

Отчет по лабораторной работе №5

**Дискреционное разграничение прав в Linux. Исследование влияния
дополнительных атрибутов**

Галацан Николай, НПИбд-01-22

Содержание

| | | |
|----------|---|-----------|
| 1 | Цель работы | 4 |
| 2 | Выполнение лабораторной работы [1] | 5 |
| 2.1 | Создание программы | 5 |
| 2.2 | Исследование Sticky-бита | 9 |
| 3 | Выводы | 12 |
| | Список литературы | 13 |

Список иллюстраций

| | | |
|------|--|----|
| 2.1 | Программа simpleid.c | 5 |
| 2.2 | Компиляция и выполнение. Сравнение с id | 6 |
| 2.3 | Программа simpleid2.c | 6 |
| 2.4 | Компиляция и выполнение | 6 |
| 2.5 | Запуск simpleid2 с SetUID. Сравнение результатов | 7 |
| 2.6 | Запуск simpleid2 с SetGID. Сравнение результатов | 7 |
| 2.7 | Программа readfile.c | 8 |
| 2.8 | Изменение прав доступа, проверка от имени пользователя guest . | 8 |
| 2.9 | Установка SetUID для readfile и проверка | 9 |
| 2.10 | Создание файла, изменение прав, просмотр и попытки записи . . | 10 |
| 2.11 | Попытка удаления | 10 |
| 2.12 | Повторение операций без атрибута t | 11 |

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. [1]

2 Выполнение лабораторной работы [1]

2.1 Создание программы

От имени пользователя guest создаю программу simpleid.c (рис. 2.1).

A screenshot of a terminal window with a dark background. The title bar at the top reads "guest@ngalacan:~ — mcedit simpleid.c". The terminal content shows the creation of a file named "simpleid.c". The first line is a comment: "simpleid.c [-M--] 0 L:[1+ 3 4/ 12] *(62 / 175b) 0010 0x00A [*][X]". This is followed by three include statements: "#include <sys/types.h>", "#include <unistd.h>", and "#include <stdio.h>". Then, the "main" function is defined, starting with "int main ()" and a curly brace. Inside the function, there are three lines: "uid_t uid = geteuid ();", "gid_t gid = getegid ();", and "printf ("uid=%d, gid=%d\n", uid, gid);". The function ends with "return 0;" and a closing curly brace. The terminal window has standard search and menu icons in the top right corner.

```
simpleid.c [-M--] 0 L:[ 1+ 3 4/ 12] *(62 / 175b) 0010 0x00A [*][X]
#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;
}
```

