

# **Лабораторная работа №5**

**Дисциплина: Основы информационной безопасности**

Жибицкая Евгения Дмитриевна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
<b>3</b>	<b>Выводы</b>	<b>13</b>
	<b>Список литературы</b>	<b>14</b>

## Список иллюстраций

2.1	Проверка gcc . . . . .	6
2.2	Создание файла simpleid.c . . . . .	7
2.3	Запуск файла . . . . .	7
2.4	Файл simpleid2.c . . . . .	8
2.5	Работа с правами . . . . .	8
2.6	Исполнение simpleid2.c . . . . .	8
2.7	Файл readfile . . . . .	9
2.8	Работа с правами . . . . .	10
2.9	Запуск файла . . . . .	10
2.10	Sticky-бит . . . . .	10
2.11	Чтение от guest2 . . . . .	11
2.12	Попытка удаления . . . . .	11
2.13	Удаление файла без sticky-бита . . . . .	11
2.14	Возвращение sticky-бит . . . . .	12

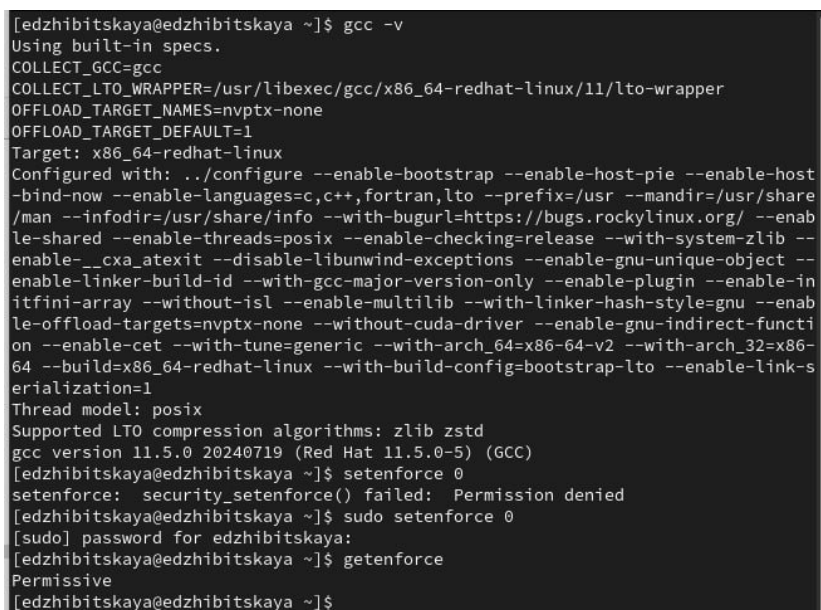
## **Список таблиц**

# 1 Цель работы

Продолжение работы на ОС Rocky. Изучение механизмов изменения идентификаторов, применения SetUID- и Sticky-битов и рассмотрение работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

