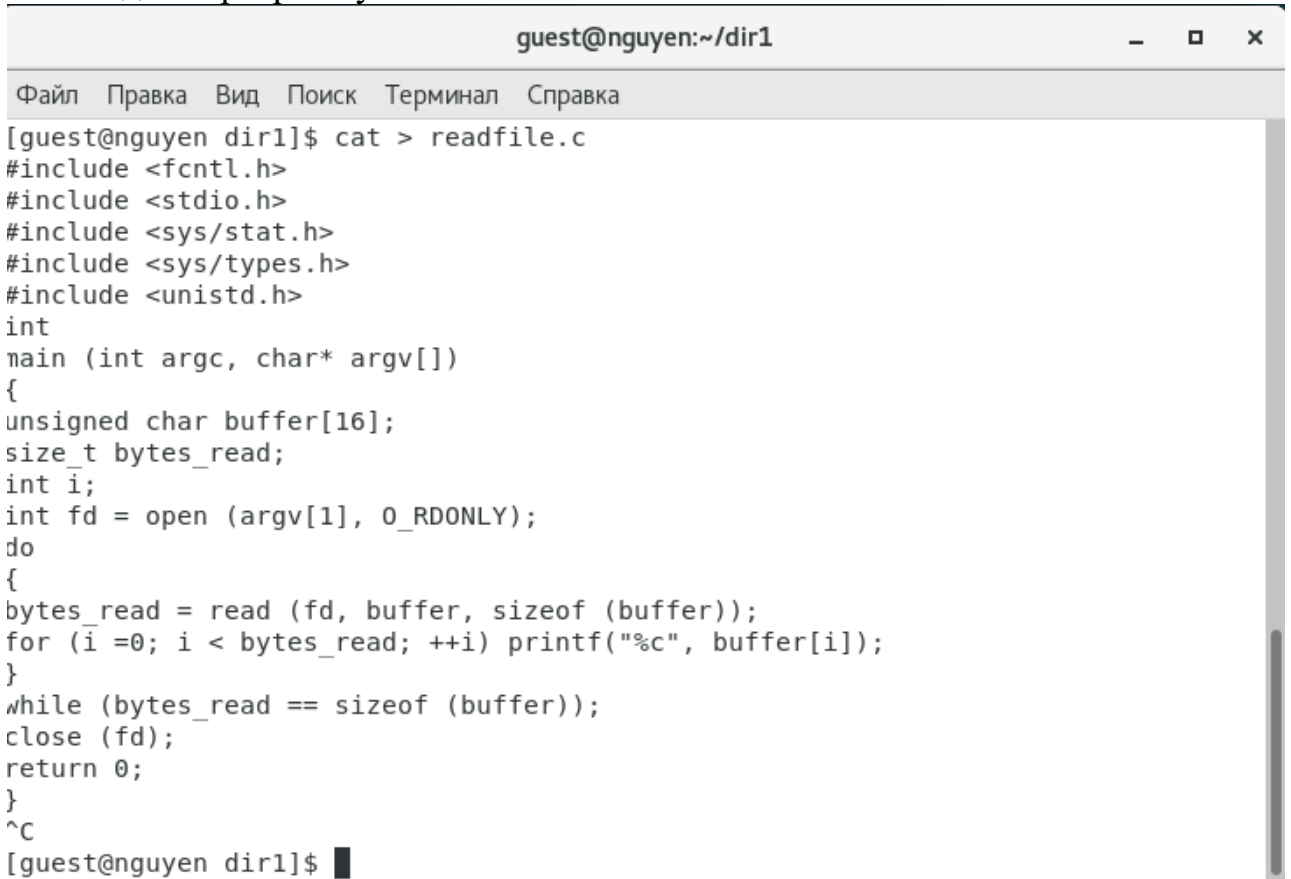
```
id
```

12. Прodelать тоже самое относительно SetGID-бита.

```
exit  
[guest@nguyen dir1]$ ls -l simpleid2  
-rwsrwxr-x. 1 root guest 8576 ноя 13 22:41 simpleid2  
[guest@nguyen dir1]$ ./simpleid2  
e_uid=0, e_gid=1003  
real_uid=1003, real_gid=1003  
[guest@nguyen dir1]$ id  
uid=1003(guest) gid=1003(guest) группы=1003(guest) контекст=unconfined_u:unconfi  
ned_r:unconfined_t:s0-s0:c0.c1023  
[guest@nguyen dir1]$
```

