

## **Отчет по лабораторной работе №5**

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

*Лебедев Ярослав Борисович*

## Содержание

Цель работы .....	3
Выполнение лабораторной работы .....	4
Создание программы.....	4
Исследование Sticky-бита.....	9
Выводы .....	11
Список литературы .....	12

## **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].

## Выполнение лабораторной работы

### Создание программы

1. Войдите в систему от имени пользователя guest (Рис.1).
2. Создайте программу simpleid.c (Рис.1 - Рис.2).

```
[yblebedev@yblebedev ~]$ getenforce
Permissive
[yblebedev@yblebedev ~]$ su guest
Пароль:
[guest@yblebedev yblebedev]$ touch simpleid.c
touch: невозможно выполнить touch для 'simpleid.c': Отказано в доступе
[guest@yblebedev yblebedev]$ cd ~
[guest@yblebedev ~]$ touch simpleid.c
[guest@yblebedev ~]$ ls
dir1 simpleid.c
```

Рис.1. Пункт 1-2



The screenshot shows a code editor window titled 'simpleid.c' with a toolbar containing 'Открыть', 'Сохранить', and a close button. The code is as follows:

```
1 #include <sys/types.h>
2 #include <unistd.h>
3 #include <stdio.h>
4 int
5 main ()
6 {
7     uid_t uid = geteuid ();
8     gid_t gid = getegid ();
9     printf ("uid=%d, gid=%d\n", uid, gid);
10    return 0;
11 }
```

Рис.2. Пункт 2. Программа

3. Скомпилируйте программу и убедитесь, что файл программы создан (Рис.3): `gcc simpleid.c -o simpleid`
4. Выполните программу simpleid (Рис.3): `./simpleid`
5. Выполните системную программу id (Рис.3): `id` и сравните полученный вами результат с данными предыдущего пункта задания.
6. Усложните программу, добавив вывод действительных идентификаторов. Получившуюся программу назовите simpleid2.c (Рис.3 - Рис.4).

```
[guest@yblebedev ~]$ gcc simpleid.c -o simpleid
[guest@yblebedev ~]$ ls
dir1 simpleid simpleid.c
[guest@yblebedev ~]$ ./simpleid
uid=1001, gid=1001
[guest@yblebedev ~]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@yblebedev ~]$ touch simpleid2.c
[guest@yblebedev ~]$ ls
dir1 simpleid simpleid2.c simpleid.c
```

Рис.3. Пункт 3-6



```
1 #include <sys/types.h>
2 #include <unistd.h>
3 #include <stdio.h>
4 int
5 main ()
6 {
7     uid_t real_uid = getuid ();
8     uid_t e_uid = geteuid ();
9     gid_t real_gid = getgid ();
10    gid_t e_gid = getegid ();
11    printf ("e_uid=%d, e_gid=%d\n", e_uid, e_gid);
12    printf ("real_uid=%d, real_gid=%d\n", real_uid,
13           real_gid);
14    return 0;
15 }
```

Рис.4. Пункт 6. Программа

7. Скомпилируйте и запустите simpleid2.c (Рис.5) `gcc simpleid2.c -o simpleid2`  
`./simpleid2`
8. От имени суперпользователя выполните команды (Рис.5): `chown root:guest /home/guest/simpleid2`  
`chmod u+s /home/guest/simpleid2`
9. Используйте `sudo` или повысьте временно свои права с помощью `su` (Рис.5). Поясните, что делают эти команды.
10. Выполните проверку правильности установки новых атрибутов и смены владельца файла simpleid2 (Рис.5): `ls -l simpleid2`

```
[guest@yblebedev ~]$ gcc simpleid2.c -o simpleid2
[guest@yblebedev ~]$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
[guest@yblebedev ~]$ su
Пароль:
[root@yblebedev guest]# chown root:guest /home/guest/simpleid2
[root@yblebedev guest]# chmod u+s /home/guest/simpleid2
[root@yblebedev guest]# ls -l simpleid2
-rwsrwxr-x. 1 root guest 26008 окт  7 13:06 simpleid2
```

