

Информационная безопасность. Отчет по лабораторной работе №5

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

Терентьев Егор Дмитриевич 1032192875

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	13
4	Список литературы	14

List of Figures

2.1	touch_simpleid	6
2.2	listing_simpleid	6
2.3	simpleid_id	7
2.4	listing_simpleid2	7
2.5	launch_simpleid	7
2.6	chown_chmod	8
2.7	ls_simpleid2	8
2.8	setgid	8
2.9	readfile	9
2.10	chown_chmod_readfile	9
2.11	ls_readfile	10
2.12	readfile_readfile	10
2.13	ls_chmod_echo	11
2.14	guest2_comm	11
2.15	without_stickybit	12
2.16	return_stickybit	12

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.

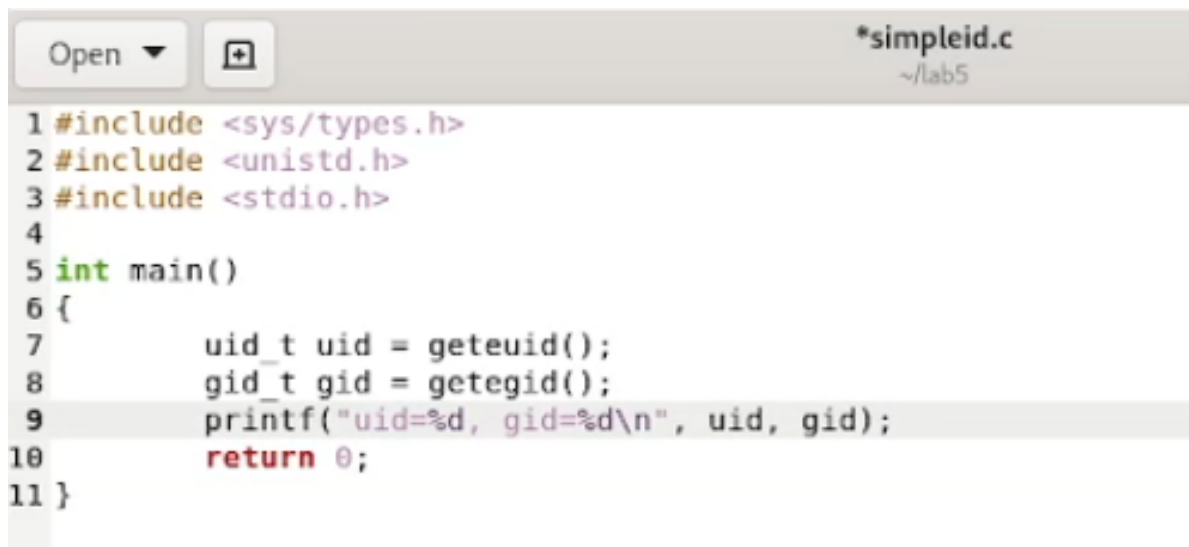
2 Выполнение лабораторной работы

Вхожу в систему от guest и создаю программу simpleid.c fig. 2.1.

```
[guest@edterentjev ~]$ mkdir lab5
[guest@edterentjev ~]$ cd lab5
[guest@edterentjev lab5]$ touch simpleid.c
[guest@edterentjev lab5]$ gedit simpleid.c
```

Figure 2.1: touch_simpleid

Листинг программы simpleid fig. 2.2.



```
1 #include <sys/types.h>
2 #include <unistd.h>
3 #include <stdio.h>
4
5 int main()
6 {
7     uid_t uid = geteuid();
8     gid_t gid = getegid();
9     printf("uid=%d, gid=%d\n", uid, gid);
10     return 0;
11 }
```

