

# **Лабораторная работа №5**

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

Азарцова Полина Валерьевна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>6</b>
<b>2</b>	<b>Задание</b>	<b>7</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>8</b>
<b>4</b>	<b>Выводы</b>	<b>17</b>
<b>5</b>	<b>Список литературы</b>	<b>18</b>

## List of Tables

## List of Figures

3.1	Установка gcc . . . . .	8
3.2	Удачное завершение установки gcc . . . . .	8
3.3	Отключение SELinux . . . . .	9
3.4	Создание программы simpleid.c . . . . .	9
3.5	Код программы simpleid.c . . . . .	9
3.6	Компиляция программы . . . . .	9
3.7	Сравнение './simpleid' и 'id' . . . . .	10
3.8	Создание программы . . . . .	10
3.9	Усложненный код программы . . . . .	10
3.10	Компиляция и запуск simpleid2 . . . . .	10
3.11	Смена владельца и установка SetUID . . . . .	11
3.12	Сравнение './simpleid2' и 'id' . . . . .	11
3.13	Установка SetGID . . . . .	11
3.14	Создание readfile.c . . . . .	12
3.15	Код программы readfile.c . . . . .	12
3.16	Компиляция программы readfile.c . . . . .	12
3.17	Смена владельца и изменение прав у readfile.c . . . . .	12
3.18	Смена владельца и установка SetUID . . . . .	13
3.19	Проверка, может ли прочитать readfile.c . . . . .	13
3.20	Проверка, может ли прочитать /etc/shadow . . . . .	13
3.21	Ищем атрибут Sticky . . . . .	14
3.22	Создание file01.txt и установка на него атрибутов . . . . .	14

3.23 Чтение файла . . . . .	14
3.24 Дозапись в файл . . . . .	14
3.25 Перезапись файла . . . . .	15
3.26 Удаление файла . . . . .	15
3.27 Снятие атрибута t (Sticky-бит) . . . . .	15
3.28 Повтор предыдущих шагов без атрибута t . . . . .	16
3.29 Возвращение атрибута t . . . . .	16

# 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 Задание

1. Подготовить к выполнению лабораторной необходимые средства разработки.
2. По порядку выполнить все пункты из раздела “создание программы”.
3. По порядку выполнить все пункты из раздела “исследование Sticky-бита”.

## 3 Выполнение лабораторной работы

1.1 Установила компилятор gcc с помощью команды 'yum install gcc'. (рис - @fig:001, рис - @fig:002).

```
[root@localhost pvazarcova]# yum install gcc
CentOS Linux 8 - AppStream                165 kB/s | 9.6 MB    00:59
CentOS Linux 8 - BaseOS                  365 kB/s | 8.5 MB    00:23
Last metadata expiration check: 0:00:11 ago on Sat 13 Nov 2021 10:55:16 AM MSK.
Dependencies resolved.
=====
Package                Arch      Version              Repository           Size
=====
Installing:
gcc                    x86_64    8.4.1-1.el8          appstream            23 M
Installing dependencies:
cpp                    x86_64    8.4.1-1.el8          appstream            10 M
glibc-devel            x86_64    2.28-151.el8         baseos               1.0 M
glibc-headers          x86_64    2.28-151.el8         baseos              478 k
isl                     x86_64    0.16.1-6.el8         appstream            841 k
kernel-headers         x86_64    4.18.0-305.25.1.el8_4 baseos              7.2 M
libxcrypt-devel        x86_64    4.1.1-4.el8          baseos              25 k
=====
```

Figure 3.1: Установка gcc

```
Installed:
  cpp-8.4.1-1.el8.x86_64
  gcc-8.4.1-1.el8.x86_64
  glibc-devel-2.28-151.el8.x86_64
  glibc-headers-2.28-151.el8.x86_64
  isl-0.16.1-6.el8.x86_64
  kernel-headers-4.18.0-305.25.1.el8_4.x86_64
  libxcrypt-devel-4.1.1-4.el8.x86_64

Complete!
[root@localhost pvazarcova]#
```

Figure 3.2: Удачное завершение установки gcc

1.2 Отключила систему защиты SELinux на текущую сессию командой 'setenforce 0'. Проверила выполнение командой 'getenforce', которая вывела Permissive (рис @fig:003).