Figure 9: ./simpleid2

13. Создать программу readfile.c:



The screenshot shows a terminal window titled "guest@nguyen:~/dir1". The window has a menu bar with "Файл", "Правка", "Вид", "Поиск", "Терминал", and "Справка". The terminal content shows the command "cat > readfile.c" being executed, followed by the creation of a C program. The program includes headers for `<fcntl.h>`, `<stdio.h>`, `<sys/stat.h>`, `<sys/types.h>`, and `<unistd.h>`. It defines an `int` and a `main` function that takes `argc` and `argv`. Inside `main`, it declares `unsigned char buffer[16]`, `size_t bytes_read`, and `int i`. It opens a file at `argv[1]` with `O_RDONLY` flags. A `do` loop reads data from the file into the buffer and prints each character. The loop continues until `bytes_read` is less than the size of the buffer. Finally, it closes the file and returns 0. The prompt shows the user pressing `^C` to exit the editor.

```
[guest@nguyen dir1]$ cat > 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;
}
^C
[guest@nguyen dir1]$
```

Figure 10: readfile.c

14. Откомпилировать её. `gcc readfile.c -o readfile`

15. Сменить владельца у файла `readfile.c` и изменить права так, чтобы только суперпользователь (`root`) мог прочитать его, а `guest` не мог. `chown root:guest /home/guest/readfile.c chmod 700 /home/guest/readfile.c`

16. Проверить, что пользователь `guest` не может прочитать файл `readfile.c`.

17. Сменить у программы `readfile` владельца и установить `SetU'D`-бит.

18. Проверить, может ли программа `readfile` прочитать файл `readfile.c`

19. Проверить, может ли программа `readfile` прочитать файл `/etc/shadow`

```
[guest@nguyen dir1]$ gcc readfile.c -o readfile
[guest@nguyen dir1]$ chown root:guest /home/guest/dir1/readfile.c
chown: изменение владельца «/home/guest/dir1/readfile.c»: Операция не позволена
[guest@nguyen dir1]$ su
Пароль:
[root@nguyen dir1]# chown root:guest /home/guest/dir1/readfile.c
[root@nguyen dir1]# chmod 700 /home/guest/readfile.c
chmod: невозможно получить доступ к «/home/guest/readfile.c»: Нет такого файла и
ли каталога
[root@nguyen dir1]# chmod 700 /home/guest/dir1/readfile.c
[root@nguyen dir1]# exit
exit
[guest@nguyen dir1]$ ls -l readfile
-rwxrwxr-x. 1 guest guest 8512 ноя 13 23:22 readfile
[guest@nguyen dir1]$ ./readfile
```

Figure 11: результат программы readfile

3. Исследование Sticky-бита

1. Выясните, установлен ли атрибут Sticky на директории /tmp, для чего выполните команду

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

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

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

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

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

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

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

4. От пользователя guest2 (не являющегося владельцем) попробуйте прочитать файл /tmp/file01.txt:

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

5. От пользователя guest2 попробуйте дозаписать в файл /tmp/file01.txt слово test2 командой

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

Удалось ли вам выполнить операцию?

6. Проверьте содержимое файла командой

`cat /tmp/file01.txt`

7. От пользователя `guest2` попробуйте записать в файл `/tmp/file01.txt` слово `test3`, стерев при этом всю имеющуюся в файле информацию командой `echo "test3" > /tmp/file01.txt`

Удалось ли вам выполнить операцию?

