

# **Лабораторная работа № 5.**

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

Диана Алексеевна Садова

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Последовательность выполнения работы</b>	<b>6</b>
2.1	Создание программы . . . . .	6
2.2	Исследование Sticky-бита . . . . .	13
<b>3</b>	<b>Выводы</b>	<b>18</b>
	<b>Список литературы</b>	<b>19</b>

## Список иллюстраций

2.1	Заходим в систему пользователя guest . . . . .	6
2.2	Создаем файл с именем simpleid . . . . .	6
2.3	Код . . . . .	7
2.4	Скомпилируем программу . . . . .	7
2.5	Выполняем код . . . . .	7
2.6	Выполняем системную программу id . . . . .	8
2.7	Код . . . . .	8
2.8	Скомпилируем программу . . . . .	9
2.9	Выполняем команды . . . . .	9
2.10	Переходим в режим суперпользователя . . . . .	9
2.11	Проверяем правильности новых атрибутов . . . . .	10
2.12	Запускаем simpleid2 и id . . . . .	10
2.13	Код . . . . .	11
2.14	Скомпилируем программу . . . . .	11
2.15	Меняем прова доступа . . . . .	12
2.16	Проверяем, что guest не может прочитать файл . . . . .	12
2.17	Снимаем у readfile владельца . . . . .	12
2.18	Проверяем может ли readfile прочитать файл readfile.c . . . . .	12
2.19	Проверяем может ли readfile прочитать файл /etc/shadow . . . . .	13
2.20	Определяем установили ли атрибут Sticky . . . . .	13
2.21	Создаем файл file01.txt . . . . .	13
2.22	Просматриваем все атрибуты . . . . .	13
2.23	Пробуем читать файл от guest2 . . . . .	14
2.24	Пробуем что-то дописать в файл . . . . .	14
2.25	Проверяем содержимое файла . . . . .	14
2.26	Пробуем что-то дописать в файл . . . . .	15
2.27	Проверяем содержимое файла . . . . .	15
2.28	Пробуем удалить файл . . . . .	15
2.29	Переходим в режим суперпользователя и снимаем атрибут . . . . .	16
2.30	Выходим из режима суперпользователя . . . . .	16
2.31	Проверяем наличие атрибута t . . . . .	16
2.32	Повторяем предыдущие шаги . . . . .	17
2.33	Возвращаем предыдущие шаги . . . . .	17

## Список таблиц

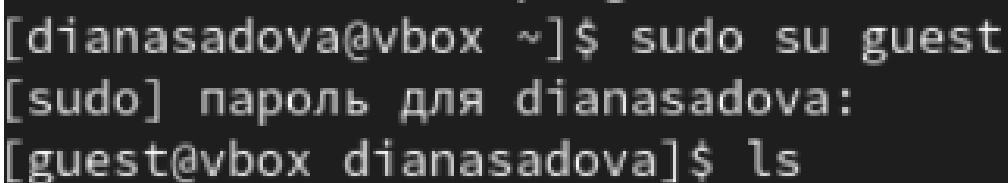
# 1 Цель работы

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 Последовательность выполнения работы

### 2.1 Создание программы

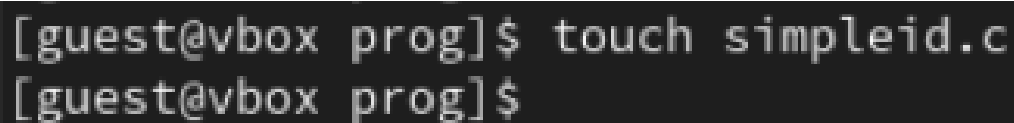
1. Войдите в систему от имени пользователя guest.(рис. 2.1).



```
[dianasadova@vbox ~]$ sudo su guest
[sudo] пароль для dianasadova:
[guest@vbox dianasadova]$ ls
```

Рис. 2.1: Заходим в систему пользователя guest

2. Создайте программу simpleid.c:(рис. 2.2),(рис. 2.3).



```
[guest@vbox prog]$ touch simpleid.c
[guest@vbox prog]$
```

Рис. 2.2: Создаем файл с именем simpleid

```

GNU nano 5.6.1 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;
}

```

Рис. 2.3: Код

3. Скомпилируйте программу и убедитесь, что файл программы создан:(рис. 2.4).

```

[guest@vbox prog]$ gcc simpleid.c -o simpleid
[guest@vbox prog]$ ls
simpleid simpleid.c
[guest@vbox prog]$

```

Рис. 2.4: Скомпилируем программу

4. Выполните программу simpleid:(рис. 2.5).

```

[guest@vbox prog]$ ./simpleid
uid=1001, gid=1001
[guest@vbox prog]$

```

Рис. 2.5: Выполняем код

5. Выполните системную программу id:(рис. 2.6).

```
[guest@vbox prog]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest)
ned_r:unconfined_t:s0-s0:c0.c1023
[guest@vbox prog]$
```

Рис. 2.6: Выполняем системную программу id

и сравните полученный вами результат с данными предыдущего пункта задания.