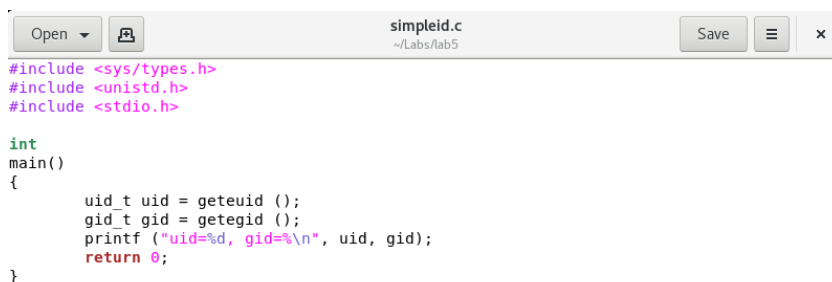
```
[root@localhost pvazarcova]# setenforce 0
[root@localhost pvazarcova]# getenforce
Permissive
```

Figure 3.3: Отключение SELinux

2.1 Вошла в систему от имени пользователя guest, создала программу simpleid.c и папку лабораторной с помощью команд 'touch' и 'mkdir'. (рис @fig:004, рис @fig:005).

```
[guest@localhost ~]$ cd /home/guest/Labs
[guest@localhost Labs]$ mkdir lab5
[guest@localhost Labs]$ cd /home/guest/Labs/lab5
[guest@localhost lab5]$ touch simpleid.c
[guest@localhost lab5]$ gedit simpleid.c
```

Figure 3.4: Создание программы simpleid.c



```
Open simpleid.c Save
~/Labs/lab5

#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>

int
main()
{
    uid_t uid = getuid ();
    gid_t gid = getgid ();
    printf ("uid=%d, gid=%d\n", uid, gid);
    return 0;
}
```

Figure 3.5: Код программы simpleid.c

2.2 Скомпилировала программу и убедилась, что файл программы создан с помощью программы 'gcc simpleid.c -o simpleid'. (рис @fig:006)

```
[guest@localhost lab5]$ gcc simpleid.c -o simpleid
```

Figure 3.6: Компиляция программы

2.3 Выполнила программу simpleid с помощью './simpleid' и системную программу 'id' и сравнила полученный результат (рис. @fig:007)

```
[guest@localhost lab5]$ ./simpleid
uid=1001, gid=%
[guest@localhost lab5]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

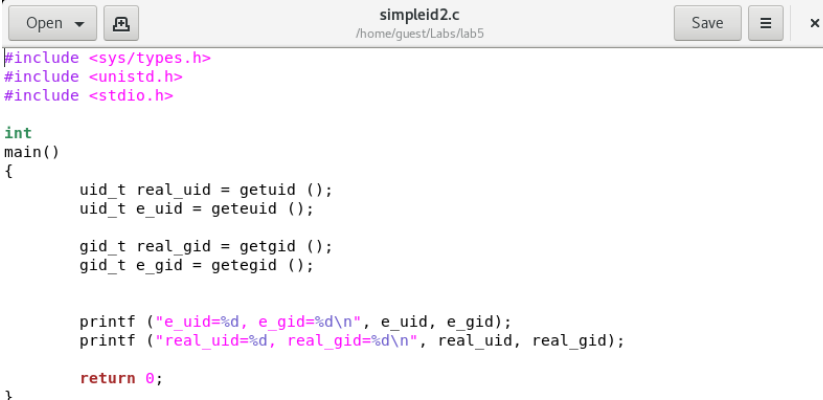
Figure 3.7: Сравнение './simpleid' и 'id'

Полученные значения id совпадают.

2.4 Усложнила программу, добавив вывод действительных идентификаторов, назвала программу simpleid2. (рис @fig:008, рис @fig:009).

