

# **Лабораторная работа № 5**

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

Казакова Виктория Алексеевна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
1.1	### 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. . .	5
	<pre>[guest2@victoria22 ~]\$ su Пароль: [root@victoria22 guest2]# chmod -t /tmp [root@victoria22 guest2]# exit exit [guest2@victoria22 ~]\$ ls -l /   grep tmp drwxrwxrwx. 31 root root 4096 окт 7 17:52 tmp [guest2@victoria22 ~]\$ rm /tmp/file01.txt [guest2@victoria22 ~]\$ cat /tmp/file01.txt cat: /tmp/file01.txt: Нет такого файла или каталога [guest2@victoria22 ~]\$ █</pre>	
1.2	. . . . .	13
<b>2</b>	<b>Выводы</b>	<b>14</b>
<b>3</b>	<b>Библиография</b>	<b>15</b>

# Список иллюстраций

1.1	Подготовка лабораторного стенда . . . . .	5
1.2	Файл simpleid.c . . . . .	6
1.3	Файл скомпилирован . . . . .	6
1.4	Результаты запуска . . . . .	7
1.5	Программа simpleid2.c . . . . .	7
1.6	Результат работы simpleid2.c . . . . .	7
1.7	Чтение файла . . . . .	8
1.8	Проверка правильности . . . . .	8
1.9	Попытка стереть содержимое файла . . . . .	8
1.10	Изменение прав . . . . .	9
1.11	Программа readfile.c . . . . .	9
1.12	Применение команд . . . . .	10
1.13	Смена владельца и ограничений . . . . .	10
1.14	Чтение файла readfile.c . . . . .	11
1.15	Чтение файла /etc/shadow . . . . .	11
1.16	Применение команд, заполнение файла . . . . .	12
1.17	Просмотр и установка атрибутов . . . . .	12
1.18	Чтение и дозапись в файл . . . . .	12
1.19	Чтение и дозапись в файл . . . . .	13
1.20	Попытка удалить файл . . . . .	13

## Список таблиц

# 1 Цель работы

## 1.1 ### 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. Подготовила лабораторный стенд, выполнила все необходимые команды из пункта 5.2.1. (рис. 1.1)

```
[root@victoria22 victoria22]# gcc -v
Используются внутренние спецификации.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/usr/libexec/gcc/x86_64-redhat-linux/4.8.5/lto-wrapper
Целевая архитектура: x86_64-redhat-linux
Параметры конфигурации: ../configure --prefix=/usr --mandir=/usr/share/man --infodir=
/usr/share/info --with-bugurl=http://bugzilla.redhat.com/bugzilla --enable-bootstrap
--enable-shared --enable-threads=posix --enable-checking=release --with-system-zlib -
--enable-__cxa_atexit --disable-libunwind-exceptions --enable-gnu-unique-object --enab
le-linker-build-id --with-linker-hash-style=gnu --enable-languages=c,c++,objc,obj-c++
,java,fortran,ada,go,lto --enable-plugin --enable-initfini-array --disable-libgcj --w
ith-isl=/builddir/build/BUILD/gcc-4.8.5-20150702/obj-x86_64-redhat-linux/isl-install
--with-cloog=/builddir/build/BUILD/gcc-4.8.5-20150702/obj-x86_64-redhat-linux/cloog-i
ninstall --enable-gnu-indirect-function --with-tune=generic --with-arch_32=x86_64 --bui
ld=x86_64-redhat-linux
Модель многопоточности: posix
gcc версия 4.8.5 20150623 (Red Hat 4.8.5-44) (GCC)
[root@victoria22 victoria22]# setenforce 0
[root@victoria22 victoria22]# getenforce
Permissive
```

Рис. 1.1: Подготовка лабораторного стенда

2. Вошла в пользователя guest. Создала программу simpleid.c командой touch (рис. 1.2)