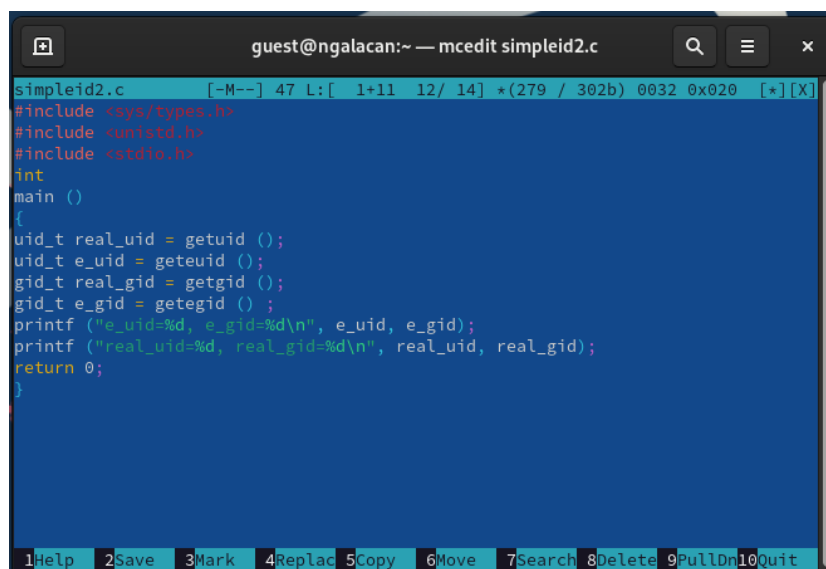
Рис. 2.1: Программа simpleid.c

Компилирую программу командой `gcc simpleid.c -o simpleid` и выполняю. Выполняю системную программу `id` и убеждаюсь, что обе программы выводят одинаковые данные (рис. 2.2).

```
[guest@ngalacan ~]$ gcc simpleid.c -o simpleid
[guest@ngalacan ~]$ ls -l
total 32
drwxrwxrwx. 2 guest users 19 Mar 18 11:19 bit
drwxr-xr-x. 2 guest users 6 Sep 13 2023 Documents
drwxr-xr-x. 2 guest users 6 Sep 13 2023 Pictures
-rwxr-xr-x. 1 guest users 25960 Mar 31 14:08 simpleid
-rw-r--r--. 1 guest users 175 Mar 31 14:07 simpleid.c
[guest@ngalacan ~]$ ./simpleid
uid=1004, gid=100
[guest@ngalacan ~]$ id
uid=1004(guest) gid=100(users) groups=100(users),1005(guest) context=unconfined_
u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@ngalacan ~]$
```

Рис. 2.2: Компиляция и выполнение. Сравнение с id

Создаю усложненную программу simpleid2.c (рис. 2.3).



```
simpleid2.c [-M--] 47 L: [ 1+11 12/ 14] *(279 / 302b) 0032 0x020 [*][X]
#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.c

Компилирую и запускаю (рис. 2.4).

```
[guest@ngalacan ~]$ mcedit simpleid2.c
[guest@ngalacan ~]$ gcc simpleid2.c -o simpleid2
[guest@ngalacan ~]$ ./simpleid2
e_uid=1004, e_gid=100
real_uid=1004, real_gid=100
[guest@ngalacan ~]$
```

Рис. 2.4: Компиляция и выполнение

От имени суперпользователя изменяю владельца файла и добавляю атрибут s. Это означает, что пользователь будет выполнять файл с разрешениями владель-

ца файла. Проверяю правильность и запускаю программу, вновь сравниваю с `id`. Исходя из этого, можно сказать, что теперь владельцем файла является пользователь с `id 0` (`root`), а изначально владельцем файла был пользователь с `id 1004` (`guest`) (рис. 2.5).

```
[guest@ngalacan ~]$ su -
Password:
[root@ngalacan ~]# chown root:guest /home/guest/simpleid2
[root@ngalacan ~]# chmod u+s /home/guest/simpleid2
[root@ngalacan ~]# exit
logout
[guest@ngalacan ~]$ ls -l simpleid2
-rwsr-xr-x. 1 root guest 26064 Mar 31 14:10 simpleid2
[guest@ngalacan ~]$ ./simpleid2
e_uid=0, e_gid=100
real_uid=1004, real_gid=100
[guest@ngalacan ~]$ id
uid=1004(guest) gid=100(users) groups=100(users),1005(guest) context=unconfined_
u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@ngalacan ~]$
```

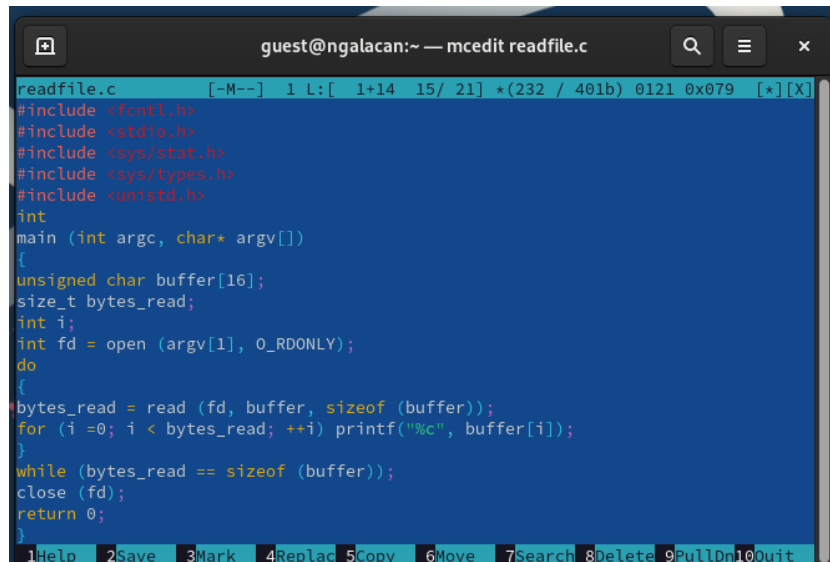
Рис. 2.5: Запуск `simpleid2` с `SetUID`. Сравнение результатов

То же самое проделываю для `SetGID` бита (рис. 2.6).

