Лабораторная работа №5

Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

Липатникова М.С. группа НФИбд-02-19

Содержание

List of Figures

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

Удостоверилась, что установлен gcc (fig. 2.1). Вошла в систему от имени пользователя guest. Создала 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;

}
```

Скомплилировала программу и убедилась, что файл программы создан: gcc simpleid.c -o simpleid. Выполнила программу simpleid: ./simpleid. Выполнила

системную программу id: id. Полученный результат с данными предыдущего пункта задания совпадает (fig. 2.2).

```
[root@mslipatnikova ~]# yum install gcc
Last metadata expiration check: 0:00:13 ago on Sat 08 Oct 2022 04:05:49 PM MSK.
Package gcc-11.2.1-9.4.el9.x86_64 is already installed.
```

Figure 2.1: Проверка gcc

```
[guest@mslipatnikova ~]$ touch simpleid.c
[guest@mslipatnikova ~]$ nano simpleid.c
[guest@mslipatnikova ~]$ cat 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;
}
[guest@mslipatnikova ~]$ gcc simpleid.c -o simpleid
[guest@mslipatnikova ~]$ ./simpleid
uid=1001, gid=1001
[guest@mslipatnikova ~]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfin
ed_r:unconfined_t:s0-s0:c0.c1023
```

Figure 2.2: Работа с программой simpleid

Усложнила 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 назвала simpleid2.c. Скомпилировала и запустила simpleid2.c (fig. 2.3):

gcc simpleid2.c -o simpleid2

./simpleid2

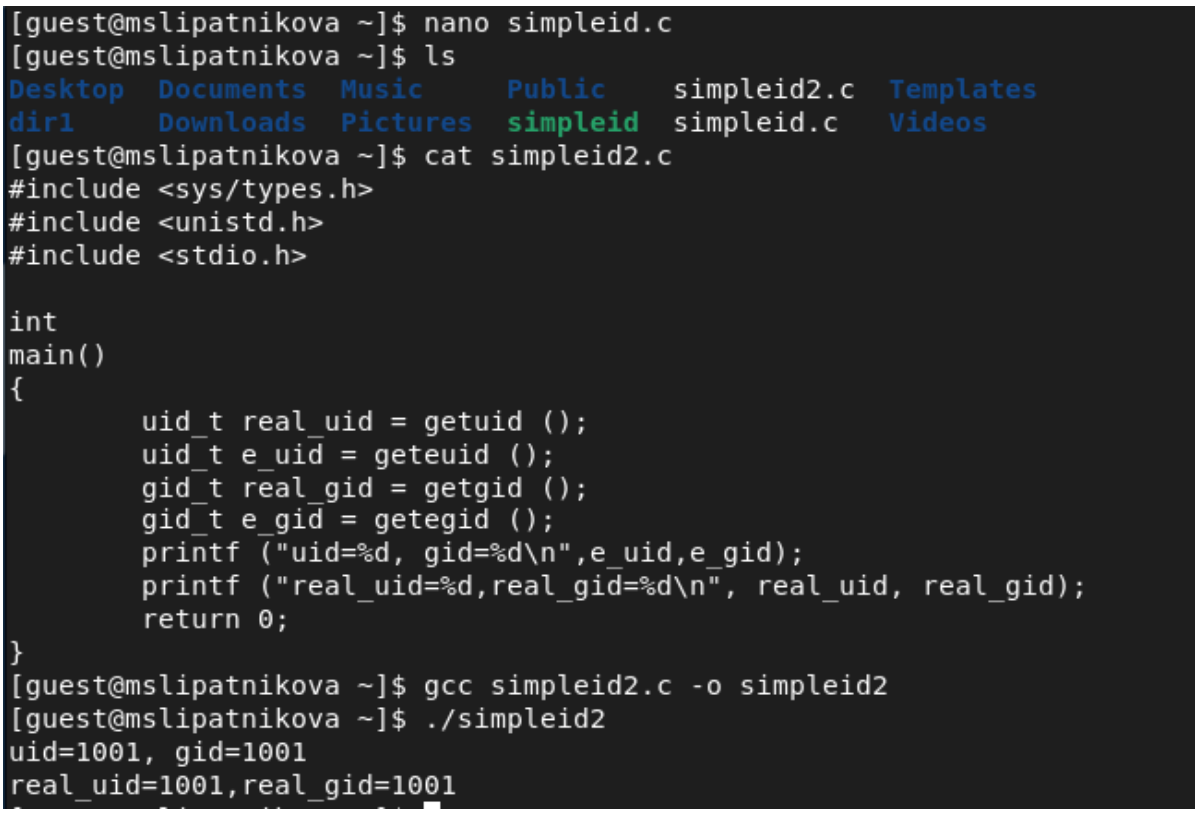

Figure 2.3: Работа с программой simpleid2

От имени суперпользователя выполнила команды (fig. 2.6):

chown root:guest /home/guest/simpleid2

chmod u+s /home/guest/simpleid2

Выполнила проверку правильности установки новых атрибутов и смены владельца файла simpleid2: ls -l simpleid2. Запустила simpleid2 и id:

./simpleid2

id

Замечаем, что все совпадает, кроме uid = 0 (fig. 2.4).

Проделала тоже самое относительно SetGID-бита (fig. 2.5).