Записи идентичны.

6. Усложните программу, добавив вывод действительных идентификаторов:(рис. 2.7).

```
GNU nano 5.6.1 simpleid.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. 2.7: Код

Получившуюся программу назовите simpleid2.c.

7. Скомпилируйте и запустите simpleid2.c:(рис. 2.8).



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

Рис. 2.8: Скомпилируем программу

8. От имени суперпользователя выполните команды:(рис. 2.9).

```
[dianasadova@vbox ~]$ sudo -i
[sudo] пароль для dianasadova:
[root@vbox ~]# chown root:guest /home/guest/prog/simpleid2
[root@vbox ~]# chmod u+s /home/guest/prog/simpleid2
[root@vbox ~]#
```

Рис. 2.9: Выполняем команды

9. Используйте sudo или повысьте временно свои права с помощью su.(рис. 2.10).

```
[guest@vbox prog]$ sudo ./simpleid2
[sudo] пароль для guest:
guest is not in the sudoers file. This incident will be reported.
[guest@vbox prog]$ sudo id
[sudo] пароль для guest:
guest is not in the sudoers file. This incident will be reported.
```

Рис. 2.10: Переходим в режим суперпользователя

Поясните, что делают эти команды.

Мы пытаемся запустить код в файле simpleid2.c от имени суперпользователя.

10. Выполните проверку правильности установки новых атрибутов и смены владельца файла simpleid2:(рис. 2.11).

```
[guest@vbox prog]$ ls -l simpleid2
-rwsr-xr-x. 1 root guest 17656 map 14 12:25 simpleid2
[guest@vbox prog]$
```

Рис. 2.11: Проверяем правильности новых атрибутов

11. Запустите simpleid2 и id:(рис. 2.12).

```
[guest@vbox prog]$ ./simpleid2
id
e_uid=0, e_gid=1001
real_uid=1001, real_gid=1001
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfi
ned_r:unconfined_t:s0-s0:c0.c1023
[guest@vbox prog]$
```

Рис. 2.12: Запускаем simpleid2 и id

Сравните результаты.

simpleid2 - выдает только информацию о id (номер).

id - дает больше информации об пользователе и его группах.

12. Прodelайте тоже самое относительно SetGID-бита.

13. Создайте программу readfile.c:(рис. 2.13).

```

GNU nano 5.6.1                                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. 2.13: Код

14. Откомпилируйте её.(рис. 2.14).

```

[guest@vbox prog]$ gcc readfile.c -o readfile
[guest@vbox prog]$

```

Рис. 2.14: Скомпилируем программу

15. Смените владельца у файла readfile.c (или любого другого текстового файла в системе) и измените права так, чтобы только суперпользователь (root) мог прочитать его, а guest не мог.(рис. 2.15).

```
[guest@vbox prog]$ su dianasadova
Пароль:
[dianasadova@vbox prog]$ sudo chown root:guest /home/guest/simpleid2
[sudo] пароль для dianasadova:
chown: невозможно получить доступ к '/home/guest/simpleid2': Нет такого файла ил
и каталога
[dianasadova@vbox prog]$ sudo chown root:guest /home/guest/prog/simpleid2
[dianasadova@vbox prog]$ sudo chown root:guest /home/guest/prog/readfile
[dianasadova@vbox prog]$ sudo chmod u+s /home/guest/prog/readfile
[dianasadova@vbox prog]$
```

Рис. 2.15: Меняем прова доступа

16. Проверьте, что пользователь guest не может прочитать файл readfile.c.(рис. 2.16).

```
[dianasadova@vbox prog]$ sudo chmod ga-rwx /home/guest/prog/readfile
[dianasadova@vbox prog]$
```

Рис. 2.16: Проверяем, что guest не может прочитать файл

17. Смените у программы readfile владельца и установите SetU'D-бит.(рис. 2.17).

```
[root@vbox ~]# sudo chown root:guest /home/guest/prog/readfile
[root@vbox ~]#
```