```
[guest@ngalacan ~]$ su -
Password:
[root@ngalacan ~]# chmod u-s /home/guest/simpleid2
[root@ngalacan ~]# chmod g+s /home/guest/simpleid2
[root@ngalacan ~]# exit
logout
[guest@ngalacan ~]$ ls -l
total 64
drwxrwxrwx. 2 guest users 19 Mar 18 11:19 dir
drwxr-xr-x. 2 guest users 6 Sep 13 2023 Documents
drwxr-xr-x. 2 guest users 6 Sep 13 2023 Pictures
-rwxr-xr-x. 1 guest users 25960 Mar 31 14:08 simpleid
-rwxr-sr-x. 1 root guest 26064 Mar 31 14:10 simpleid2
-rw-r--r--. 1 guest users 302 Mar 31 14:10 simpleid2.c
-rw-r--r--. 1 guest users 175 Mar 31 14:07 simpleid.c
[guest@ngalacan ~]$ ./simpleid2
e_uid=1004, e_gid=1005
real_uid=1004, real_gid=100
[guest@ngalacan ~]$
```

Рис. 2.6: Запуск `simpleid2` с `SetGID`. Сравнение результатов

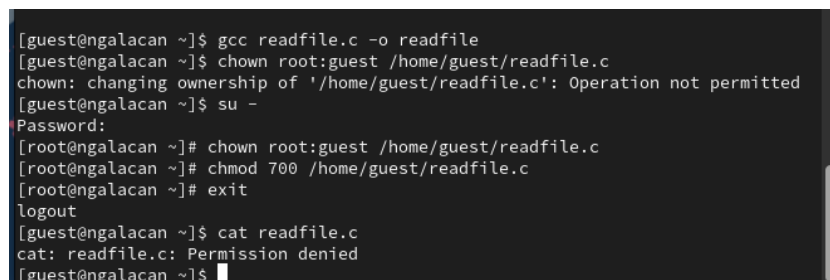
Создаю программу `readfile.c` (рис. 2.7).



```
readfile.c  [-M--]  1 L: [ 1+14 15/ 21] *(232 / 401b) 0121 0x079 [*][X]
#include <ctype.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. 2.7: Программа readfile.c

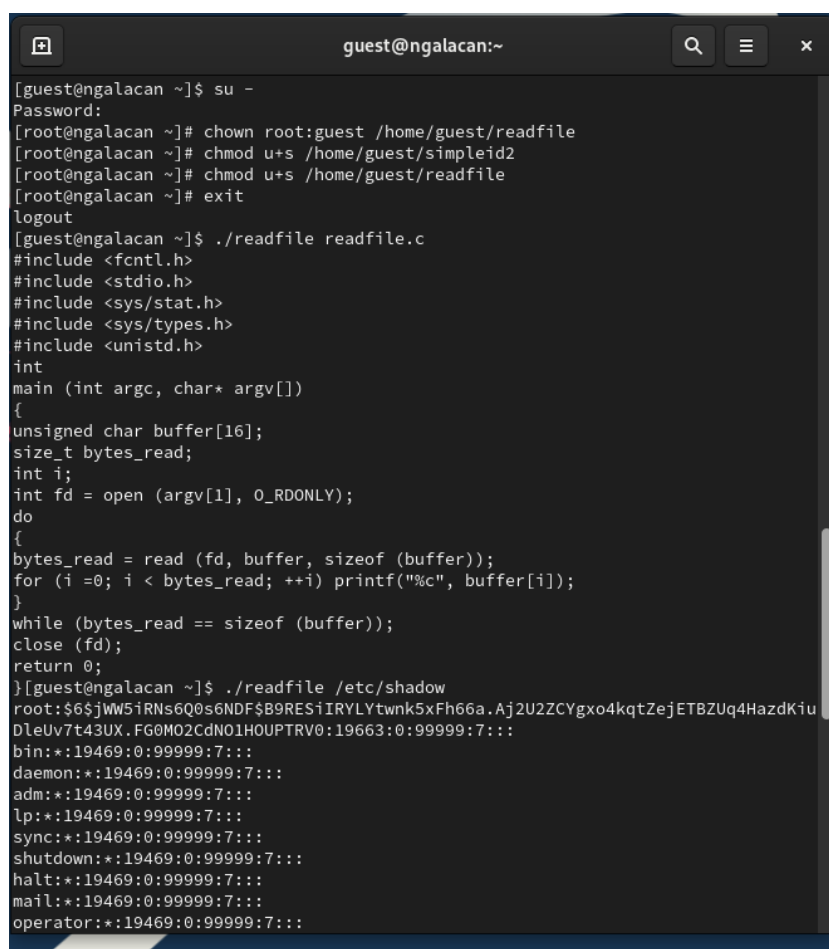
Компилирую ее, изменяю у файла readfile.c владельца, изменяю права доступа так, чтобы только суперпользователь мог прочитать его, а guest не мог. После проверяю, что пользователь guest не может прочитать файл (рис. 2.8).



```
[guest@ngalacan ~]$ gcc readfile.c -o readfile
[guest@ngalacan ~]$ chown root:guest /home/guest/readfile.c
chown: changing ownership of '/home/guest/readfile.c': Operation not permitted
[guest@ngalacan ~]$ su -
Password:
[root@ngalacan ~]# chown root:guest /home/guest/readfile.c
[root@ngalacan ~]# chmod 700 /home/guest/readfile.c
[root@ngalacan ~]# exit
logout
[guest@ngalacan ~]$ cat readfile.c
cat: readfile.c: Permission denied
[guest@ngalacan ~]$
```

