

Лабораторная работа №5

Лукьянова Ирина Владимировна, НФИбд-02-19

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
2.1	Исследование Sticky-бита	11
3	Выводы	15
4	Список литературы	16

List of Figures

2.1	Установка gcc	6
2.2	Отключение системы запретов	6
2.3	Компиляторы C и C++	7
2.4	Программа simpleid.c	7
2.5	Вход и создание программы	7
2.6	Выполнение программ	7
2.7	Программа simpleid2.c	8
2.8	Компилируем и запускаем simpleid2.c	8
2.9	Выполнение команд	8
2.10	Выполняем проверку	9
2.11	Повтор операций для SetGID-бита	9
2.12	Создаем программу readfile.c	9
2.13	Меняем владельца и права файла	10
2.14	Меняем владельца	10
2.15	Проверка	11
2.16	Проверка	11
2.17	Работа с атрибутами	12
2.18	Проверка	12
2.19	Пробуем прочитатъ файл	12
2.20	Работа с файлом	12
2.21	Попытка удалить файл	13
2.22	Работа с атрибутами	13
2.23	Проверка	13
2.24	Повтор команд	13
2.25	Попытка удалить файл №2	14
2.26	Работа с атрибутами	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.¹

¹Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов.

2 Выполнение лабораторной работы

От имени суперпользователя устанавливаем gcc командой `yum install gcc`(рис. 2.1)

```
[irina@irina ~]$ su
Password:
[root@irina irina]# yum install gcc
Rocky Linux 9 - BaseOS                    5.9 kB/s | 3.6 kB      00:00
Rocky Linux 9 - BaseOS                    370 kB/s | 1.7 MB      00:04
Rocky Linux 9 - AppStream                  5.9 kB/s | 3.6 kB      00:00
Rocky Linux 9 - AppStream                  492 kB/s | 6.0 MB      00:12
Rocky Linux 9 - Extras                     4.6 kB/s | 2.9 kB      00:00
Package gcc-11.2.1-9.4.el9.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@irina irina]#
```

Figure 2.1: Установка gcc

Отключаем систему запретов до очередной перезагрузки системы командой `setenforce 0`. После этого команда `getenforce` выводит `Permissive`.(рис. 2.2)

```
[root@irina irina]# setenforce 0
[root@irina irina]# getenforce
Permissive
[root@irina irina]#
```

Figure 2.2: Отключение системы запретов

Компилятор языка C называется gcc. Компилятор языка C++ называется g++ и запускается с параметрами почти так же, как gcc. Проверяем это следующими командами:(рис. 2.3)

```
[root@irina irinal# whereis gcc
gcc: /usr/bin/gcc /usr/lib/gcc /usr/libexec/gcc /usr/share/man/man1/gcc.1.gz /usr/share/info/gcc.info.gz
[root@irina irinal# whereis g++
g++: /usr/bin/g++ /usr/share/man/man1/g++.1.gz
[root@irina irinal#
```

Figure 2.3: Компиляторы C и C++

Входим в систему от имени пользователя guest и создаем программу simpleid.c:(рис. 2.4), (рис. 2.5)

```
guest@irina:~
#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;
}
```

Figure 2.4: Программа simpleid.c

```
[root@irina irinal# su guest
[guest@irina irinal]$ su - guest
Password:
[guest@irina ~]$ vi simpleid.c
[guest@irina ~]$
```

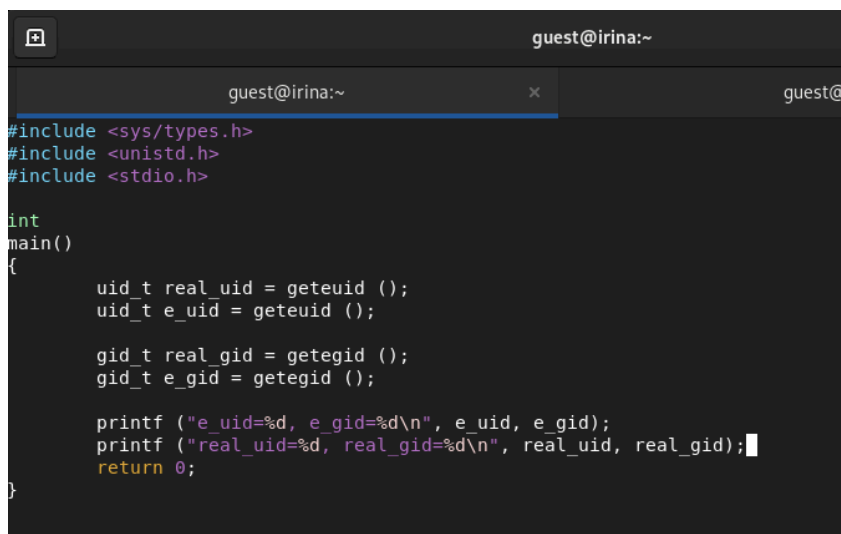
Figure 2.5: Вход и создание программы

Компилируем и выполняем программу simpleid. После выполняем программу id.(рис. 2.6).