Рис. 2.17: Снимаем у readfile владельца

18. Проверьте, может ли программа readfile прочитать файл readfile.c?(рис. 2.18).

```
[guest@vbox prog]$ ./readfile readfile.c
bash: ./readfile: Отказано в доступе
[guest@vbox prog]$
```

Рис. 2.18: Проверяем может ли readfile прочитать файл readfile.c

19. Проверьте, может ли программа readfile прочитать файл /etc/shadow?(рис. 2.19).

```
[guest@vbox prog]$ ./readfile /etc/shadow
bash: ./readfile: Отказано в доступе
[guest@vbox prog]$
```

Рис. 2.19: Проверяем может ли readfile прочитать файл /etc/shadow

## 2.2 Исследование Sticky-бита

1. Выясните, установлен ли атрибут Sticky на директории /tmp, для чего выполните команду(рис. 2.20).

```
[guest@vbox prog]$ ls -l / | grep tmp
drwxrwxrwt. 17 root root 4096 мар 14 12:42 tmp
[guest@vbox prog]$
```

Рис. 2.20: Определяем установлен ли атрибут Sticky

2. От имени пользователя guest создайте файл file01.txt в директории /tmp со словом test:(рис. 2.21).

```
[guest@vbox prog]$ echo "test" > /tmp/file01.txt
[guest@vbox prog]$ ls -l /tmp/file01.txt
-rw-r--r--. 1 guest guest 5 мар 14 12:45 /tmp/file01.txt
[guest@vbox prog]$
```

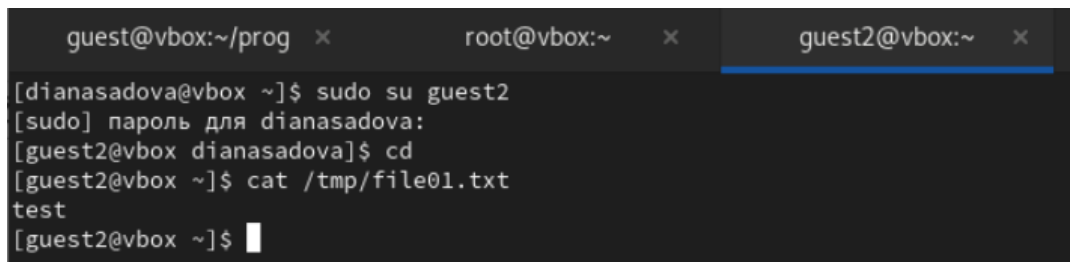
Рис. 2.21: Создаем файл file01.txt

3. Просмотрите атрибуты у только что созданного файла и разрешите чтение и запись для категории пользователей «все остальные»:(рис. 2.22).

```
[guest@vbox prog]$ ls -l /tmp/file01.txt
-rw-r--r--. 1 guest guest 5 мар 14 12:45 /tmp/file01.txt
[guest@vbox prog]$ chmod o+rw /tmp/file01.txt
[guest@vbox prog]$ ls -l /tmp/file01.txt
-rw-r--rw-. 1 guest guest 5 мар 14 12:45 /tmp/file01.txt
[guest@vbox prog]$
```

Рис. 2.22: Просматриваем все атрибуты

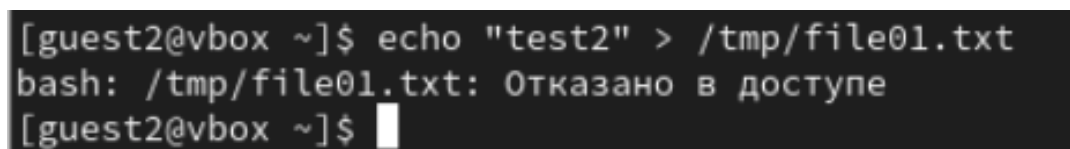
4. От пользователя guest2 (не являющегося владельцем) попробуйте прочитать файл /tmp/file01.txt:(рис. 2.23).



```
guest@vbox:~/prog x      root@vbox:~ x      guest2@vbox:~ x
[dianasadova@vbox ~]$ sudo su guest2
[sudo] пароль для dianasadova:
[guest2@vbox dianasadova]$ cd
[guest2@vbox ~]$ cat /tmp/file01.txt
test
[guest2@vbox ~]$
```

Рис. 2.23: Пробуем читать файл от guest2

5. От пользователя guest2 попробуйте дозаписать в файл /tmp/file01.txt слово test2 командой(рис. 2.24).



