

Отчёт по лабораторной работе №5

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

Тимур Дмитриевич Калинин

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
2.1	Создание программы	6
2.2	Исследование Sticky-бита	11
3	Выводы	15
4	Библиография	16

List of Figures

2.1	Код simpleid.c	6
2.2	Компиляция и выполнение	6
2.3	Выполнение id	7
2.4	Код simpleid2.c	7
2.5	Компиляция и выполнение simpleid2.c	7
2.6	Установка прав	7
2.7	Проверка	8
2.8	Запуск simpleid2	8
2.9	Установка setGID-бита	8
2.10	Код readfile.c	9
2.11	Компиляция readfile.c	9
2.12	Изменяем владельца	9
2.13	Проверка чтения	9
2.14	Смена владельца	10
2.15	Установка setUID-бита	10
2.16	Проверка чтения readfile.c	10
2.17	Проверка чтения /etc/shadow	11
2.18	Атрибуты	11
2.19	Создание нового файла	11
2.20	Просмотр атрибутов	12
2.21	Установка прав доступа	12
2.22	Чтение файла	12
2.23	Дозапись в файл	12
2.24	Перезапись файла	12
2.25	Проверка содержимого	13
2.26	Попытка удаления	13
2.27	Снятие атрибута t	13
2.28	Проверка атрибутов	13
2.29	Повтор действий	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 Выполнение лабораторной работы

2.1 Создание программы

1. Войдите в систему от имени пользователя guest
2. Создайте программу simpleid.c (Рис. 2.1).

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

Figure 2.1: Код simpleid.c

3. Скомпилируйте программу и убедитесь, что файл программы создан (Рис. 2.2)
4. Выполните программу simpleid (Рис. 2.2).

```
[guest@tdkalinin lab5]$ gcc simpleid.c -o simpleid
[guest@tdkalinin lab5]$ ./simpleid
uid=1001, gid=1001
[guest@tdkalinin lab5]$
```

Figure 2.2: Компиляция и выполнение

5. Выполните системную программу `id` и сравните полученный вами результат с данными предыдущего пункта задания (Рис. 2.3).

```
[guest@tdkalinin lab5]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@tdkalinin lab5]$
```

Figure 2.3: Выполнение `id`

6. Усложните программу, добавив вывод действительных идентификаторов (Рис. 2.4).

```
1  #include <sys/types.h>
2  #include <unistd.h>
3  #include <stdio.h>
4
5  int main ()
6  {
7      uid_t real_uid = getuid();
8      uid_t e_uid = geteuid();
9
10     gid_t real_gid = getgid();
11     gid_t e_gid = getegid();
12
13
14     printf ("uid=%d, gid=%d\n", e_uid, e_gid);
15     printf ("real_uid=%d, real_gid=%d\n", real_uid, real_gid);
16     return 0;
17 }
```

Figure 2.4: Код `simpleid2.c`

7. Скомпилируйте и запустите `simpleid2.c` (Рис. 2.5).

```
[guest@tdkalinin lab5]$ gcc simpleid2.c -o simpleid2
[guest@tdkalinin lab5]$ ./simpleid2
uid=1001, gid=1001
real_uid=1001, real_gid=1001
[guest@tdkalinin lab5]$
```

Figure 2.5: Компиляция и выполнение `simpleid2.c`

8. От имени суперпользователя выполните команды (Рис. 2.6).

```
[root@tdkalinin ~]# chown root:guest /home/guest/ib/lab5/simpleid2
[root@tdkalinin ~]# chown u+s /home/guest/ib/lab5/simpleid2
chown: неверный пользователь: «u+s»
[root@tdkalinin ~]# chmod u+s /home/guest/ib/lab5/simpleid2
[root@tdkalinin ~]#
```

Figure 2.6: Установка прав

9. Используйте `sudo` или повысьте временно свои права с помощью `su`.
10. Выполните проверку правильности установки новых атрибутов и смены владельца файла `simpleid2` (Рис. 2.7).

```
[root@tdkalinin ~]# ls -l /home/guest/ib/lab5/simpleid2
-rwsrwxr-x. 1 root guest 26008 окт  3 14:01 /home/guest/ib/lab5/simpleid2
[root@tdkalinin ~]#
```