```
[guest@localhost lab5]$ touch simpleid2.c
[guest@localhost lab5]$ gedit simpleid2.c
```

Figure 3.8: Создание программы



```
Open simpleid2.c Save
/home/guest/Labs/lab5

#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;
}
```

Figure 3.9: Усложненный код программы

2.5 Скомпилировала и запустила программу simpleid2. (рис. @fig:010)

```
[guest@localhost lab5]$ gcc simpleid2.c -o simpleid2
[guest@localhost lab5]$ ./simpleid2
e_uid=1001, e_gid=1001
real uid=1001, real gid=1001
```

Figure 3.10: Компиляция и запуск simpleid2

2.6 От имени суперпользователя выполнила команды: 'chown root:guest /home/guest/simpleid2' и 'chmod u+s /home/guest/simpleid2', а также выполнила проверку изменений с помощью команды 'ls -l simpleid2'. (рис. @fig:011)

```
[guest@localhost lab5]$ su
Password:
[root@localhost lab5]# chown root:guest simpleid2
[root@localhost lab5]# ls -l
total 48
-rwxrwxr-x. 1 guest guest 17544 Nov 13 12:02 simpleid
-rwxrwxr-x. 1 root  guest 17648 Nov 13 12:29 simpleid2
-rw-rw-r--. 1 guest guest  315 Nov 13 12:29 simpleid2.c
-rw-rw-r--. 1 guest guest  178 Nov 13 12:02 simpleid.c
[root@localhost lab5]# chmod u+s simpleid2
[root@localhost lab5]# ls -l
total 48
-rwxrwxr-x. 1 guest guest 17544 Nov 13 12:02 simpleid
-rwxrwxr-x. 1 root  guest 17648 Nov 13 12:29 simpleid2
-rw-rw-r--. 1 guest guest  315 Nov 13 12:29 simpleid2.c
-rw-rw-r--. 1 guest guest  178 Nov 13 12:02 simpleid.c
[root@localhost lab5]#
```

Figure 3.11: Смена владельца и установка SetUID

Первая команда изменяет права на файл с guest на root. А затем устанавливает атрибут SetUID.

2.7 Запустила simpleid2 и id, сравнила полученные результаты. (рис. @fig:012)

```
[root@localhost lab5]# ./simpleid2
e uid=0, e gid=0
real uid=0, real gid=0
[root@localhost lab5]# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfi
ned t:s0-s0:c0.c1023
```

Figure 3.12: Сравнение './simpleid2' и 'id'

Полученные значения совпадают.

2.8 Проделала то же самое относительно SetGID (установление прав для владеющей группы). (рис @fig:013)

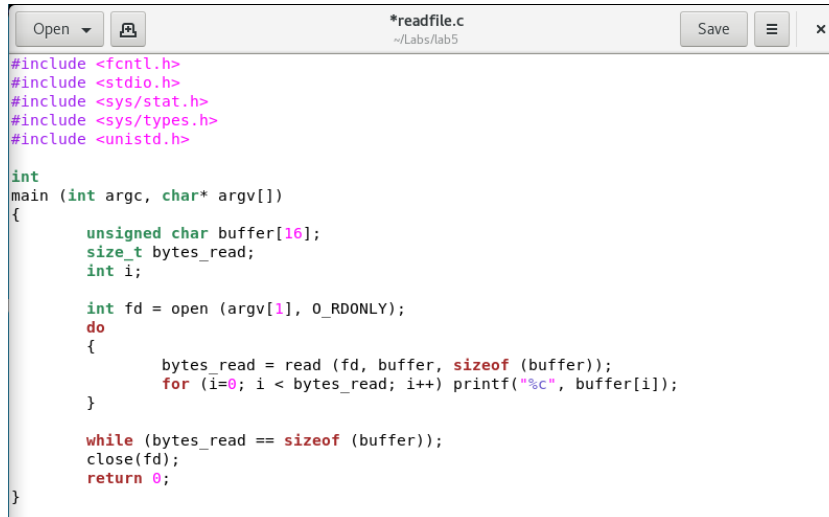
```
[root@localhost lab5]# chmod g+s simpleid2
[root@localhost lab5]# ls -l
total 48
-rwxrwxr-x. 1 guest guest 17544 Nov 13 12:02 simpleid
-rwxr-sr-x. 1 root  root  17648 Nov 13 13:15 simpleid2
-rw-rw-r--. 1 guest guest  315 Nov 13 12:29 simpleid2.c
-rw-rw-r--. 1 guest guest  178 Nov 13 12:02 simpleid.c
```

Figure 3.13: Установка SetGID

2.9 Создала программу readfile.c (рис @fig:014, рис @fig:015)

