

Лабораторная работа №5

Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов.

Легиньких Г.А.

Российский университет дружбы народов, Москва, Россия

Информация

- Легиньких Галина Андреевна
- НФИбд-02-21
- Российский университет дружбы народов
- 1032216447@pfur.ru
- <https://github.com/galeginkikh>

Выполнение

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.

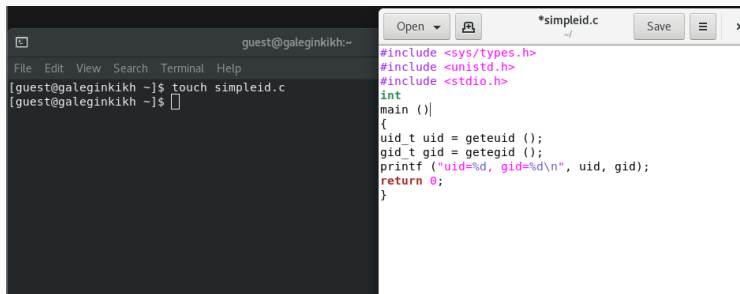
Выполнение лабораторной работы

1. Установила компилятор gcc. Отключила систему защиты до очередной перезагрузки системы.

```
[guest@galeginkikh ~]$ su
Password:
[root@galeginkikh guest]# yum install gcc
Rocky Linux 8 - AppStream                    5.0 kB/s | 4.8 kB      00:00
Rocky Linux 8 - AppStream                    3.0 MB/s | 12 MB      00:04
Rocky Linux 8 - BaseOS                       5.3 kB/s | 4.3 kB      00:00
Rocky Linux 8 - BaseOS                      2.7 MB/s | 6.1 MB      00:02
Rocky Linux 8 - Extras                      5.0 kB/s | 3.1 kB      00:00
Rocky Linux 8 - Extras                      17 kB/s | 14 kB      00:00
Package gcc-8.5.0-22.el8_10.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@galeginkikh guest]# setenforce 0
[root@galeginkikh guest]# getenforce
Permissive
[root@galeginkikh guest]#
```

Рис. 1: Компилятор gcc

2. Вошла в систему от имени пользователя guest.
3. Создала программу simpleid.c.



The image shows a terminal window on the left and a code editor on the right. The terminal window has a title bar 'guest@galeginkikh:~' and a menu bar 'File Edit View Search Terminal Help'. It shows the command 'touch simpleid.c' being executed. The code editor has a title bar '*simpleid.c' and buttons for 'Open', 'Save', and a close button. It contains the following C code:

```
#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;
}
```

Рис. 2: файл simpleid.c

4. Скомпилировала программу и убедилась, что файл программы создан.