## 2 Выполнение лабораторной работы

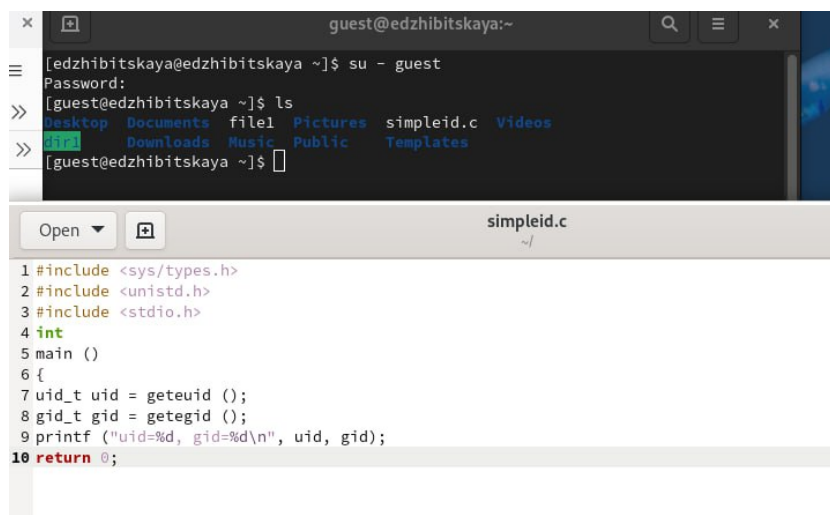
Перед началом выполнения проверим наличие у нас gcc, а также установим setenforce на 0 (рис. 2.1).



```
[edzhibitskaya@edzhibitskaya ~]$ gcc -v
Using built-in specs.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/usr/libexec/gcc/x86_64-redhat-linux/11/lto-wrapper
OFFLOAD_TARGET_NAMES=nvptx-none
OFFLOAD_TARGET_DEFAULT=1
Target: x86_64-redhat-linux
Configured with: ../configure --enable-bootstrap --enable-host-pie --enable-host
-bind-now --enable-languages=c,c++,fortran,lto --prefix=/usr --mandir=/usr/share
/man --infodir=/usr/share/info --with-bugurl=https://bugs.rockylinux.org/ --enab
le-shared --enable-threads=posix --enable-checking=release --with-system-zlib --
enable-__cxa_atexit --disable-libunwind-exceptions --enable-gnu-unique-object --
enable-linker-build-id --with-gcc-major-version-only --enable-plugin --enable-in
itfini-array --without-isl --enable-multilib --with-linker-hash-style=gnu --enab
le-offload-targets=nvptx-none --without-cuda-driver --enable-gnu-indirect-functi
on --enable-cet --with-tune=generic --with-arch_64=x86-64-v2 --with-arch_32=x86-
64 --build=x86_64-redhat-linux --with-build-config=bootstrap-lto --enable-link-s
erialization=1
Thread model: posix
Supported LTO compression algorithms: zlib zstd
gcc version 11.5.0 20240719 (Red Hat 11.5.0-5) (GCC)
[edzhibitskaya@edzhibitskaya ~]$ setenforce 0
setenforce: security_setenforce() failed: Permission denied
[edzhibitskaya@edzhibitskaya ~]$ sudo setenforce 0
[sudo] password for edzhibitskaya:
[edzhibitskaya@edzhibitskaya ~]$ getenforce
Permissive
[edzhibitskaya@edzhibitskaya ~]$
```

Рис. 2.1: Проверка gcc

Далее приступим к работе. От имени пользователя создаем файл и вставляем туда код по получению информации о пользователе(рис. 2.2).



```
guest@edzhbitskaya:~  
[edzhbitskaya@edzhbitskaya ~]$ su - guest  
Password:  
[guest@edzhbitskaya ~]$ ls  
Desktop Documents file1 Pictures simpleid.c Videos  
Downloads Music Public Templates  
[guest@edzhbitskaya ~]$  
  
simpleid.c  
~/  
1 #include <sys/types.h>  
2 #include <unistd.h>  
3 #include <stdio.h>  
4 int  
5 main ()  
6 {  
7     uid_t uid = geteuid ();  
8     gid_t gid = getegid ();  
9     printf ("uid=%d, gid=%d\n", uid, gid);  
10    return 0;
```

Рис. 2.2: Создание файла simpleid.c

Компилируем его, запускаем и смотрим на вывод. Также сравним результат программы с выводом команды id(рис. 2.3).