```
[guest@mslipatnikova ~]$ ls -l simpleid2
-rwsrwxr-x. 1 root guest 26008 Oct  8 16:21 simpleid2
[guest@mslipatnikova ~]$ ./simpleid2
uid=0, gid=1001
real_uid=1001,real_gid=1001
[guest@mslipatnikova ~]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfi
ned_r:unconfined_t:s0-s0:c0.c1023
```

Figure 2.4: Работа simpleid2(u+s)

```
[guest@mslipatnikova ~]$ ls -l simpleid2
-rwsrwsr-x. 1 root guest 26008 Oct  8 16:21 simpleid2
[guest@mslipatnikova ~]$ ./simpleid2
uid=0, gid=1001
real_uid=1001,real_gid=1001
[guest@mslipatnikova ~]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfi
ned_r:unconfined_t:s0-s0:c0.c1023
```

Figure 2.5: Работа simpleid2(g+s)

```
[root@mslipatnikova ~]# chown root:guest /home/guest/simpleid2
[root@mslipatnikova ~]# chmod u+s /home/guest/simpleid2
[root@mslipatnikova ~]# chmod g+s /home/guest/simpleid2
```

Figure 2.6: Команды от суперпользователя

Создала программу 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;

}
```

Откомпилировала её: gcc readfile.c -o readfile (fig. 2.7). Сменила владельца у файла readfile.c и изменила права так, чтобы только суперпользователь (root) мог прочитать его, а guest не мог (fig. 2.9). Проверила, что пользователь guest

не может прочитать файл readfile.c (fig. 2.8). Сменила у программы readfile владельца и установила SetU'D-бит (fig. 2.9). Проверила, может ли программа readfile прочитать файл readfile.c (может) (fig. 2.10). Проверила, может ли программа readfile прочитать файл /etc/shadow (может) (fig. 2.10).

```
[guest@mslipatnikova ~]$ touch readfile.c
[guest@mslipatnikova ~]$ nano readfile.c
[guest@mslipatnikova ~]$ cat 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@mslipatnikova ~]$ gcc readfile.c -o readfile
```

Figure 2.7: Программа readfile

```
[guest@mslipatnikova ~]$ ls -l readfile.c
-rwx------. 1 root guest 418 Oct  8 17:10 readfile.c
[guest@mslipatnikova ~]$ cat readfile.c
cat: readfile.c: Permission denied
```