```
[root@localhost lab5]# su - guest
[guest@localhost ~]$ cd /home/guest/Labs/lab5
[guest@localhost lab5]$ touch readfile.c
[guest@localhost lab5]$ gedit readfile.c
```

Figure 3.14: Создание readfile.c



```
*readfile.c
~/Labs/lab5

#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 3.15: Код программы readfile.c

2.10 Откомпилировала программу readfile.c (рис @fig:016).

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

Figure 3.16: Компиляция программы readfile.c

2.11 Сменила владельца у файла readfile.c и изменила права так, чтобы только суперпользователь мог прочитать его, а guest не мог. Проверила, что пользователь guest не может прочитать файл readfile.c (рис @fig:017)

```
[guest@localhost lab5]$ su
Password:
[root@localhost lab5]# chmod 700 readfile.c
[root@localhost lab5]# su - guest
[guest@localhost ~]$ cd /home/guest/Labs/lab5
[guest@localhost lab5]$ cat readfile.c
cat: readfile.c: Permission denied
```

Figure 3.17: Смена владельца и изменение прав у readfile.c

2.12 Сменила у программы readfile владельца и установила SetUID (рис @fig:018).

```
[guest@localhost lab5]$ su
Password:
[root@localhost lab5]# chown root:guest readfile
[root@localhost lab5]# chmod u+s readfile
[root@localhost lab5]# ls -l
total 72
-rwsrwxr-x. 1 root  guest 17592 Nov 13 13:38 readfile
-rwx----- 1 root  guest  418 Nov 13 13:37 readfile.c
-rwxrwxr-x. 1 guest  guest 17544 Nov 13 12:02 simpleid
-rwxr-sr-x. 1 root   root  17648 Nov 13 13:15 simpleid2
-rw-rw-r-- 1 guest  guest  315 Nov 13 12:29 simpleid2.c
-rw-rw-r-- 1 guest  guest  178 Nov 13 12:02 simpleid.c
```

Figure 3.18: Смена владельца и установка SetUID

2.13 Проверила, может ли программа readfile прочитать файл readfile.c (рис @fig:019).

```
[root@localhost 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 3.19: Проверка, может ли прочитать readfile.c

Может.

2.14 Проверила, может ли программа readfile прочитать файл /etc/shadow (рис @fig:020).

```
[root@localhost ~]# cd /etc
[root@localhost etc]# ./readfile shadow
```

Figure 3.20: Проверка, может ли прочитать /etc/shadow

3.1 Выяснила, установлен ли атрибут Sticky на директории /tmp, для чего выполнила команду 'ls -l / | grep tmp' (рис @fig:021).

```
[guest@localhost lab5]$ ls -l / | grep tmp
drwxrwxrwt. 15 root root 4096 Nov 13 14:24 tmp
```

Figure 3.21: Ищем атрибут Sticky

Установлен.

3.2 От имени пользователя guest создала файл file01.txt в директории /tmp со словом test, просмотрела атрибуты у только что созданного файла и разрешила чтение и запись для категории пользователей «все остальные» (рис @fig:022).

```
[guest@localhost lab5]$ echo "test" > /tmp/file01.txt
[guest@localhost lab5]$ ls -l /tmp/file01.txt
-rw-rw-r--. 1 guest guest 5 Nov 13 14:28 /tmp/file01.txt
[guest@localhost lab5]$ chmod o+rw /tmp/file01.txt
[guest@localhost lab5]$ ls -l /tmp/file01.txt
-rw-rw-rw-. 1 guest guest 5 Nov 13 14:28 /tmp/file01.txt
```

Figure 3.22: Создание file01.txt и установка на него атрибутов

3.3 От пользователя guest2 (не являющегося владельцем) попробовала прочитать файл /tmp/file01.txt с помощью команды 'cat /tmp/file01.txt'. Действие удалось. (рис @fig:023).