Рис.5. Пункт 7-10

11. Запустите simpleid2 и `id`: `./simpleid2 id` Сравните результаты (Рис.6).

```
[root@yblebedev guest]# ./simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@yblebedev guest]# id
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@yblebedev guest]# su guest
[guest@yblebedev ~]$ ./simpleid2
e_uid=0, e_gid=1001
real_uid=1001, real_gid=1001
[guest@yblebedev ~]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

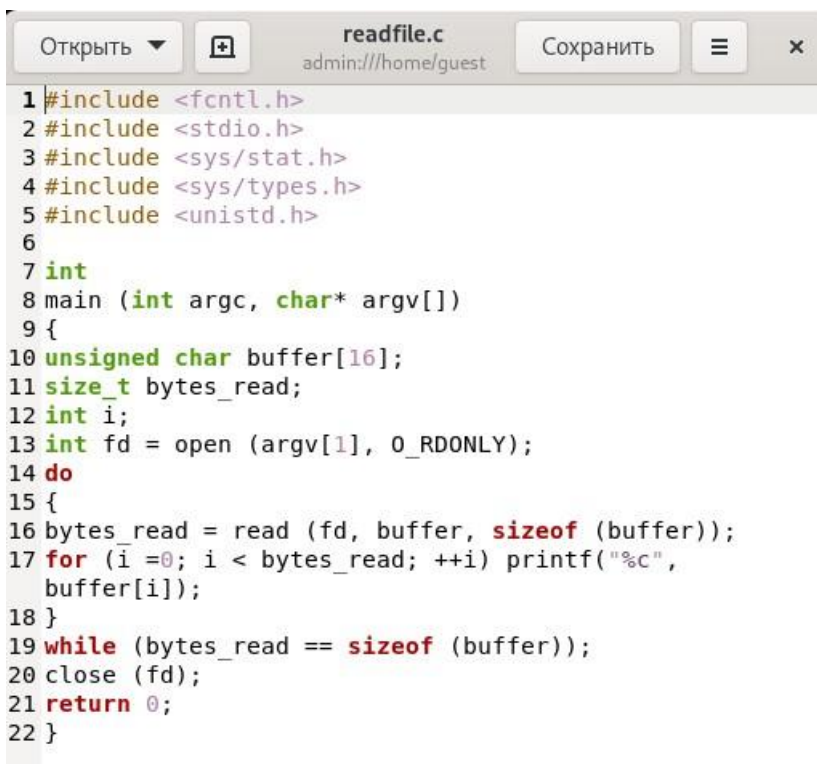
Рис.6. Пункт 11

12. Прodelайте тоже самое относительно SetGID-бита (Рис.7).

13. Создайте программу readfile.c (Рис.7 - Рис.8).

```
[guest@yblebedev ~]$ su
Пароль:
[root@yblebedev guest]# chmod g+s /home/guest/simpleid2
[root@yblebedev guest]# ls -l simpleid2
-rwsrwsr-x. 1 root guest 26008 окт 7 13:06 simpleid2
[root@yblebedev guest]# su guest
[guest@yblebedev ~]$ ls -l simpleid2
-rwsrwsr-x. 1 root guest 26008 окт 7 13:06 simpleid2
[guest@yblebedev ~]$ ./simpleid2
e_uid=0, e_gid=1001
real_uid=1001, real_gid=1001
[guest@yblebedev ~]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@yblebedev ~]$ touch readfile.c
```

Рис.7. Пункт 12-13



```
1 #include <fcntl.h>
2 #include <stdio.h>
3 #include <sys/stat.h>
4 #include <sys/types.h>
5 #include <unistd.h>
6
7 int
8 main (int argc, char* argv[])
9 {
10 unsigned char buffer[16];
11 size_t bytes_read;
12 int i;
13 int fd = open (argv[1], O_RDONLY);
14 do
15 {
16 bytes_read = read (fd, buffer, sizeof (buffer));
17 for (i = 0; i < bytes_read; ++i) printf ("%c",
18     buffer[i]);
19 } while (bytes_read == sizeof (buffer));
20 close (fd);
21 return 0;
22 }
```

Рис.8. Пункт 13. Программа

14. Откомпилируйте её (Рис.9). gcc readfile.c -o readfile

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.9).
16. Проверьте, что пользователь guest не может прочитать файл readfile.c (Рис.9).