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

Figure 2.6: Выполнение программ

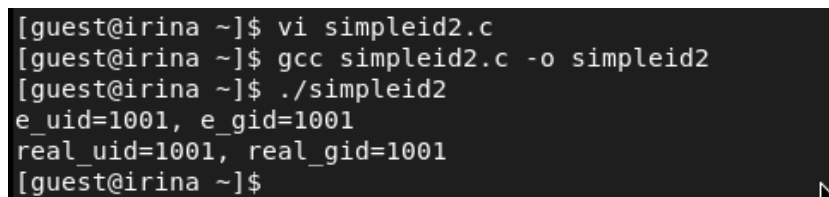
Усложняем программу simpleid2.c, добавив вывод действительных идентификаторов.(рис. 2.7).



```
guest@irina:~  
guest@irina:~  
#include <sys/types.h>  
#include <unistd.h>  
#include <stdio.h>  
  
int  
main()  
{  
    uid_t real_uid = geteuid ();  
    uid_t e_uid = geteuid ();  
  
    gid_t real_gid = getegid ();  
    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 2.7: Программа simpleid2.c

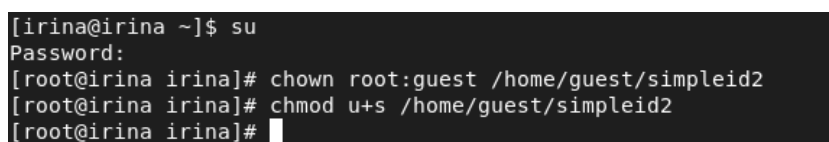
Компилируем и запускаем simpleid2.c. (рис. 2.8)



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

Figure 2.8: Компилируем и запускаем simpleid2.c

От имени суперпользователя выполняем команды:(рис. 2.9)



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

Figure 2.9: Выполнение команд

Используйте `sudo` или повысьте временно свои права с помощью `su`. Поясните, что делают эти команды. Команда `sudo` позволяет пользователям выполнять

указанные программы с административными привилегиями без ввода пароля суперпользователя root.

Выполняем проверку правильности установки новых атрибутов и смены владельца файла simpleid2. После запускаем simpleid2 и id(рис. 2.10)

```
[guest@irina ~]$ ls -l simpleid2
-rwsr-xr-x. 1 root guest 25904 Sep 28 19:15 simpleid2
[guest@irina ~]$ ./simpleid2
e_uid=0, e_gid=1001
real_uid=0, real_gid=1001
[guest@irina ~]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@irina ~]$
```

Figure 2.10: Выполняем проверку

Продельываем тоже самое относительно SetGID-бита.(рис. 2.11)

```
[root@irina irina]# chown root:guest /home/guest/simpleid2
[root@irina irina]# chmod g+s /home/guest/simpleid2
[root@irina irina]# su - guest
[guest@irina ~]$ ls -l simpleid2
ls: cannot access 'simpleid2': No such file or directory
[guest@irina ~]$ ls -l simpleid2
-rwxr-sr-x. 1 root guest 25904 Sep 28 19:15 simpleid2
[guest@irina ~]$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
[guest@irina ~]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@irina ~]$
```

Figure 2.11: Повтор операций для SetGID-бита

Создаем программу readfile.c(рис. 2.12)

```
guest@irina:~
#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 2.12: Создаем программу readfile.c

Компилируем эту программу(рис. ??)

```
[guest@irina ~]$ vi readfile.c
[guest@irina ~]$ gcc readfile.c -o readfile
[guest@irina ~]$
```

Сменим владельца у файла readfile.c и изменим права так, чтобы только суперпользователь мог прочитать его, а guest не мог.(рис. 2.13)

```
[root@irina irina]# chown root:guest /home/guest/readfile.c
[root@irina irina]# chmod 700 /home/guest/readfile.c
[root@irina irina]#
```

Figure 2.13: Меняем владельца и права файла

Проверяем, что пользователь guest не может прочитать файл readfile.c.(рис.

```
[root@irina irina]# su - guest
[guest@irina ~]$ ls -l readfile
-rwx-----. 1 root guest 25952 Sep 28 23:10 readfile
[guest@irina ~]$ cat readfile
cat: readfile: Permission denied
[guest@irina ~]$
```

??)