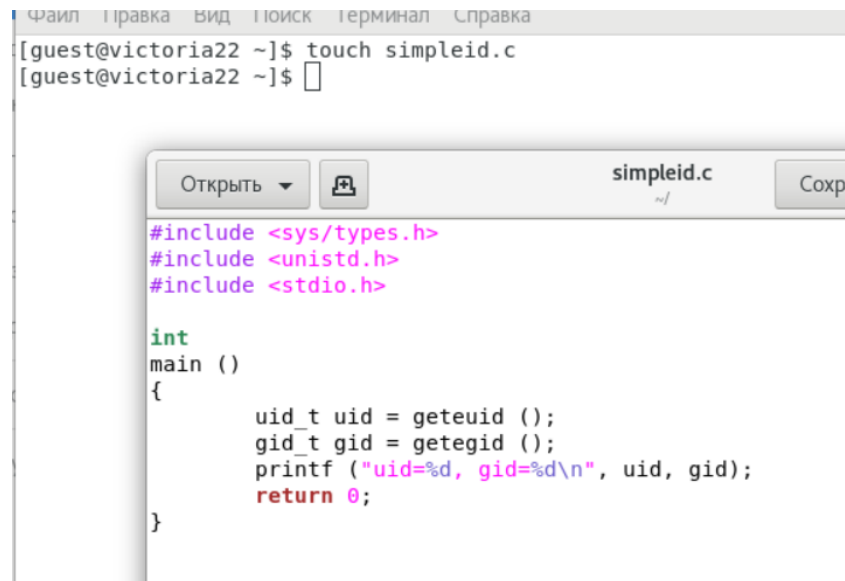


Рис. 1.2: Файл simpleid.c

3. Скомпилировала программу и убедилась, что файл программы создан(рис. 1.3)



Рис. 1.3: Файл скомпилирован

4. Выполнила программу simpleid командой ./simpleid. Выполнила системную программу id. Результат совпал(рис. 1.4)

```
[guest@victoria22 ~]$ ./simpleid
uid=1001, gid=1001
[guest@victoria22 ~]$ id
uid=1001(guest) gid=1001(guest) rpyнны=1001(guest) контекст=unconfined_u:unconfi
ned_r:unconfined_t:s0-s0:c0.c1023
[guest@victoria22 ~]$ █
```

Рис. 1.4: Результаты запуска

5. Создала программу simpleid2.c, содержащую в себе усложненный код программы simpleid.c. (рис. 1.5)

```
#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. 1.5: Программа simpleid2.c

6. Скомпилировала и запустила simpleid2.c (рис. 1.6).

```
[guest@victoria22 ~]$ gcc simpleid2.c -o simpleid2
[guest@victoria22 ~]$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
[guest@victoria22 ~]$ █
```

Рис. 1.6: Результат работы simpleid2.c

7. От имени суперпользователя (перешла в него командой `su`) выполнила команды `chown root:guest /home/guest/simpleid2` и `chmod u+s /home/guest/simpleid2`. Эти команды установили права над файлом суперпользователя (рис. 1.7).

```
[guest@victoria22 ~]$ su
Пароль:
[root@victoria22 guest]# chown root:guest /home/guest/simpleid2
[root@victoria22 guest]# chmod u+s /home/guest/simpleid2
[root@victoria22 guest]# sudo
usage: sudo -h | -K | -k | -V
usage: sudo -v [-AknS] [-g group] [-h host] [-p prompt] [-u user]
usage: sudo -l [-AknS] [-g group] [-h host] [-p prompt] [-U user] [-u user]
[command]
usage: sudo [-AbEHknPS] [-r role] [-t type] [-C num] [-g group] [-h host] [-p
prompt] [-T timeout] [-u user] [VAR=value] [-i|-s] [<command>]
usage: sudo -e [-AknS] [-r role] [-t type] [-C num] [-g group] [-h host] [-p
prompt] [-T timeout] [-u user] file ...
[root@victoria22 guest]#
```

Рис. 1.7: Чтение файла

8. Выполнила проверку правильности установки новых атрибутов и смены владельца файла `simpleid2` (рис. 1.8).

```
[root@victoria22 guest]# ls -l simpleid2
-rwsrwxr-x. 1 root guest 8616 окт  7 17:01 simpleid2
[root@victoria22 guest]#
```

Рис. 1.8: Проверка правильности

9. Запустила `simpleid2` и `id`. Результат совпал (рис. 1.9).