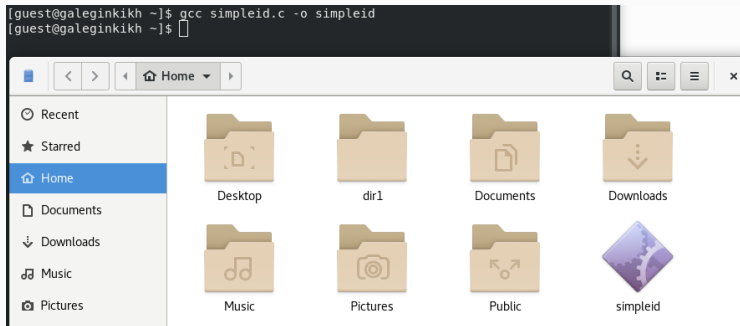


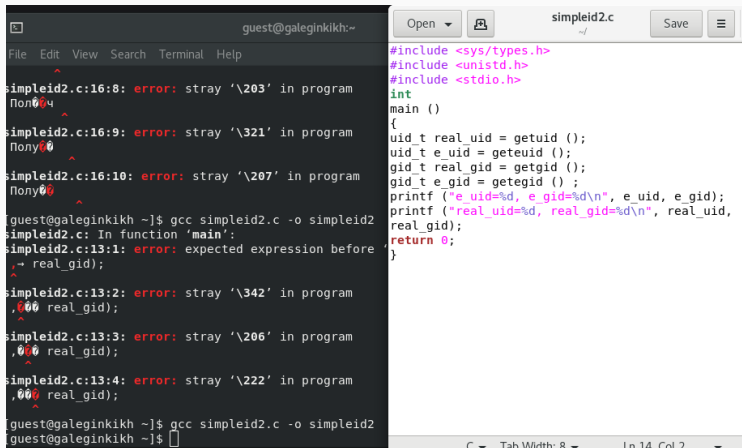
Рис. 3: программа simpleid.c

5. Выполнила программу simpleid. Выполнила системную программу id. Сравнила полученные результаты. Они схожи.

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

Рис. 4: simpleid.c as id

6. Усложнила программу, добавив вывод действительных идентификаторов. Получившуюся программу назвала simpleid2.c. Скомпилировала и запустила simpleid2.c.



The screenshot shows a terminal window with a dark background and a code editor window with a light background. The terminal window displays several compilation errors for the file simpleid2.c. The errors are related to stray characters in the program and an expected expression before a semicolon. The code editor window shows the source code of simpleid2.c, which includes headers for sys/types.h, unistd.h, and stdio.h. The code defines main() and uses getuid(), geteuid(), getgid(), and getegid() to retrieve user and group IDs. It then prints the effective and real user and group IDs using printf.

```
guest@galeginkikh:~  
File Edit View Search Terminal Help  
simpleid2.c:16:8: error: stray '\203' in program  
Пол00ч  
simpleid2.c:16:9: error: stray '\321' in program  
Полу00  
simpleid2.c:16:10: error: stray '\207' in program  
Полу00  
[guest@galeginkikh ~]$ gcc simpleid2.c -o simpleid2  
simpleid2.c: In function 'main':  
simpleid2.c:13:1: error: expected expression before  
↪ real_gid);  
simpleid2.c:13:2: error: stray '\342' in program  
,000 real_gid);  
simpleid2.c:13:3: error: stray '\206' in program  
,000 real_gid);  
simpleid2.c:13:4: error: stray '\222' in program  
,000 real_gid);  
[guest@galeginkikh ~]$ gcc simpleid2.c -o simpleid2  
[guest@galeginkikh ~]$
```

```
#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;  
}
```

C ▾ Tab Width: 8 ▾ Ln 14, Col 2 ▾

```
[guest@galeginkikh ~]$ ./simpleid2  
e_uid=1001, e_gid=1001  
real_uid=1001, real_gid=1001  
[guest@galeginkikh ~]$
```

Рис. 6: программа simpleid2.c

7. От имени суперпользователя выполнила команды.

```
[guest@galeginkikh ~]$ su  
Password:  
[root@galeginkikh guest]# chown root:guest /home/guest/simpleid2  
[root@galeginkikh guest]# chmod u+s /home/guest/simpleid2
```

Рис. 7: Изменение прав

8. Выполнила проверку правильности установки новых атрибутов и смены владельца файла simpleid2. Запустила simpleid2 и id. Сравнила результаты.

```
[root@galeginkikh guest]# ls -l simpleid2
-rwsrwxr-x. 1 root guest 18312 Sep 18 21:22 simpleid2
[root@galeginkikh guest]# ./simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@galeginkikh guest]# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfi
ned_t:s0-s0:c0.c1023
[root@galeginkikh guest]#
```