Figure 2.7: Проверка

11. Запустите `simpleid2` (Рис. 2.8).

```
[guest@tdkalinin lab5]$ ./simpleid2
uid=0, gid=1001
real_uid=1001, real_gid=1001
[guest@tdkalinin lab5]$
```

Figure 2.8: Запуск `simpleid2`

12. Прodelайте тоже самое относительно SetGID-бита (Рис. 2.9).

```
[root@tdkalinin lab5]# chmod g+s /home/guest/ib/lab5/simpleid2
[root@tdkalinin lab5]# sudo guest2
sudo: guest2: command not found
[root@tdkalinin lab5]# su guest2
[guest2@tdkalinin lab5]$ cd ~/ib/lab5
bash: cd: /home/guest2/ib/lab5: Нет такого файла или каталога
[guest2@tdkalinin lab5]$ cd /home/guest/ib/lab5
[guest2@tdkalinin lab5]$ ./simpleid2
uid=0, gid=1001
real_uid=1002, real_gid=1002
```

Figure 2.9: Установка setGID-бита

13. Создайте программу `readfile.c` (Рис. 2.10).


```

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

```

Figure 2.10: Код readfile.c

14. Откомпилируйте её (Рис. 2.11).

```

[guest@tdkalinin lab5]$ gcc readfile.c -o readfile
[guest@tdkalinin lab5]$

```

Figure 2.11: Компиляция readfile.c

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. 2.12).

```

[root@tdkalinin ~]# chown root:root /home/guest/ib/lab5/readfile.c
[root@tdkalinin ~]# ls -l /home/guest/ib/lab5/readfile.c
-rw-rw-r--. 1 root root 455 окт 3 14:47 /home/guest/ib/lab5/readfile.c
[root@tdkalinin ~]# chmod 660 /home/guest/ib/lab5/readfile.c

```

Figure 2.12: Изменяем владельца

16. Проверьте, что пользователь guest не может прочитать файл readfile.c (Рис. 2.13).

```

[guest@tdkalinin root]$ cat /home/guest/ib/lab5/readfile.c
cat: /home/guest/ib/lab5/readfile.c: Отказано в доступе

```

Figure 2.13: Проверка чтения

17. Смените у программы readfile владельца и установите SetU'D-бит (Рис. 2.14, Рис. 2.15).

```
[root@tdkalinin ~]# chown root /home/guest/ib/lab5/readfile
```

Figure 2.14: Смена владельца

```
[root@tdkalinin ~]# chmod u+s /home/guest/ib/lab5/readfile
[root@tdkalinin ~]# ls -l /home/guest/ib/lab5/readfile
-rwsrwxr-x. 1 root guest 25952 окт  3 14:47 /home/guest/ib/lab5/readfile
[root@tdkalinin ~]#
```

Figure 2.15: Установка setUID-бита

18. Проверьте, может ли программа readfile прочитать файл readfile.c? (Рис. 2.16). Да, может

```
[guest@tdkalinin lab5]$ ./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;
}
```

Figure 2.16: Проверка чтения readfile.c

19. Проверьте, может ли программа readfile прочитать файл /etc/shadow? Отразите полученный результат и ваши объяснения в отчёте (Рис. 2.17). Да, может, так как мы запускаем программу на исполнение от имени владельца, то есть пользователя root.

```

}[guest@tdkalinin lab5]$ ./readfile /etc/shadow
root:$6$tRK2YvsF9HrcmrE$uQGKswcl7ITg6c7ZtsQj4IeSmtMs0TnIYX0ewZVJtlvmc7gs1
27ie77GsCbJzm51EBoIlsyXTqaHpbBq1::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:!!:19241:::
dbus:!!:19241:::
polkitd:!!:19241:::
rtkit:!!:19241:::
sssd:!!:19241:::
avahi:!!:19241:::
pipewire:!!:19241:::
libstoragemgmt:!!:19241:::

```

Figure 2.17: Проверка чтения /etc/shadow

2.2 Исследование Sticky-бита

1. Выясните, установлен ли атрибут Sticky на директории /tmp, для чего выполните команду (Рис. 2.18). Да установлен (буква t)

```

[guest@tdkalinin ~]$ ls -l / | grep tmp
drwxrwxrwt. 24 root root 4096 окт 3 15:26 tmp