```
[root@victoria22 guest]# ./simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@victoria22 guest]# id
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:unconf
ined_t:s0-s0:c0.c1023
```

Рис. 1.9: Попытка стереть содержимое файла

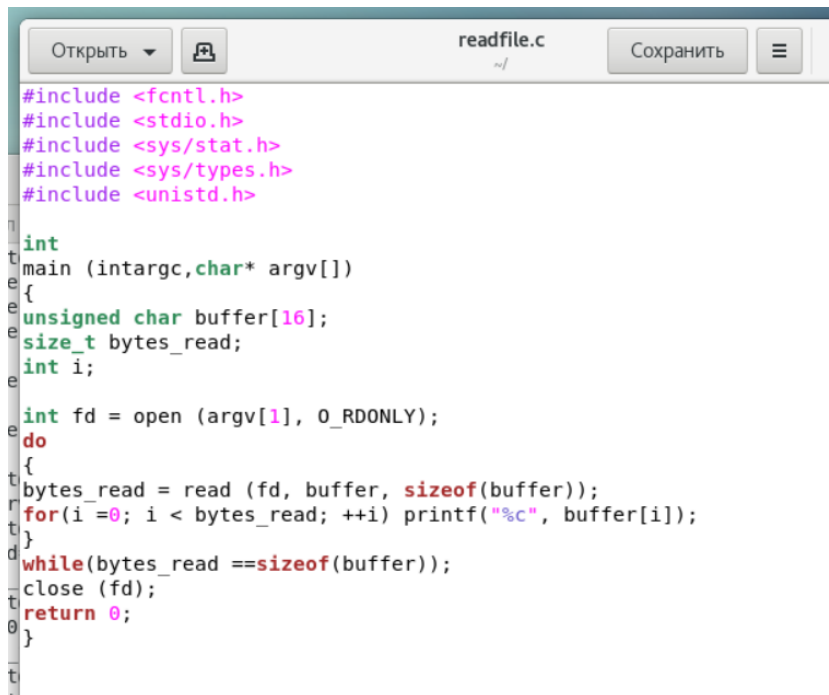
10. Проделала тоже самое относительно SetGID-бита (рис. 1.10).



```
[root@victoria22 guest]# chmod g+s /home/guest/simpleid2
[root@victoria22 guest]# ls -l simpleid2
-rwsrwsr-x. 1 root guest 8616 окт  7 17:01 simpleid2
[root@victoria22 guest]#
```

Рис. 1.10: Изменение прав

11. Создала программу readfile.c (рис. 1.11).



```

#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. 1.11: Программа readfile.c

12. Откомпилировала её. Сменила владельца у файла readfile.c и изменила права так, чтобы только суперпользователь (root) мог прочитать его, а guest не мог. Убедилась, что пользователь guest не может прочитать файл readfile.c (рис. 1.12)

```

[guest@victoria22 ~]$ su
Пароль:
[root@victoria22 guest]# chown root:guest readfile.c
[root@victoria22 guest]# chmod 700 readfile.c
[root@victoria22 guest]# exit
exit
[guest@victoria22 ~]$ cat readfile.c
cat: readfile.c: Отказано в доступе
[guest@victoria22 ~]$

```

Рис. 1.12: Применение команд

13. Сменила у программы readfile владельца и установила SetU'D-бит (рис. 1.13)

```

[guest@victoria22 ~]$ su
Пароль:
[root@victoria22 guest]# chown root:guest readfile
[root@victoria22 guest]# chmod u+s readfile
[root@victoria22 guest]# exit
exit
[guest@victoria22 ~]$

```

Рис. 1.13: Смена владельца и ограничений

14. Убедилась, что программа readfile может прочитать файл readfile.c (рис. 1.14) и файл /etc/shadow (рис. 1.15).

```
[guest@victoria22 ~]$ ./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;
}
[guest@victoria22 ~]$
```

Рис. 1.14: Чтение файла readfile.c

```
[guest@victoria22 ~]$ ./readfile /etc/shadow
root:$6$bHeuHZMEZvuI939y$gIfP4WQNZAu1EUW6QRv.ZrxZmEIInC.q55u5KIZYEN13EZ2s7/zKV9
isy8Twjha2WzPqnXLP8Eko5b1t9JD1::0:99999:7:::
bin:!:18353:0:99999:7:::
daemon:!:18353:0:99999:7:::
adm:!:18353:0:99999:7:::
lp:!:18353:0:99999:7:::
sync:!:18353:0:99999:7:::
shutdown:!:18353:0:99999:7:::
halt:!:18353:0:99999:7:::
mail:!:18353:0:99999:7:::
operator:!:18353:0:99999:7:::
games:!:18353:0:99999:7:::
ftp:!:18353:0:99999:7:::
nobody:!:18353:0:99999:7:::
systemd-network:!!:19607:::::::
dbus:!!:19607:::::::
polkitd:!!:19607:::::::
libstoragemgmt:!!:19607:::::::
colord:!!:19607:::::::
rpc:!!:19607:0:99999:7:::
saned:!!:19607:::::::
```