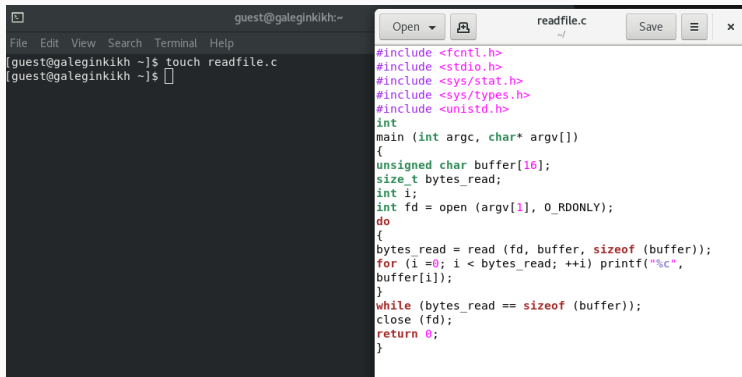
Рис. 8: Проверка правильности установки новых атрибутов

9. Проделала тоже самое относительно SetGID-бита.

```
[root@galeginkikh guest]# sudo chown root:guest /home/guest/simpleid2
[root@galeginkikh guest]# sudo chmod g+s /home/guest/simpleid2
[root@galeginkikh guest]# ls -l simpleid2
-rwxrwsr-x. 1 root guest 18312 Sep 18 21:22 simpleid2
[root@galeginkikh guest]# ./simpleid2
e_uid=0, e_gid=1001
real_uid=0, real_gid=0
[root@galeginkikh guest]# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfi
ned_t:s0-s0:c0.c1023
[root@galeginkikh guest]#
```

Рис. 9: SetGID-бита

10. Создала программу readfile.c. Откомпилировала ее.



The screenshot shows a terminal window on the left and a code editor on the right. The terminal window has a title bar 'guest@galeginkikh:~' and a menu bar 'File Edit View Search Terminal Help'. It shows the command 'touch readfile.c' being executed. The code editor window has a title bar 'readfile.c' and a menu bar 'Open Save'. It displays the C code for the 'readfile.c' program.

```
guest@galeginkikh:~  
File Edit View Search Terminal Help  
[guest@galeginkikh ~]$ touch readfile.c  
[guest@galeginkikh ~]$  
  
#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;  
}
```

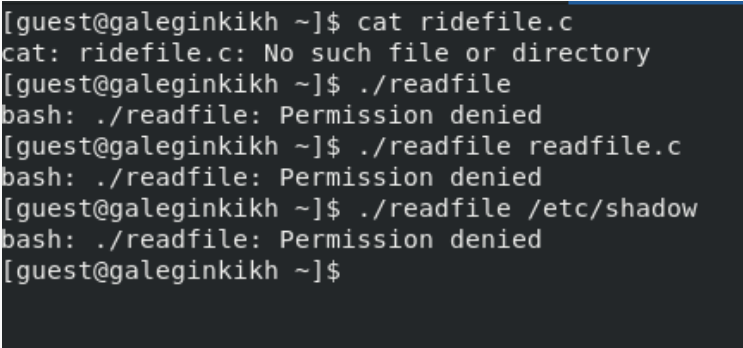
Рис. 10: readfile.c

11. . Сменила владельца у файла readfile.c и изменила права так, чтобы только суперпользователь (root) мог прочитать его, а guest не мог. Проверила, что пользователь guest не может прочитать файл readfile.c. Сменила у программы readfile владельца и установила SetU'D-бит.

```
[root@galeginkikh guest]# sudo chown root:guest readfile
[root@galeginkikh guest]# chmod 700 redfile
chmod: cannot access 'redfile': No such file or directory
[root@galeginkikh guest]# chmod 700 readfile
[root@galeginkikh guest]# chmod -r readfile.c
[root@galeginkikh guest]# chmod u+c readfile.c
chmod: invalid mode: 'u+c'
Try 'chmod --help' for more information.
[root@galeginkikh guest]# chmod u+s readfile.c
```

Рис. 11: Смена владельца файла readfile.c

12. Проверила, может ли программа readfile прочитать файл readfile.c. Проверила, может ли программа readfile прочитать файл /etc/shadow.



```
[guest@galeginkikh ~]$ cat ridefile.c
cat: ridefile.c: No such file or directory
[guest@galeginkikh ~]$ ./readfile
bash: ./readfile: Permission denied
[guest@galeginkikh ~]$ ./readfile readfile.c
bash: ./readfile: Permission denied
[guest@galeginkikh ~]$ ./readfile /etc/shadow
bash: ./readfile: Permission denied
[guest@galeginkikh ~]$
```