Рис. 2.8: Изменение прав доступа, проверка от имени пользователя guest

Изменяю у программы readfile владельца, устанавливаю SetUID-бит. Проверяю, что программа может прочитать файлы readfile.c и /etc/shadow (рис. 2.9).

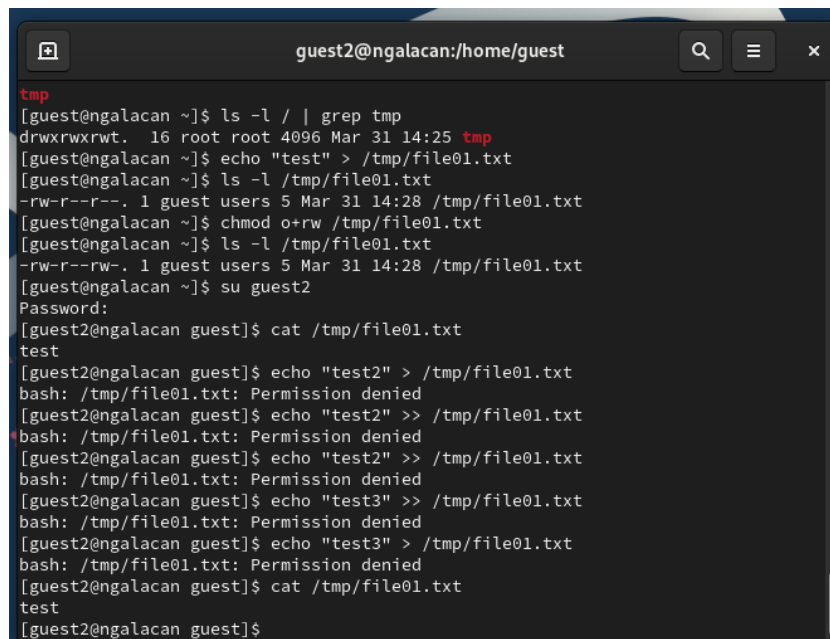


```
guest@ngalacan:~  
[guest@ngalacan ~]$ su -  
Password:  
[root@ngalacan ~]# chown root:guest /home/guest/readfile  
[root@ngalacan ~]# chmod u+s /home/guest/simpleid2  
[root@ngalacan ~]# chmod u+s /home/guest/readfile  
[root@ngalacan ~]# exit  
logout  
[guest@ngalacan ~]$ ./readfile readfile.c  
#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;  
}  
[guest@ngalacan ~]$ ./readfile /etc/shadow  
root:$6$jWw5iRNs6Q0s6NDF$B9RESiIRYLYtwnk5xFh66a.Aj2U2ZCYgx04kqtZeJtBZUq4HazzKiu  
DleUv7t43UX.FG0M02CdN01H0UPTRV0:19663:0:99999:7::  
bin:*:19469:0:99999:7::  
daemon:*:19469:0:99999:7::  
adm:*:19469:0:99999:7::  
lp:*:19469:0:99999:7::  
sync:*:19469:0:99999:7::  
shutdown:*:19469:0:99999:7::  
halt:*:19469:0:99999:7::  
mail:*:19469:0:99999:7::  
operator:*:19469:0:99999:7::
```