```
[guest@yblebedev ~]$ gcc readfile.c -o readfile
[guest@yblebedev ~]$ ls -l
итого 96
drwxrwxr-x. 2 guest guest 19 сен 28 13:04 dir1
-rwxrwxr-x. 1 guest guest 25952 окт 7 13:12 readfile
-rw-rw-r--. 1 guest guest 403 окт 7 13:12 readfile.c
-rwxrwxr-x. 1 guest guest 25904 окт 7 13:04 simpleid
-rwsrwsr-x. 1 root guest 26008 окт 7 13:06 simpleid2
-rw-rw-r--. 1 guest guest 303 окт 7 13:06 simpleid2.c
-rw-rw-r--. 1 guest guest 175 окт 7 13:04 simpleid.c
[guest@yblebedev ~]$ su
Пароль:
[root@yblebedev guest]# chown root:guest /home/guest/readfile.c
[root@yblebedev guest]# ls -l
итого 96
drwxrwxr-x. 2 guest guest 19 сен 28 13:04 dir1
-rwxrwxr-x. 1 guest guest 25952 окт 7 13:12 readfile
-rw-rw-r--. 1 root guest 403 окт 7 13:12 readfile.c
-rwxrwxr-x. 1 guest guest 25904 окт 7 13:04 simpleid
-rwsrwsr-x. 1 root guest 26008 окт 7 13:06 simpleid2
-rw-rw-r--. 1 guest guest 303 окт 7 13:06 simpleid2.c
-rw-rw-r--. 1 guest guest 175 окт 7 13:04 simpleid.c
[root@yblebedev guest]# chmod 600 readfile.c
[root@yblebedev guest]# ls -l
итого 96
drwxrwxr-x. 2 guest guest 19 сен 28 13:04 dir1
-rwxrwxr-x. 1 guest guest 25952 окт 7 13:12 readfile
-rw-----. 1 root guest 403 окт 7 13:12 readfile.c
-rwxrwxr-x. 1 guest guest 25904 окт 7 13:04 simpleid
-rwsrwsr-x. 1 root guest 26008 окт 7 13:06 simpleid2
-rw-rw-r--. 1 guest guest 303 окт 7 13:06 simpleid2.c
-rw-rw-r--. 1 guest guest 175 окт 7 13:04 simpleid.c
[root@yblebedev guest]# su guest
[guest@yblebedev ~]$ cat readfile.c
cat: readfile.c: Отказано в доступе
```

Рис.9. Пункт 14-16

17. Смените у программы readfile владельца и установите SetU'D-бит (Рис.10).
18. Проверьте, может ли программа readfile прочитать файл readfile.c? (Рис.10)



```

[guest@yblebedev ~]$ su
Пароль:
[root@yblebedev guest]# chown root:guest /home/guest/readfile
[root@yblebedev guest]# chmod u+s /home/guest/readfile
[root@yblebedev guest]# ls -l
итого 96
drwxrwxr-x. 2 guest guest    19 сен 28 13:04 dir1
-rwsrwxr-x. 1 root  guest 25952 окт  7 13:12 readfile
-rw----- 1 root  guest   403 окт  7 13:12 readfile.c
-rwxrwxr-x. 1 guest guest 25904 окт  7 13:04 simpleid
-rwsrwsr-x. 1 root  guest 26008 окт  7 13:06 simpleid2
-rw-rw-r-- 1 guest guest   303 окт  7 13:06 simpleid2.c
-rw-rw-r-- 1 guest guest   175 окт  7 13:04 simpleid.c
[root@yblebedev guest]# readfile readfile.c
bash: readfile: command not found...
[root@yblebedev guest]# ./
dir1/      .mozilla/ readfile  simpleid  simpleid2
[root@yblebedev guest]# ./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;
}

```

Рис.10. Пункт 17-18

19. Проверьте, может ли программа readfile прочитать файл /etc/shadow? Отрадите полученный результат и ваши объяснения в отчёте (Рис.11).

```

[root@yblebedev guest]# ./readfile /etc/shadow
root:$6$06hexZzINnLDtBX$ioYYHZBlGsaVwXZUw7B010SBUjTn8PeKzX7PDiGAU16rEs rcL4WHwhyM0SnW/v7ykChqD2nvCQHngnh
BN6P1/...:0:99999:7:::
bin*:19123:0:99999:7:::
daemon*:19123:0:99999:7:::
adm*:19123:0:99999:7:::
lp*:19123:0:99999:7:::
sync*:19123:0:99999:7:::
shutdown*:19123:0:99999:7:::
halt*:19123:0:99999:7:::
mail*:19123:0:99999:7:::
operator*:19123:0:99999:7:::
games*:19123:0:99999:7:::
ftp*:19123:0:99999:7:::
nobody*:19123:0:99999:7:::
systemd-coredump:!!:19244:::
dbus:!!:19244:::
polkitd:!!:19244:::
rtkit:!!:19244:::

```

Рис.11. Пункт 19



## Исследование Sticky-бита

1. Выясните, установлен ли атрибут Sticky на директории /tmp, для чего выполните команду (Рис.12) `ls -l / | grep tmp`
2. От имени пользователя guest создайте файл file01.txt в директории /tmp со словом test (Рис.12): `echo "test" > /tmp/file01.txt`
3. Просмотрите атрибуты у только что созданного файла и разрешите чтение и запись для категории пользователей «все остальные» (Рис.12): `ls -l /tmp/file01.txt`  
`chmod o+rw /tmp/file01.txt` `ls -l /tmp/file01.txt`