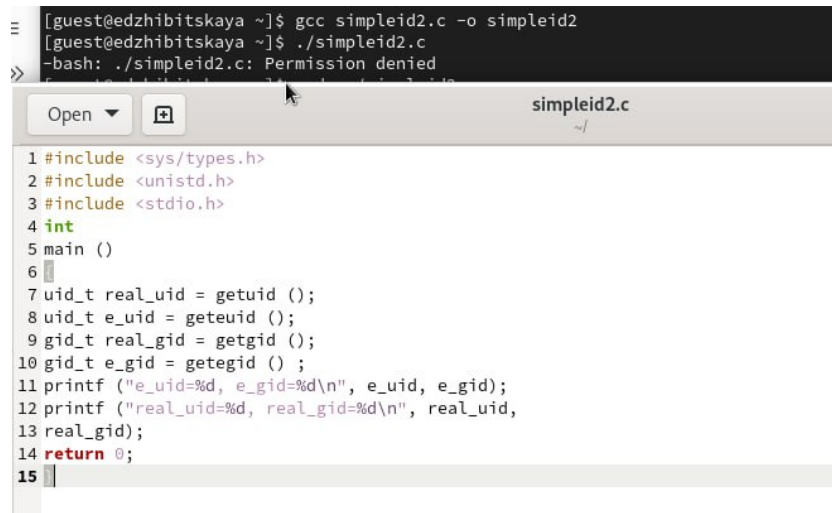


```
guest@edzhbitskaya:~  
[guest@edzhbitskaya ~]$ gcc simpleid.c -o simpleid  
[guest@edzhbitskaya ~]$ ./simpleid  
uid=1001, gid=1001  
[guest@edzhbitskaya ~]$ id  
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023  
[guest@edzhbitskaya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

Рис. 2.3: Запуск файла

Доработаем код, переимнуем файл на simpleid.2 и также скомпилируем и запустим(рис. 2.4).

```
[guest@edzhibitskaya ~]$ gcc simpleid2.c -o simpleid2
[guest@edzhibitskaya ~]$ ./simpleid2.c
-bash: ./simpleid2.c: Permission denied
```



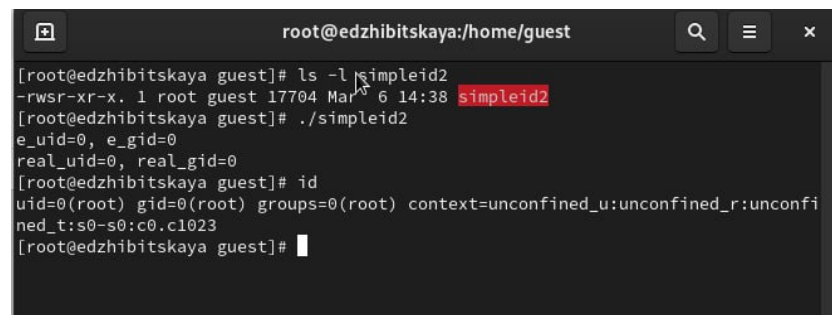
```
1 #include <sys/types.h>
2 #include <unistd.h>
3 #include <stdio.h>
4 int
5 main ()
6 {
7     uid_t real_uid = getuid ();
8     uid_t e_uid = geteuid ();
9     gid_t real_gid = getgid ();
10    gid_t e_gid = getegid ();
11    printf ("e_uid=%d, e_gid=%d\n", e_uid, e_gid);
12    printf ("real_uid=%d,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

Рис. 2.4: Файл simpleid2.c

Далее от администратора добавим его как владельца и повысим права на этот файл(рис. 2.5). Проверим, что все хорошо и запустим его. Сравним с командой id(рис. 2.6).

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

Рис. 2.5: Работа с правами



```
root@edzhibitskaya:/home/guest
[root@edzhibitskaya guest]# ls -l simpleid2
-rwsr-xr-x. 1 root guest 17704 Mar 6 14:38 simpleid2
[root@edzhibitskaya guest]# ./simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@edzhibitskaya guest]# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfi
ned_t:s0-s0:c0.c1023
[root@edzhibitskaya guest]#
```