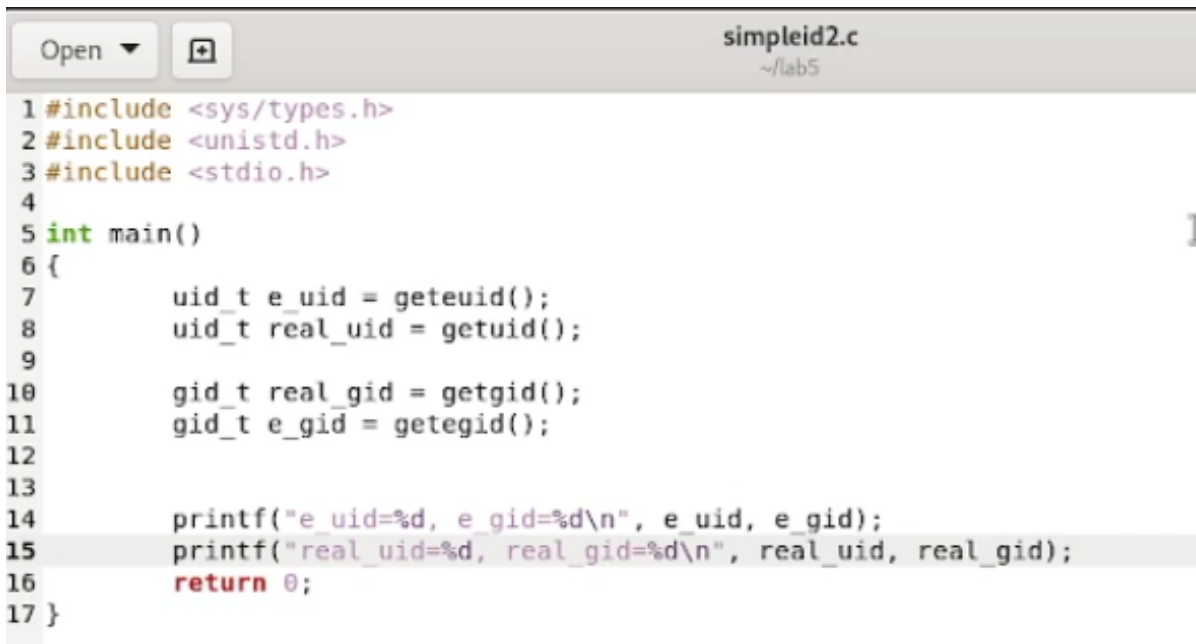
Figure 2.2: listing_simpleid

Скомпилируйте программу и убедитесь, что файл программы создан: `gcc simpleid.c -o simpleid`. Выполните программу simpleid: `./simpleid`. Выполните системную программу id: fig. 2.3.

```
[guest@edterentjev lab5]$ ./simpleid
uid=1002, gid=1002
[guest@edterentjev lab5]$ id
uid=1002(guest) gid=1002(guest) groups=1002(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Figure 2.3: simpleid_id

Усложните программу, добавив вывод действительных идентификаторов и назовите ее simpleid2 fig. 2.4



```
simpleid2.c
~/lab5

1 #include <sys/types.h>
2 #include <unistd.h>
3 #include <stdio.h>
4
5 int main()
6 {
7     uid_t e_uid = geteuid();
8     uid_t real_uid = getuid();
9
10    gid_t real_gid = getgid();
11    gid_t e_gid = getegid();
12
13
14    printf("e_uid=%d, e_gid=%d\n", e_uid, e_gid);
15    printf("real_uid=%d, real_gid=%d\n", real_uid, real_gid);
16    return 0;
17 }
```

Figure 2.4: listing_simpleid2

Скомпилируйте и запустите simpleid2.c fig. 2.5

```
[guest@edterentjev lab5]$ gedit simpleid2.c
[guest@edterentjev lab5]$ gcc simpleid2.c -o simpleid2
[guest@edterentjev lab5]$ ./simpleid2
e_uid=1002, e_gid=1002
real_uid=1002, real_gid=1002
```

Figure 2.5: launch_simpleid

От имени суперпользователя выполните команды: `chown root:guest /home/guest/simpleid2 chmod u+s /home/guest/simpleid2` fig. 2.6.

```
[guest@edterentjev lab5]$ su
Password:
[root@edterentjev lab5]# chown root:guest /home/guest/simpleid2
chown: cannot access '/home/guest/simpleid2': No such file or directory
[root@edterentjev lab5]# chown root:guest /home/guest/lab5/simpleid2
[root@edterentjev lab5]# chmod u+s /home/guest/lab5/simpleid2
```

Figure 2.6: chown_chmod

Выполните проверку правильности установки новых атрибутов и смены владельца файла simpleid2: `ls -l simpleid2` Запустите simpleid2 и `id: ./simpleid2` id fig. 2.7.