```
[root@yblebedev guest]# ls -l / | grep tmp
drwxrwxrwt. 16 root root 4096 окт  7 13:16 tmp
[root@yblebedev guest]# echo "test" > /tmp/file01.txt
[root@yblebedev guest]# ls -l /tmp/file01.txt
-rw-r--r--. 1 root root 5 окт  7 13:18 /tmp/file01.txt
[root@yblebedev guest]# chmod o+rw /tmp/file01.txt
[root@yblebedev guest]# ls -l /tmp/file01.txt
-rw-r--rw-. 1 root root 5 окт  7 13:18 /tmp/file01.txt
[root@yblebedev guest]#
```

Рис.12. Пункт 1-3. Исследование Sticky-бита

4. От пользователя guest2 (не являющегося владельцем) попробуйте прочитать файл /tmp/file01.txt (Рис.13): `cat /tmp/file01.txt`
5. От пользователя guest2 попробуйте дозаписать в файл /tmp/file01.txt слово test2 командой (Рис.13) `echo "test2" > /tmp/file01.txt` Удалось ли вам выполнить операцию?
6. Проверьте содержимое файла командой (Рис.13) `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 (Рис.13) `echo "test3" > /tmp/file01.txt` Удалось ли вам выполнить операцию?
8. Проверьте содержимое файла командой (Рис.13) `cat /tmp/file01.txt`
9. От пользователя guest2 попробуйте удалить файл /tmp/file01.txt командой `rm /tmp/file01.txt` Удалось ли вам удалить файл? (Рис.13)
10. Повысьте свои права до суперпользователя следующей командой (Рис.13) `su -` и выполните после этого команду, снимающую атрибут t (Sticky-бит) с директории /tmp: `chmod -t /tmp`
11. Покиньте режим суперпользователя командой (Рис.13) `exit`
12. От пользователя guest2 проверьте, что атрибута t у директории /tmp нет (Рис.13): `ls -l / | grep tmp`
13. Повторите предыдущие шаги. Какие наблюдаются изменения? (Рис.13)
14. Удалось ли вам удалить файл от имени пользователя, не являющегося его владельцем? Ваши наблюдения занесите в отчёт (Рис.13).

15. Повысьте свои права до суперпользователя и верните атрибут `t` на директорию `/tmp` (Рис.13): `su - chmod +t /tmp exit`

```
[yblebedev@yblebedev ~]$ su guest2
Пароль:
[guest2@yblebedev yblebedev]$ cat /tmp/file01.txt
test
[guest2@yblebedev yblebedev]$ echo "test2" >> /tmp/file01.txt
[guest2@yblebedev yblebedev]$ cat /tmp/file01.txt
test
test2
[guest2@yblebedev yblebedev]$ echo "test3" > /tmp/file01.txt
[guest2@yblebedev yblebedev]$ cat /tmp/file01.txt
test3
[guest2@yblebedev yblebedev]$ rm /tmp/file01.txt
rm: невозможно удалить '/tmp/file01.txt': Нет такого файла или каталога
[guest2@yblebedev yblebedev]$ rm /tmp/file01.txt
rm: невозможно удалить '/tmp/file01.txt': Операция не позволена
[guest2@yblebedev yblebedev]$ su
Пароль:
[root@yblebedev yblebedev]# chmod -t /tmp
[root@yblebedev yblebedev]# exit
exit
[guest2@yblebedev yblebedev]$ ls -l / | grep tmp
drwxrwxrwx. 16 root root 4096 окт  7 13:22 tmp
[guest2@yblebedev yblebedev]$ cat /tmp/file01.txt
test3
[guest2@yblebedev yblebedev]$ rm /tmp/file01.txt
[guest2@yblebedev yblebedev]$ ls -l /tmp/file01.txt
ls: невозможно получить доступ к '/tmp/file01.txt': Нет такого файла или каталога
[guest2@yblebedev yblebedev]$ su
Пароль:
[root@yblebedev yblebedev]# chmod +t /tmp
[root@yblebedev yblebedev]# exit
exit
[guest2@yblebedev yblebedev]$ ls -l / | grep tmp
drwxrwxrwt. 17 root root 4096 окт  7 13:23 tmp
[guest2@yblebedev yblebedev]$
```

Рис.13. Пункт 4-15. Исследование Sticky-бита

## **Выводы**

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

## **Список литературы**

1. Методические материалы курса