Рис. 12: Проверка readfile.c

13. От имени суперпользователя все выполняется.

```
[root@galeginkikh guest]# 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]);
    }
```

Рис. 13: cat readfile.c

```
[root@galeginkikh guest]# ./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]);
    }
```

Рис. 14: Запуск readfile.c

[illegible]

Рис. 15: Запуск shadow

14. Выяснила, установлен ли атрибут Sticky на директории /tmp. 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”.

```
[guest@galeginkikh ~]$ ls -l / | grep tmp
drwxrwxrwt. 11 root root 4096 Sep 18 21:37 tmp
[guest@galeginkikh ~]$ echo "test" > /tmp/file01.txt
[guest@galeginkikh ~]$
[guest@galeginkikh ~]$ ls -l /tmp/file01.txt
-rw-rw-r--. 1 guest guest 5 Sep 18 21:37 /tmp/file01.txt
[guest@galeginkikh ~]$ chmod o+rw /tmp/file01.txt
[guest@galeginkikh ~]$ ls -l /tmp/file01.txt
-rw-rw-rw-. 1 guest guest 5 Sep 18 21:37 /tmp/file01.txt
[guest@galeginkikh ~]$
```

Рис. 16: file01.txt

15. От пользователя guest2 попробовала дозаписать в файл /tmp/file01.txt слово test2. Проверила содержимое файла.

```
[guest@galeginkikh ~]$ su guest2
Password:
[guest2@galeginkikh guest]$ echo "test2" > /tmp/file01.txt
[guest2@galeginkikh guest]$ cat /tmp/file01.txt
test2
[guest2@galeginkikh guest]$
```

Рис. 17: Изменение file01.txt

16. От пользователя guest2 попробовала дозаписать в файл /tmp/file01.txt слово test3. Проверила содержимое файла. А вот удалить файл не удалось.

```
[guest2@galeginkikh guest]$ echo "test3" > /tmp/file01.txt
[guest2@galeginkikh guest]$ cat /tmp/file01.txt
test3
[guest2@galeginkikh guest]$ rm /tmp/file01.txt
rm: cannot remove '/tmp/file01.txt': No such file or directory
[guest2@galeginkikh guest]$
```

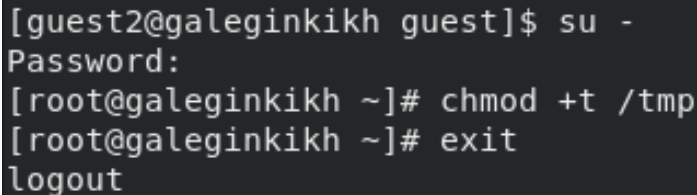
Рис. 18: Удаление file01.txt

17. Повысила прова до суперпользователя. Сняла атрибут t. Покинула режим суперпользователя. Проверила отсутствие атрибута t. Повторила предыдущие шаги.

```
[guest2@galeginkikh guest]$ rm /tmp/file0l.txt
rm: cannot remove '/tmp/file0l.txt': No such file or directory
[guest2@galeginkikh guest]$ su -
Password:
[root@galeginkikh ~]# chmod -t /tmp
[root@galeginkikh ~]# exit
logout
[guest2@galeginkikh guest]$ ls -l / | grep tmp
drwxrwxrwx. 11 root root 4096 Sep 18 21:41 tmp
[guest2@galeginkikh guest]$ rm /tmp/file0l.txt
rm: cannot remove '/tmp/file0l.txt': No such file or directory
[guest2@galeginkikh guest]$
```

Рис. 19: Атрибут -t

18. Вернула атрибут t.



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

Рис. 20: Атрибут +t

Вывод

Изучила механизмы 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.