Сменим у программы readfile владельца и установим SetU'D-бит.(рис. 2.14)

```
[root@irina irina]# chown root:guest /home/guest/readfile
[root@irina irina]# chmod u+s /home/guest/readfile
[root@irina irina]#
```

Figure 2.14: Меняем владельца

Проверяем, может ли программа readfile прочитать файл readfile.c (рис. 2.15)


```
[guest@irina ~]$ ls -l / | grep tmp
drwxrwxrwt. 13 root root 4096 Sep 28 23:23 tmp
[guest@irina ~]$ echo "test" > /tmp/file01.txt
[guest@irina ~]$ ls -l /tmp/file01.txt
-rw-r--r--. 1 guest guest 5 Sep 28 23:42 /tmp/file01.txt
[guest@irina ~]$ chmod o+rw /tmp/file01.txt
```

Figure 2.17: Работа с атрибутами

```
[guest@irina ~]$ ls -l /tmp/file01.txt
-rw-rw-rw-. 1 guest guest 5 Sep 28 23:42 /tmp/file01.txt
[guest@irina ~]$
```

Figure 2.18: Проверка

От пользователя guest2 (не являющегося владельцем) пробуем прочитать файл /tmp/file01.txt. Далее от пользователя guest2 пробуем дозаписать в файл /tmp/file01.txt слово test2 и проверяем содержимое файла (рис. 2.19)

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

Figure 2.19: Пробуем прочитать файл

От пользователя guest2 пробуем записать в файл /tmp/file01.txt слово test3, стерев при этом всю имеющуюся в файле информацию и проверяем содержимое файла (рис. 2.20)

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

Figure 2.20: Работа с файлом

От пользователя guest2 пробуем удалить файл /tmp/file01.txt (рис. 2.21)

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

Figure 2.21: Попытка удалить файл

Повышаем свои права до суперпользователя и выполняем после этого команду, снимающую атрибут t (Sticky-бит) с директории /tmp:(рис. 2.22)

```
[guest2@irina ~]$ su -
Password:
[root@irina ~]# chmod -t /tmp
[root@irina ~]#
```

Figure 2.22: Работа с атрибутами

Покидаем режим суперпользователя командой exit и от пользователя guest2 проверяем, что атрибута t у директории /tmp нет:(рис. 2.23)

```
[root@irina ~]# chmod -t /tmp
[root@irina ~]# exit
logout
[guest2@irina ~]$ ls -l / | grep tmp
drwxrwxrwx. 13 root root 4096 Sep 28 23:54 tmp
[guest2@irina ~]$
```

Figure 2.23: Проверка

Повторяем предыдущие шаги.(рис. 2.24)

```
[guest2@irina ~]$ ls -l / | grep tmp
drwxrwxrwx. 13 root root 4096 Sep 28 23:54 tmp
[guest2@irina ~]$ echo "test4" > /tmp/file01.txt
[guest2@irina ~]$ cat /tmp/file01.txt
test4
[guest2@irina ~]$ rm /tmp/file01.txt
[guest2@irina ~]$ cat /tmp/file01.txt
cat: /tmp/file01.txt: No such file or directory
[guest2@irina ~]$
```

Figure 2.24: Повтор команд

Удалось ли вам удалить файл от имени пользователя, не являющегося его владельцем? Да, удалось. (рис. 2.25)

```
[guest2@irina ~]$ ls -l / | grep tmp
drwxrwxrwx. 13 root root 4096 Sep 28 23:54 tmp
[guest2@irina ~]$ echo "test4" > /tmp/file01.txt
[guest2@irina ~]$ cat /tmp/file01.txt
test4
[guest2@irina ~]$ rm /tmp/file01.txt
[guest2@irina ~]$ cat /tmp/file01.txt
cat: /tmp/file01.txt: No such file or directory
[guest2@irina ~]$
```

Figure 2.25: Попытка удалить файл №2

Повышаем свои права до суперпользователя и возвращаем атрибут `t` на директорию `/tmp`: (рис. 2.26)

```
[guest2@irina ~]$ su -
Password:
[root@irina ~]# chmod +t /tmp
[root@irina ~]# exit
logout
[guest2@irina ~]$ ls -l / | grep tmp
drwxrwxrwt. 15 root root 4096 Sep 28 23:56 tmp
[guest2@irina ~]$
```

Figure 2.26: Работа с атрибутами

3 Выводы

В ходе выполнения данной лабораторной работы я изучила механизмы изменения идентификаторов и 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.

4 Список литературы

1. Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов. / Кулябов Д. С., Королькова А. В., Геворкян М. Н. - Москва: - 7 с.
2. Руководство по оформлению Markdown.