```
[root@edterentjev lab5]# ls -l simpleid2
-rwsrwxr-x. 1 root guest 26008 Oct  4 15:20 simpleid2
[root@edterentjev lab5]# ./simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@edterentjev lab5]# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Figure 2.7: ls_simpleid2

Прodelайте тоже самое относительно SetGID-бита fig. 2.8

```
[guest@edterentjev lab5]$ su
Password:
[root@edterentjev lab5]# chmod g+s /home/guest/lab5/simpleid2
[root@edterentjev lab5]# su - guest
[guest@edterentjev ~]$ cd lab5
[guest@edterentjev lab5]$ ls -l simpleid2
-rwsrwsr-x. 1 root guest 26008 Oct  4 15:20 simpleid2
[guest@edterentjev lab5]$ ./simpleid2
e_uid=0, e_gid=1002
real_uid=1002, real_gid=1002
[guest@edterentjev lab5]$ id
uid=1002(guest) gid=1002(guest) groups=1002(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@edterentjev lab5]$
```

Figure 2.8: setgid

Создайте программу readfile.c fig. 2.9.


```
guest@edterentjev:~/lab5
[guest@edterentjev lab5]$ 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@edterentjev lab5]$
```

Figure 2.9: readfile

Смените владельца у файла readfile.c и измените права так, чтобы только суперпользователь (root) мог прочитать его, а guest не мог fig. 2.10.

```
[root@edterentjev lab5]# chown root:root readfile
[root@edterentjev lab5]# ls -l
bash: ls -l: command not found...
[root@edterentjev lab5]# ls -l
total 96
-rwxrwxr-x. 1 root  root  25952 Oct  4 15:50 readfile
-rw-r--r--. 1 guest guest   408 Oct  4 15:49 readfile.c
-rwxrwxr-x. 1 guest guest 25904 Oct  4 15:15 simpleid
-rwsrwsr-x. 1 root  guest 26008 Oct  4 15:20 simpleid2
-rw-rw-r--. 1 guest guest   309 Oct  4 15:19 simpleid2.c
-rw-rw-r--. 1 guest guest   176 Oct  4 15:15 simpleid.c
[root@edterentjev lab5]# chmod 000 readfile
```

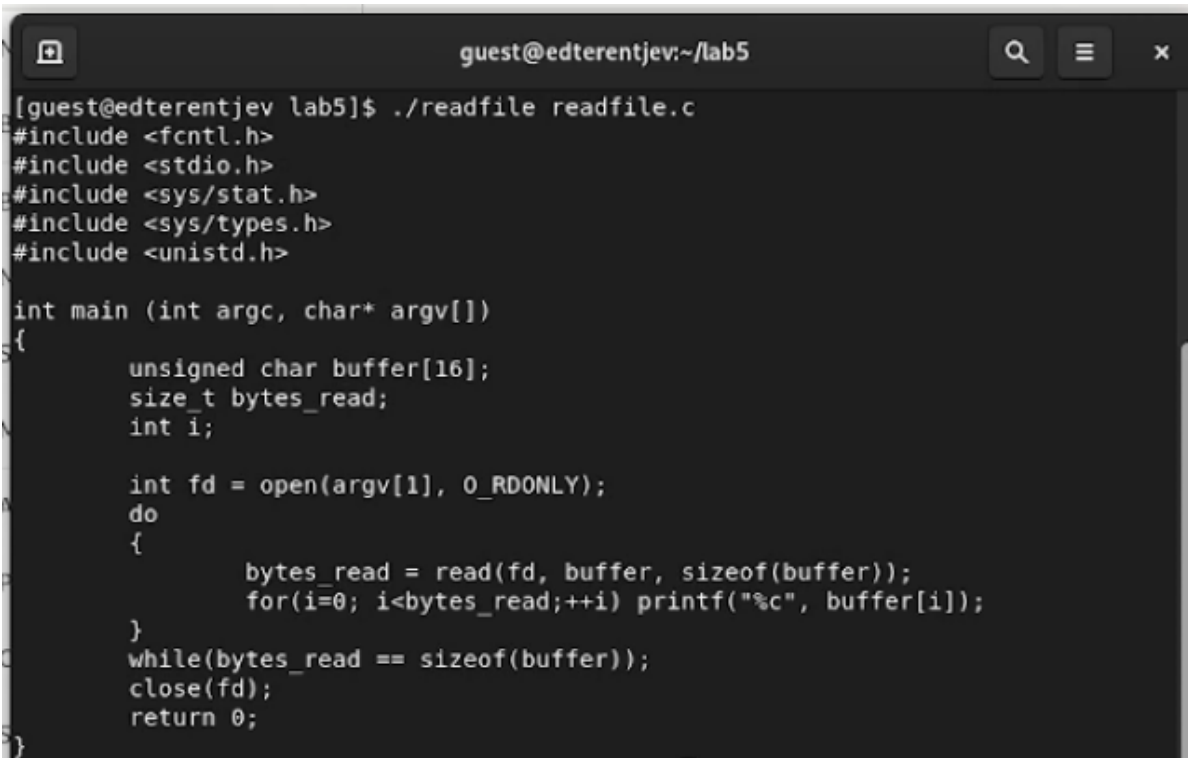
Figure 2.10: chown_chmod_readfile

Проверьте, что пользователь guest не может прочитать файл readfile.c. Смените у программы readfile владельца и установите SetU'D-бит. fig. 2.11.

```
[guest@edterentjev lab5]$ su
Password:
[root@edterentjev lab5]# chown guest:guest readfile
[root@edterentjev lab5]# chmod 700
chmod: missing operand after '700'
Try 'chmod --help' for more information.
[root@edterentjev lab5]# chmod 700 readfile
[root@edterentjev lab5]# su - guest
[guest@edterentjev ~]$ cd lab5
```