8. Проверьте содержимое файла командой

`cat /tmp/file01.txt`

9. От пользователя `guest2` попробуйте удалить файл `/tmp/file01.txt` командой `rm /tmp/file01.txt`

Удалось ли вам удалить файл?

10. Повысьте свои права до суперпользователя следующей командой

`su -`

и выполните после этого команду, снимающую атрибут `t` (Sticky-бит) с директории `/tmp`:

`chmod -t /tmp`

11. Покиньте режим суперпользователя командой

`Exit`


```
guest@nguyen:~/dir1
Файл  Правка  Вид  Поиск  Терминал  Справка
[guest@nguyen dir1]$ ls -l |grep tmp
[guest@nguyen dir1]$ echo "test" > /tmp/file01.txt
[guest@nguyen dir1]$ ls -l /tmp/file01.txt
-rw-rw-r--. 1 guest guest 5 ноя 13 23:34 /tmp/file01.txt
[guest@nguyen dir1]$ chmod o+rw /tmp/file01.txt
[guest@nguyen dir1]$ ls -l /tmp/file01.txt
-rw-rw-rw-. 1 guest guest 5 ноя 13 23:34 /tmp/file01.txt
[guest@nguyen dir1]$ cat /tmp/file01.txt
test
[guest@nguyen dir1]$ echo "test2" > /tmp/file01.txt
[guest@nguyen dir1]$ cat /tmp/file01.txt
test2
[guest@nguyen dir1]$ echo "test3" > /tmp/file01.txt
[guest@nguyen dir1]$ cat /tmp/file01.txt
test3
[guest@nguyen dir1]$ rm /tmp/file01.txt
rm: невозможно удалить «/tmp/file01.txt»: Нет такого файла или каталога
[guest@nguyen dir1]$ su -
Пароль:
Последний вход в систему:Сб ноя 13 23:23:56 MSK 2021на pts/0
[root@nguyen ~]# chmod -t /tmp
[root@nguyen ~]# exit
logout
[guest@nguyen dir1]$
```

Figure 13: исследование

12. От пользователя guest2 проверьте, что атрибута t у директории /tmp нет:

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

13. Повторите предыдущие шаги. Какие наблюдаются изменения?

14. Удалось ли вам удалить файл от имени пользователя, не являющегося его владельцем? Ваши наблюдения занесите в отчёт.

Информационная безопасность компьютерных сетей 39

15. Повысьте свои права до суперпользователя и верните атрибут t на директорию /tmp:

```
su -
```

```
chmod +t /tmp
```

```
exit
```

```
[guest@nguyen dir1]$ su guest2
Пароль:
[guest2@nguyen dir1]$ ls -l / | grep tmp
drwxrwxrwx. 17 root root 4096 ноя 13 23:41 tmp
[guest2@nguyen dir1]$ cd /tmp
[guest2@nguyen tmp]$ echo "test3" > /tmp/file01.txt
[guest2@nguyen tmp]$ cat /tmp/file01.txt
test3
[guest2@nguyen tmp]$ rm file01.txt
[guest2@nguyen tmp]$ su
Пароль:
[root@nguyen tmp]# chmod +t/tmp
chmod: пропущен операнд после «+t/tmp»
По команде «chmod --help» можно получить дополнительную информацию.
[root@nguyen tmp]# chmod +t /tmp
[root@nguyen tmp]# exit
exit
[guest2@nguyen tmp]$ █
```

Figure 14: исследование Sticky-бита

4. Вывод

Изучили механизмы изменения идентификаторов, применения SetUID- и Stickyбитов. Получили практические навыки работы в консоли с дополнительными атрибутами. Также мы рассмотрели работу механизма смены идентификатора процессов пользователей и влияние бита Sticky на запись и удаление файлов.