Рис. 2.9: Установка SetUID для readfile и проверка

2.2 Исследование Sticky-бита

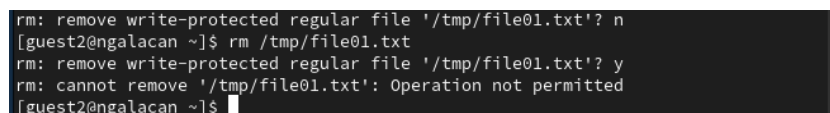
Проверяю, установлен ли атрибут Sticky на директории /tmp. От пользователя guest создаю файл со словом test командой `echo "test" > /tmp/file01.txt`. Просматриваю атрибуты у только что созданного файла и разрешаю чтение и запись для остальных пользователей. От пользователя guest2 пробую прочитать файл (успешно) и внести изменения в файл (отказ в доступе) (рис. 2.10).

A terminal window titled 'guest2@ngalacan:/home/guest' showing a series of commands and their outputs. The user 'tmp' runs 'ls -l / | grep tmp' showing file permissions for '/tmp'. Then, 'echo "test" > /tmp/file01.txt' creates the file. Subsequent 'ls -l /tmp/file01.txt' shows permissions '-rw-r--r--'. Then 'chmod o+rw /tmp/file01.txt' changes permissions to '-rw-r--rw-'. The user switches to 'guest2' and attempts to write to the file, but receives 'Permission denied' for multiple attempts. Finally, 'cat /tmp/file01.txt' shows the content 'test'.

```
guest2@ngalacan:/home/guest
tmp
[guest@ngalacan ~]$ ls -l / | grep tmp
drwxrwxrwt. 16 root root 4096 Mar 31 14:25 tmp
[guest@ngalacan ~]$ echo "test" > /tmp/file01.txt
[guest@ngalacan ~]$ ls -l /tmp/file01.txt
-rw-r--r--. 1 guest users 5 Mar 31 14:28 /tmp/file01.txt
[guest@ngalacan ~]$ chmod o+rw /tmp/file01.txt
[guest@ngalacan ~]$ ls -l /tmp/file01.txt
-rw-r--rw-. 1 guest users 5 Mar 31 14:28 /tmp/file01.txt
[guest@ngalacan ~]$ su guest2
Password:
[guest2@ngalacan guest]$ cat /tmp/file01.txt
test
[guest2@ngalacan guest]$ echo "test2" > /tmp/file01.txt
bash: /tmp/file01.txt: Permission denied
[guest2@ngalacan guest]$ echo "test2" >> /tmp/file01.txt
bash: /tmp/file01.txt: Permission denied
[guest2@ngalacan guest]$ echo "test2" >> /tmp/file01.txt
bash: /tmp/file01.txt: Permission denied
[guest2@ngalacan guest]$ echo "test3" >> /tmp/file01.txt
bash: /tmp/file01.txt: Permission denied
[guest2@ngalacan guest]$ echo "test3" > /tmp/file01.txt
bash: /tmp/file01.txt: Permission denied
[guest2@ngalacan guest]$ cat /tmp/file01.txt
test
[guest2@ngalacan guest]$
```