Рис. 1.15: Чтение файла /etc/shadow

15. Приступила ко второй части работы. Выяснила, установлен ли атрибут Sticky на директории /tmp, для чего выполнила команду `ls -l / | grep tmp` и от имени пользователя guest создала файл file01.txt в директории /tmp со словом test(рис. 1.16).

```
[guest@victoria22 ~]$ ls -l / | grep tmp
drwxrwxrwt. 28 root root 4096 окт  7 17:36 tmp
[guest@victoria22 ~]$ echo "test" > /tmp/file01.txt
[guest@victoria22 ~]$
```

Рис. 1.16: Применение команд, заполнение файла

16. Просмотрела атрибуты у только что созданного файла и разрешила чтение и запись для категории пользователей «все остальные» (рис. 1.17).

```
[guest@victoria22 ~]$ ls -l /tmp/file01.txt
-rw-rw-r--. 1 guest guest 5 окт  7 17:43 /tmp/file01.txt
[guest@victoria22 ~]$ chmod o+rw /tmp/file01.txt
[guest@victoria22 ~]$ ls -l /tmp/file01.txt
-rw-rw-rw-. 1 guest guest 5 окт  7 17:43 /tmp/file01.txt
[guest@victoria22 ~]$
```

Рис. 1.17: Просмотр и установка атрибутов

17. От пользователя guest2 попробовала прочитать файл /tmp/file01.txt и дозаписать в файл слово test2. Снова проверила содержимое файла (рис. 1.18). Файл получается прочитать, но не получается дозаписать в него текст. Команда его стирает и пишет новый

```
[guest2@victoria22 ~]$ cat /tmp/file01.txt
test
[guest2@victoria22 ~]$ echo "test2" > /tmp/file01.txt
[guest2@victoria22 ~]$ cat /tmp/file01.txt
test2
```

Рис. 1.18: Чтение и дозапись в файл

18. От пользователя guest2 записала в файл /tmp/file01.txt слово test3, стерев при этом всю имеющуюся в файле информацию командой, а затем прочитала его (рис. 1.19). Команду выполнить удалось.

```

~~~~~
[guest2@victoria22 ~]$ echo "test3" > /tmp/file01.txt
[guest2@victoria22 ~]$ cat /tmp/file01.txt
test3
[guest2@victoria22 ~]$

```

Рис. 1.19: Чтение и дозапись в файл

19. От пользователя guest2 попробовала удалить файл /tmp/file01.txt командой `rm /tmp/file01.txt` (рис. 1.20). Команду выполнить не удалось.

```

[guest2@victoria22 ~]$ rm /tmp/file01.txt
rm: невозможно удалить «/tmp/file01.txt»: Операция не позволена
[guest2@victoria22 ~]$ █

```

Рис. 1.20: Попытка удалить файл

20. Повысила права до суперпользователя и выполнила после этого команду, снимающую атрибут `t` (Sticky-бит) с директории /tmp. Затем покинула режим суперпользователя и от пользователя guest2 проверьте, что атрибута `t` у директории /tmp нет. Затем снова попыталась удалить файл (рис. 1.19). Файл удалить получилось.

```

[guest2@victoria22 ~]$ su
Пароль:
[root@victoria22 guest2]# chmod -t /tmp
[root@victoria22 guest2]# exit
exit
[guest2@victoria22 ~]$ ls -l / | grep tmp
drwxrwxrwx. 31 root root 4096 окт 7 17:52 tmp
[guest2@victoria22 ~]$ rm /tmp/file01.txt
[guest2@victoria22 ~]$ cat /tmp/file01.txt
cat: /tmp/file01.txt: Нет такого файла или каталога
[guest2@victoria22 ~]$ █

```

## 1.2

## 2 Выводы

В результате выполнения работы я изучила механизм изменения идентификаторов, применения SetUID- и Sticky-битов. Получила практические навыки работы в консоли с дополнительными атрибутами. Рассмотрела работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов

### 3 Библиография

1. [Методический материал] [[https://esystem.rudn.ru/pluginfile.php/2090279/mod\\_resource/content/1/lab\\_discret\\_sticky.pdf](https://esystem.rudn.ru/pluginfile.php/2090279/mod_resource/content/1/lab_discret_sticky.pdf)]
2. [Сайт для поиска команд] [<https://www.ibm.com/>]