```

Figure 2.18: Атрибуты

2. От имени пользователя guest создайте файл file01.txt в директории /tmp со словом test (Рис. 2.19).

```

[guest@tdkalinin ~]$ echo "test" > /tmp/file01.txt
[guest@tdkalinin ~]$

```

Figure 2.19: Создание нового файла

3. Просмотрите атрибуты у только что созданного файла и разрешите чтение и запись для категории пользователей «все остальные» (Рис. 2.20, Рис. 2.21).

```
[guest@tdkalinin ~]$ ls -l /tmp/file01.txt
-rw-rw-r--. 1 guest guest 5 окт  3 15:39 /tmp/file01.txt
[guest@tdkalinin ~]$
```

Figure 2.20: Просмотр атрибутов

```
[guest@tdkalinin ~]$ chmod o+rw /tmp/file01.txt
```

Figure 2.21: Установка прав доступа

4. От пользователя guest2 (не являющегося владельцем) попробуйте прочитать файл /tmp/file01.txt (Рис. 2.22).

```
[guest2@tdkalinin guest]$ cat /tmp/file01.txt
test
[guest2@tdkalinin guest]$
```

Figure 2.22: Чтение файла

5. От пользователя guest2 попробуйте дозаписать в файл /tmp/file01.txt слово test2 (Рис. 2.23).

```
[guest2@tdkalinin guest]$ echo "test2" >> /tmp/file01.txt
[guest2@tdkalinin guest]$
```

Figure 2.23: Дозапись в файл

6. От пользователя guest2 попробуйте записать в файл /tmp/file01.txt слово test3, стерев при этом всю имеющуюся в файле информацию командой (Рис. 2.24).

```
[guest2@tdkalinin guest]$ echo "test3" > /tmp/file01.txt
[guest2@tdkalinin guest]$ cat /tmp/file01.txt
test3
[guest2@tdkalinin guest]$
```

Figure 2.24: Перезапись файла

7. Проверьте содержимое файла командой (Рис. 2.25).

```
[guest2@tdkalinin guest]$ cat /tmp/file01.txt  
test3  
[guest2@tdkalinin guest]$
```

Figure 2.25: Проверка содержимого

8. От пользователя guest2 попробуйте удалить файл /tmp/file01.txt командой (Рис. 2.26).

```
[guest2@tdkalinin guest]$ rm /tmp/file01.txt  
rm: невозможно удалить '/tmp/file01.txt': Операция не позволена  
[guest2@tdkalinin guest]$ su -
```

Figure 2.26: Попытка удаления

9. Повысьте свои права до суперпользователя следующей командой и выполните после этого команду, снимающую атрибут t (Sticky-бит) с директории /tmp (Рис. 2.27).

```
[root@tdkalinin ~]# chmod -t /tmp  
[root@tdkalinin ~]#
```

Figure 2.27: Снятие атрибута t

10. Покиньте режим суперпользователя
11. От пользователя guest2 проверьте, что атрибута t у директории /tmp нет (Рис. 2.28).

```
[guest2@tdkalinin root]$ ls -l / | grep tmp  
drwxrwxrwx. 25 root root 4096 окт  3 15:42 tmp  
[guest2@tdkalinin root]$
```

Figure 2.28: Проверка атрибутов

12. Повторите предыдущие шаги. Какие наблюдаются изменения? (Рис. 2.29).
В этот раз удалось удалить файл.

```
[guest2@tdkalinin root]$ echo "test2" >> /tmp/file01.txt
[guest2@tdkalinin root]$ cat /tmp/file01.txt
test3
test2
[guest2@tdkalinin root]$ echo "test2" > /tmp/file01.txt
[guest2@tdkalinin root]$ cat /tmp/file01.txt
test2
[guest2@tdkalinin root]$ rm /tmp/file01.txt
[guest2@tdkalinin root]$
```

Figure 2.29: Повтор действий

13. Удалось ли вам удалить файл от имени пользователя, не являющегося его владельцем? Да, теперь удалось.

3 Выводы

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

4 Библиография

1. Лабораторная работа №5. - 2 с. URL: https://esystem.rudn.ru/pluginfile.php/1651889/mod_resource/content/2/005-lab_discret_sticky.pdf