Рис. 2.10: Создание файла, изменение прав, просмотр и попытки записи

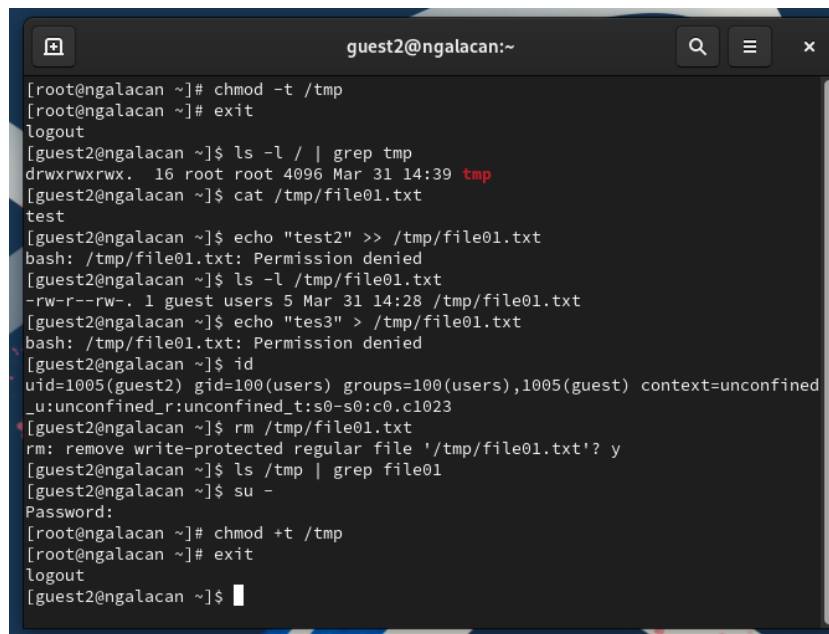
Пытаюсь удалить файл (отказ в доступе) (рис. 2.11).

A terminal window showing the user 'guest2' attempting to delete the file '/tmp/file01.txt' using the 'rm' command. The command fails with the error 'rm: cannot remove '/tmp/file01.txt': Operation not permitted' because the user lacks the necessary permissions.

```
rm: remove write-protected regular file '/tmp/file01.txt'? n
[guest2@ngalacan ~]$ rm /tmp/file01.txt
rm: remove write-protected regular file '/tmp/file01.txt'? y
rm: cannot remove '/tmp/file01.txt': Operation not permitted
[guest2@ngalacan ~]$
```

Рис. 2.11: Попытка удаления

От имени суперпользователя снимаю атрибут `t` с директории `/tmp` и от имени `guest2` проверяю. Повторяю предыдущие шаги: просмотр файла разрешен, запись в файл не разрешена, удаление файла разрешено. В конце возвращаю атрибут `t` на директорию `/tmp` от имени суперпользователя (рис. 2.12).



```
guest2@ngalacan:~  
[root@ngalacan ~]# chmod -t /tmp  
[root@ngalacan ~]# exit  
logout  
[guest2@ngalacan ~]$ ls -l / | grep tmp  
drwxrwxrwx. 16 root root 4096 Mar 31 14:39 tmp  
[guest2@ngalacan ~]$ cat /tmp/file01.txt  
test  
[guest2@ngalacan ~]$ echo "test2" >> /tmp/file01.txt  
bash: /tmp/file01.txt: Permission denied  
[guest2@ngalacan ~]$ ls -l /tmp/file01.txt  
-rw-r--rw-. 1 guest users 5 Mar 31 14:28 /tmp/file01.txt  
[guest2@ngalacan ~]$ echo "tes3" > /tmp/file01.txt  
bash: /tmp/file01.txt: Permission denied  
[guest2@ngalacan ~]$ id  
uid=1005(guest2) gid=100(users) groups=100(users),1005(guest) context=unconfined  
_u:unconfined_r:unconfined_t:s0-s0:c0.c1023  
[guest2@ngalacan ~]$ rm /tmp/file01.txt  
rm: remove write-protected regular file '/tmp/file01.txt'? y  
[guest2@ngalacan ~]$ ls /tmp | grep file01  
[guest2@ngalacan ~]$ su -  
Password:  
[root@ngalacan ~]# chmod +t /tmp  
[root@ngalacan ~]# exit  
logout  
[guest2@ngalacan ~]$
```

Рис. 2.12: Повторение операций без атрибута t

3 Выводы

Были изучены механизмы изменения идентификаторов, применения SetUID-и Sticky-битов. Получены практические навыки работы в консоли с дополнительными атрибутами. Рассмотрены работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

Список литературы

1. Кулябов Д.С., Королькова А.В., Геворкян М.Н. Информационная безопасность компьютерных сетей. Лабораторные работы, учебное пособие. Москва: РУДН, 2015. 64 с.