Figure 2.8: Чтение readfile.c

```
[root@mslipatnikova ~]# chown root /home/guest/readfile.c
[root@mslipatnikova ~]# chmod 700 /home/guest/readfile.c
[root@mslipatnikova ~]# chown root:guest /home/guest/readfile
[root@mslipatnikova ~]# chmod u+s /home/guest/readfile
[root@mslipatnikova ~]# chmod g+s /home/guest/readfile
```

Figure 2.9: Команды от суперпользователя

Figure 2.10: Чтение с помощью программы readfile

Выяснила, установлен ли атрибут Sticky на директории /tmp, для чего выполнила команду: ls -l / | grep tmp. От имени пользователя guest создала файл file01.txt в директории /tmp со словом test: echo “test” > /tmp/file01.txt.

Просмотрела атрибуты у только что созданного файла и разрешила чтение и запись для категории пользователей «все остальные»:

ls -l /tmp/file01.txt

chmod o+rw /tmp/file01.txt

ls -l /tmp/file01.txt

От пользователя guest2 (не являющегося владельцем) попробовала прочитать файл /tmp/file01.txt: cat /tmp/file01.txt. От пользователя guest2 попробовала дозаписать в файл /tmp/file01.txt слово test2 командой: echo “test2” » /tmp/file01.txt. Удалось выполнить операцию. Проверила содержимое файла командой: cat /tmp/file01.txt. 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. Удалось выполнить операцию. Проверила содержимое файла командой: cat /tmp/file01.txt. От пользователя guest2 попробовала удалить файл /tmp/file01.txt командой: rm /tmp/fileOl.txt. Не удалось удалить файл (fig. 2.11).

