

**Лабораторная работа № 5.
Дискреционное разграничение прав в
Linux. Исследование влияния
дополнительных атрибутов**

Дмитревская Софья Алексеевна, НФИбд-01-19

Содержание

1	Цель работы	5
2	Последовательность выполнения работы	6
2.1	Создание программы	6
2.2	Исследование Sticky-бита	12
3	Выводы	15
4	Библиография	16

List of Figures

2.1	создание и редактурa файла	6
2.2	Программа	7
2.3	компиляция и запуск	8
2.4	simpleid2	10
2.5	readfile	11
2.6	Скриншот выполнения	14

List of Tables

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

```
Complete!  
[root@sdmitrevsckaya g  
[root@sdmitrevsckaya g  
Permissive  
[root@sdmitrevsckaya g
```

1. Войдите в систему от имени пользователя guest.

```
[guest@sdmitrevsckaya ~]$ mkdir lab5  
[guest@sdmitrevsckaya ~]$ cd lab5  
[guest@sdmitrevsckaya lab5]$ touch simpleid.c  
[guest@sdmitrevsckaya lab5]$ touch simpleid2.c  
[guest@sdmitrevsckaya lab5]$ touch readfile.c  
[guest@sdmitrevsckaya lab5]$ gedit simpleid.c  
[guest@sdmitrevsckaya lab5]$ gcc simpleid.c  
[guest@sdmitrevsckaya lab5]$ gcc simpleid.c -o simpleid  
[guest@sdmitrevsckaya lab5]$ ./simpleid  
uid=1001, gid=1001  
[guest@sdmitrevsckaya lab5]$  
[guest@sdmitrevsckaya lab5]$ id  
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023  
[guest@sdmitrevsckaya lab5]$
```

Figure 2.1: создание и редакция файла

- 2.
3. Создайте программу 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;
}

```

```

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 }

```

Figure 2.2: Программа

- 4.
5. Скомпилируйте программу и убедитесь, что файл программы создан:

```
gcc simpleid.c -o simpleid
```

6. Выполните программу simpleid:

```
./simpleid
```

7. Выполните системную программу id:

```
id
```

и сравните полученный вами результат с данными предыдущего пункта задания.



```
[guest@sdmitrevskaya ~]$ mkdir lab5
[guest@sdmitrevskaya ~]$ cd lab5
[guest@sdmitrevskaya lab5]$ touch simpleid.c
[guest@sdmitrevskaya lab5]$ touch simpleid2.c
[guest@sdmitrevskaya lab5]$ touch readfile.c
[guest@sdmitrevskaya lab5]$ gedit simpleid.c
[guest@sdmitrevskaya lab5]$ gcc simpleid.c
[guest@sdmitrevskaya lab5]$ gcc simpleid.c -o simpleid
[guest@sdmitrevskaya lab5]$ ./simpleid
uid=1001, gid=1001
[guest@sdmitrevskaya lab5]$
[guest@sdmitrevskaya lab5]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@sdmitrevskaya lab5]$
```

Figure 2.3: компиляция и запуск

8. Усложните программу, добавив вывод действительных идентификаторов:

```
#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;
}

```

Получившуюся программу назовите simpleid2.c.

```

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 }

```

9. Усложните программу

10. Скомпилируйте и запустите simpleid2.c:

```

gcc simpleid2.c -o simpleid2
./simpleid2

```

```

[guest@sdmitrevsckaya lab5]$ gedit simpleid2.c
[guest@sdmitrevsckaya lab5]$ gedit simpleid2.c
[guest@sdmitrevsckaya lab5]$ gcc simpleid2.c
[guest@sdmitrevsckaya lab5]$ gcc simpleid2.c -o simpleid2.c
gcc: fatal error: input file 'simpleid2.c' is the same as output file
compilation terminated.
[guest@sdmitrevsckaya lab5]$ gcc simpleid2.c -o simpleid2
[guest@sdmitrevsckaya lab5]$ su
Password:
[root@sdmitrevsckaya lab5]# su guest
[guest@sdmitrevsckaya lab5]$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
[guest@sdmitrevsckaya lab5]$ █

```

Figure 2.4: simpleid2

11. Создайте программы 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;
}

```

```

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

```

Figure 2.5: readfile

12.

13. Откомпилируйте её.

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

проверка возможности чтения файла

Отразите полученный результат и ваши объяснения в отчёте.

2.2 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 ко- мандой

```
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 ко- ман-
дой

```
rm /tmp/file01.txt
```

10. Повысьте свои права до суперпользователя следующей командой

```
su -
```

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

```
chmod -t /tmp
```

11. Покиньте режим суперпользователя командой exit

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

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

13. Скриншот выполнения:

```
[guest@sdmitrevsckaya lab5]$ echo "test" >> /tmp/file01.txt
[guest@sdmitrevsckaya lab5]$ chmod o+rx /tmp/file01.txt
[guest@sdmitrevsckaya lab5]$ ls -l tmp/file01.txt
ls: cannot access 'tmp/file01.txt': No such file or directory
[guest@sdmitrevsckaya lab5]$ touch file01.txt
[guest@sdmitrevsckaya lab5]$ echo "test" >> /tmp/file01.txt
[guest@sdmitrevsckaya lab5]$ chmod o+rx /tmp/file01.txt
[guest@sdmitrevsckaya lab5]$ ls -l tmp/file01.txt
ls: cannot access 'tmp/file01.txt': No such file or directory
[guest@sdmitrevsckaya lab5]$ ls -l / | grep tmp
drwxrwxrwt. 20 root root 4096 Oct  8 19:42 tmp
[guest@sdmitrevsckaya lab5]$ su guest2
Password:
[guest2@sdmitrevsckaya lab5]$ cat /tmp/file01.txt
test
test
[guest2@sdmitrevsckaya lab5]$ echo "test3" > /tmp/file01.txt
[guest2@sdmitrevsckaya lab5]$ cat /tmp/file01.txt
test3
```

Figure 2.6: Скриншот выполнения

3 Выводы

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.

4 Библиография

1. Методические материалы курса