```
[guest@localhost lab5]$ su - guest2
Password:
[guest2@localhost ~]$ cat /tmp/file01.txt
test
```

Figure 3.23: Чтение файла

3.4 От пользователя guest2 попробовала дозаписать в файл /tmp/file01.txt слово test2 командой 'echo "test2" » /tmp/file01.txt' и проверила содержимое файла командой 'cat /tmp/file01.txt'. Действие удалось. (рис @fig:024).

```
[guest2@localhost ~]$ echo "test2" >> /tmp/file01.txt
[guest2@localhost ~]$ cat /tmp/file01.txt
test
test2
```

Figure 3.24: Дозапись в файл

3.5 От пользователя guest2 попробовала записать в файл /tmp/file01.txt слово test3, стерев при этом всю имеющуюся в файле информацию командой 'echo "test3" > /tmp/file01.txt'. Действие удалось. (рис @fig:025).

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

Figure 3.25: Перезапись файла

3.6 От пользователя guest2 попробовала удалить файл /tmp/file01.txt командой 'rm /tmp/file01.txt'. Действие не удалось. (рис @fig:026).

```
[guest2@localhost ~]$ rm /tmp/file01.txt
rm: cannot remove '/tmp/file01.txt': Operation not permitted
```

Figure 3.26: Удаление файла

3.7 Повысила свои права до суперпользователя командой 'su -' и выполнила команду 'chmod -t /tmp', снимающую атрибут t с директории /tmp. После покинула режим суперпользователя командой 'exit' и от пользователя guest2 с помощью команды 'ls -l | grep tmp' проверила, что атрибута t у директории /tmp нет. (рис @fig:027).

```
[guest2@localhost ~]$ su -
Password:
[root@localhost ~]# chmod -t /tmp
[root@localhost ~]# exit
logout
[guest2@localhost ~]$ ls -l / | grep tmp
drwxrwxrwx. 15 root root 4096 Nov 13 14:39 tmp
```

Figure 3.27: Снятие атрибута t (Sticky-бит)

3.8 Повторила предыдущие шаги: чтение файла, дозапись в файл, перезапись файла, удаление файла. Удалось всё, включая удаление файла от имени пользователя, не являющегося его владельцем (рис @fig:028).

```

[guest2@localhost ~]$ echo "test2" >> /tmp/file01.txt
[guest2@localhost ~]$ cat /tmp/file01.txt
test3
test2
[guest2@localhost ~]$ echo "test" > /tmp/file01.txt
[guest2@localhost ~]$ cat /tmp/file01.txt
test
[guest2@localhost ~]$ rm /tmp/file01.txt
[guest2@localhost ~]$ ls /tmp
anaconda.log
dbus.log
ifcfg.log
ks-script-q_4k5188
packaging.log
program.log
sensitive-info.log
storage.log
systemd-private-e4c21bc6b7d44dfcbe28dd295ce88aca-color.service-ivpmei
systemd-private-e4c21bc6b7d44dfcbe28dd295ce88aca-geoclue.service-lryemh
systemd-private-e4c21bc6b7d44dfcbe28dd295ce88aca-ModemManager.service-fwVNGh
systemd-private-e4c21bc6b7d44dfcbe28dd295ce88aca-rtkit-daemon.service-WTtTg
tracker-extract-files.1000
tracker-extract-files.1001

```

Figure 3.28: Повтор предыдущих шагов без атрибута t

3.9 Повысила свои права до суперпользователя и вернула атрибут t на директорию /tmp (рис @fig:029).

```

[guest2@localhost ~]$ su
Password:
[root@localhost guest2]# su -
[root@localhost ~]# chmod +t /tmp
[root@localhost ~]# exit
logout

```

Figure 3.29: Возвращение атрибута t



## 4 Выводы

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

## 5 Список литературы

1. Кулябов Д. С., Королькова А. В., Геворкян М. Н. Информационная безопасность компьютерных сетей. Лабораторная работа № 4. Дискреционное разграничение прав в Linux. Расширенные атрибуты