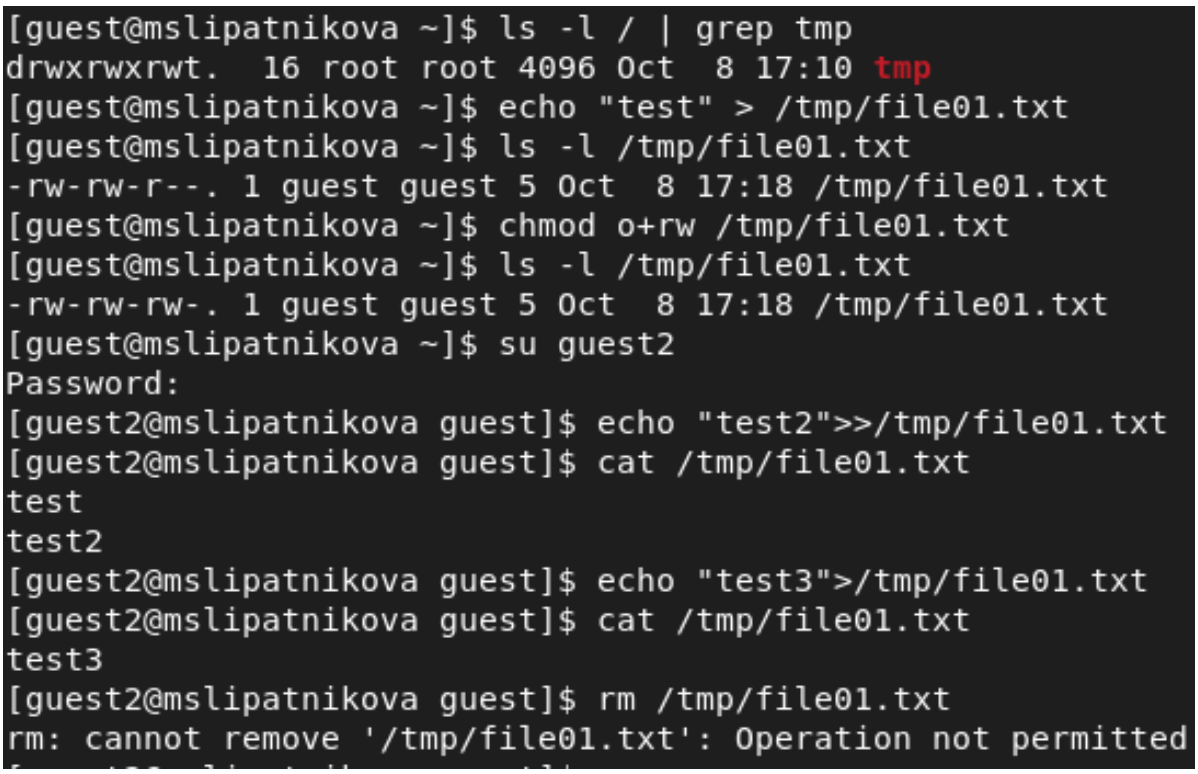

Figure 2.11: Работа с tmp/file01.txt

От суперпользователя выполнила команду, снимающую атрибут t (Sticky-бит) с директории /tmp: chmod -t /tmp (fig. 2.13). От пользователя guest2 проверила, что атрибута t у директории /tmp нет: ls -l / | grep tmp. Повторила предыдущие шаги. Удалось в этот раз удалить файл от имени пользователя, не являющегося его владельцем (fig. 2.12). От суперпользователя выполнила команду, вернувший атрибут t (Sticky-бит) в директории /tmp: chmod +t /tmp (fig. 2.13).

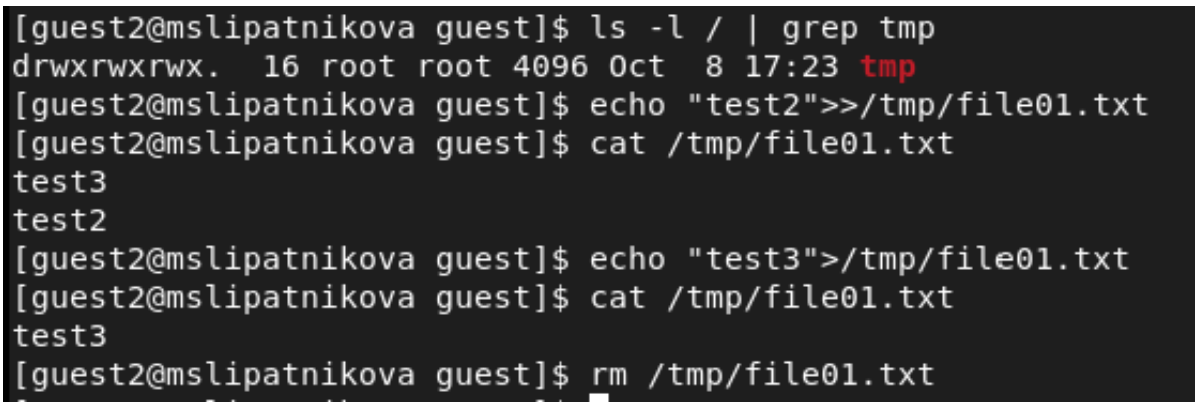

Figure 2.12: Работа с tmp/file01.txt без t

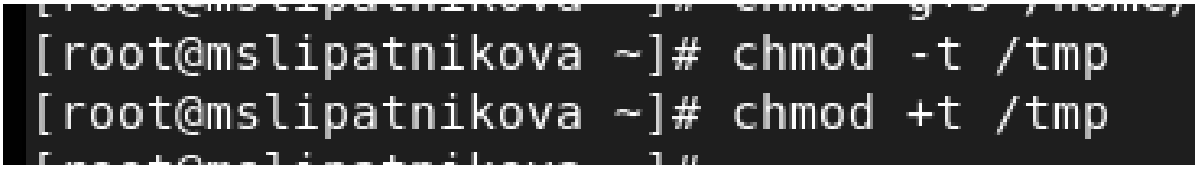

Figure 2.13: Команды от суперпользователя

3 Вывод

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.

4 Список литературы

1. Теоретические материалы курса.