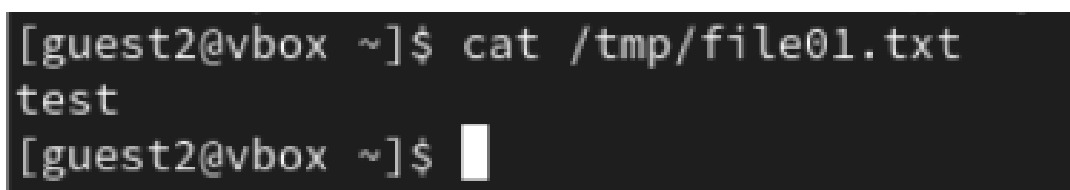
```
[guest2@vbox ~]$ echo "test2" > /tmp/file01.txt
bash: /tmp/file01.txt: Отказано в доступе
[guest2@vbox ~]$
```

Рис. 2.24: Пробуем что-то дописать в файл

Удалось ли вам выполнить операцию?

Нет. Отказано в доступе.

6. Проверьте содержимое файла командой(рис. 2.25).



```
[guest2@vbox ~]$ cat /tmp/file01.txt
test
[guest2@vbox ~]$
```

Рис. 2.25: Проверяем содержимое файла

7. От пользователя guest2 попробуйте записать в файл /tmp/file01.txt слово test3, стараясь при этом всю имеющуюся в файле информацию командой(рис. 2.26).

```
[guest2@vbox ~]$ echo "test3" > /tmp/file01.txt
bash: /tmp/file01.txt: Отказано в доступе
[guest2@vbox ~]$
```

Рис. 2.26: Пробуем что-то дописать в файл

Удалось ли вам выполнить операцию?

Нет. Отказано в доступе.

8. Проверьте содержимое файла командой (рис. 2.27).

```
[guest2@vbox ~]$ cat /tmp/file01.txt
test
[guest2@vbox ~]$
```

Рис. 2.27: Проверяем содержимое файла

9. От пользователя guest2 попробуйте удалить файл /tmp/file01.txt командой (рис. 2.28).

```
[guest2@vbox ~]$ rm /tmp/file01.txt
rm: удалить защищённый от записи обычный файл '/tmp/file01.txt'? y
rm: невозможно удалить '/tmp/file01.txt': Операция не позволена
[guest2@vbox ~]$
```

Рис. 2.28: Пробуем удалить файл

Удалось ли вам удалить файл?

Нет. Отказано в доступе.

10. Повысьте свои права до суперпользователя следующей командой `su -` и выполните после этого команду, снимающую атрибут `t` (Sticky-бит) с директории `/tmp`: (рис. 2.29).

```
[guest@vbox prog]$ su -  
Пароль:  
[root@vbox ~]# chmod -t /tmp  
[root@vbox ~]#
```

Рис. 2.29: Переходим в режим суперпользователя и снимаем атрибут

11. Покиньте режим суперпользователя командой(рис. 2.30).

```
[root@vbox ~]# exit  
выход  
[guest@vbox prog]$
```

Рис. 2.30: Выходим из режима суперпользователя

12. От пользователя guest2 проверьте, что атрибута t у директории /tmp нет:(рис. 2.31).

```
[guest2@vbox ~]$ ls -l / | grep tmp  
drwxrwxrwx. 17 root root 4096 мар 14 12:50 tmp  
[guest2@vbox ~]$
```

Рис. 2.31: Проверяем наличие атрибута t

13. Повторите предыдущие шаги. Какие наблюдаются изменения?(рис. 2.32).



```
[guest2@vbox ~]$ cat /tmp/file01.txt
test
[guest2@vbox ~]$ echo "test3" > /tmp/file01.txt
bash: /tmp/file01.txt: Отказано в доступе
[guest2@vbox ~]$ echo "test2" > /tmp/file01.txt
bash: /tmp/file01.txt: Отказано в доступе
[guest2@vbox ~]$ cat /tmp/file01.txt
test
[guest2@vbox ~]$ rm /tmp/file01.txt
rm: удалить защищённый от записи обычный файл '/tmp/file01.txt'? y
[guest2@vbox ~]$
```

Рис. 2.32: Повторяем предыдущие шаги

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

Да, мы смогли удалить файл

15. Повысьте свои права до суперпользователя и верните атрибут t на директорию /tmp:(рис. 2.33).

```
[guest@vbox prog]$ su -
chmod +t /tmp
exit
Пароль:
```

Рис. 2.33: Возвращаем предыдущие шаги

## 3 Выводы

Изучили механиз изменения идентификаторов, применения SetUID- и Sticky-битов.

Получили практические навыки работы в консоли с дополнительными атрибутами.

Рассмотрели работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов

## **Список литературы**