Figure 2.11: ls_readfile

Проверьте, может ли программа readfile прочитать файл readfile.c? fig. 2.12



```
guest@edterentjev:~/lab5
[guest@edterentjev 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 2.12: readfile_readfile

Выясните, установлен ли атрибут Sticky на директории /tmp От имени пользо-

вателя guest создайте файл file01.txt в директории /tmp со словом test: Просмотрите атрибуты у только что созданного файла и разрешите чтение и запись для категории пользователей «все остальные» fig. 2.13

```
[guest@edterentjev lab5]$ ls -l / | grep tmp
drwxrwxrwt. 16 root root 4096 Oct  4 15:54 tmp
[guest@edterentjev lab5]$ cd ..
[guest@edterentjev ~]$ echo "test" > /tmp/file01.txt
[guest@edterentjev ~]$ ls -l /tmp/file01.txt
-rw-r--r--. 1 guest guest 5 Oct  4 15:55 /tmp/file01.txt
[guest@edterentjev ~]$ chmod o+rw /tmp/file01.txt
[guest@edterentjev ~]$ ls -l /tmp/file01.txt
-rw-r--rw-. 1 guest guest 5 Oct  4 15:55 /tmp/file01.txt
[guest@edterentjev ~]$
```

Figure 2.13: ls_chmod_echo

От пользователя guest2 (не являющегося владельцем) попробуйте прочитать файл /tmp/file01.txt От пользователя guest2 попробуйте дозаписать в файл /tmp/file01.txt слово test2 Проверьте содержимое файла От пользователя guest2 попробуйте записать в файл /tmp/file01.txt Проверьте содержимое файла От пользователя guest2 попробуйте удалить файл /tmp/file01.txt fig. 2.14

```
guest@edterentjev:~ x guest2@edterentjev:~ x
[guest@edterentjev lab5]$ su - guest2
Password:
[guest2@edterentjev ~]$ cat /tmp/file01.txt
test
[guest2@edterentjev ~]$ echo "test2" >> /tmp/file01.txt
-bash: /tmp/file01.txt: Permission denied
[guest2@edterentjev ~]$ cat /tmp/file01.txt
test
[guest2@edterentjev ~]$ echo "test3" > /tmp/file01.txt
-bash: /tmp/file01.txt: Permission denied
[guest2@edterentjev ~]$ cat /tmp/file01.txt
test
[guest2@edterentjev ~]$ rm /tmp/file01.txt
rm: remove write-protected regular file '/tmp/file01.txt'? y
rm: cannot remove '/tmp/file01.txt': Operation not permitted
[guest2@edterentjev ~]$
```

Figure 2.14: guest2_comm

Повысьте свои права до суперпользователя и выполните после этого команду, снимающую атрибут t (Sticky-бит) с директории /tmp. От пользователя guest2 проверьте, что атрибута t у директории /tmp нет: Повторите предыдущие шаги fig. 2.15

```
[guest2@edterentjev ~]$ su -  
Password:  
[root@edterentjev ~]# chmod -t /tmp  
[root@edterentjev ~]# su - guest2  
[guest2@edterentjev ~]$ ls -l / | grep tmp  
drwxrwxrwx. 17 root root 4096 Oct  4 15:59 tmp  
[guest2@edterentjev ~]$ echo "test2" >> /tmp/file01.txt  
-bash: /tmp/file01.txt: Permission denied  
[guest2@edterentjev ~]$ echo "test2" > /tmp/file01.txt  
-bash: /tmp/file01.txt: Permission denied  
[guest2@edterentjev ~]$ rm /tmp/file01.txt  
rm: remove write-protected regular file '/tmp/file01.txt'? y
```

Figure 2.15: without_stickybit

Повысьте свои права до суперпользователя и верните атрибут t на директорию /tmp fig. 2.16

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

Figure 2.16: return_stickybit

3 Выводы

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

4 Список литературы

1. Методические материалы курса