Рис. 2.6: Исполнение simpleid2.c

Затем создадим файл readfile, вставим код

```
#include <fcntl.h>
```



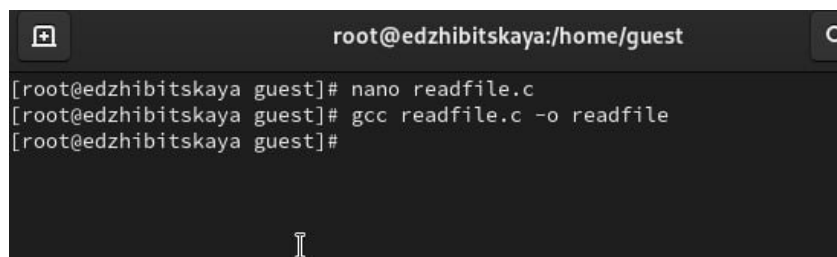
```

#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;
}

```

и скомпилируем его(рис. 2.7).



The screenshot shows a terminal window with the title bar 'root@edzhibitskaya:/home/guest'. The terminal contains the following commands and their outputs:

```

[root@edzhibitskaya guest]# nano readfile.c
[root@edzhibitskaya guest]# gcc readfile.c -o readfile
[root@edzhibitskaya guest]#

```

The cursor is visible at the end of the last line.

Рис. 2.7: Файл readfile

Сменим владельца у файла и повысим на него права(рис. 2.8).

```
root@edzhbitskaya:/home/guest
[root@edzhbitskaya guest]# chown root:guest /home/guest/readfile
[root@edzhbitskaya guest]# chmod u+s /home/guest
guest/ guest2/
[root@edzhbitskaya guest]# chmod u+s /home/guest/readfile
[root@edzhbitskaya guest]#
```

Рис. 2.8: Работа с правами

Перейдем в пользователя guest, сменим у программы readfile владельца и установим SetU'D-бит, попробуем запустить файл - получим отказ в доступе(рис. [-fig. 2.9).

```
password:
[root@edzhbitskaya ~]# sudo chown root:guest /home/guest/readfile
[root@edzhbitskaya ~]# su guest
[guest@edzhbitskaya root]$ ./readfile readfile.c
bash: ./readfile: Permission denied
[guest@edzhbitskaya root]$ ./readfile /etc/shadow
bash: ./readfile: Permission denied
[guest@edzhbitskaya root]$
```

Рис. 2.9: Запуск файла

Перейдем к следующему заданию.

Сначала проверим установлен ли на директорию sticky-бит, запишем в него сообщение. Посмотрим на установленные права, разрешим чтение и запись для всех остальных пользователей (рис. [-fig. 2.10).

```
guest2@edzhbitskaya:/home/guest
[guest@edzhbitskaya ~]$ ls -l / | grep tmp
drwxrwxrwt. 19 root root 4096 Mar  6 15:09 tmp
[guest@edzhbitskaya ~]$ echo "test" > /tmp/file01.txt
[guest@edzhbitskaya ~]$ ls -l /tmp/file01.txt
-rw-r--r--. 1 guest guest 5 Mar  6 15:12 /tmp/file01.txt
[guest@edzhbitskaya ~]$ chmod o+rw /tmp/file01.txt
[guest@edzhbitskaya ~]$ ls -l /tmp/file01.txt
-rw-r--rw-. 1 guest guest 5 Mar  6 15:12 /tmp/file01.txt
```

Рис. 2.10: Sticky-бит

От пользователя guest2 прочитаем файл, попробуем записать туда текст(без-успешно)Попробуем удалить файл - также безуспешно(рис. [-fig. 2.11) и рис. [-

fig. 2.12).

```
[guest@edzhibitskaya ~]$ su guest2
Password:
[guest2@edzhibitskaya guest]$ cat /tmp/file01.txt
test
[guest2@edzhibitskaya guest]$ echo "test2" > /tmp/file01.txt
bash: /tmp/file01.txt: Permission denied
[guest2@edzhibitskaya guest]$ echo "test3" > /tmp/file01.txt
bash: /tmp/file01.txt: Permission denied
[guest2@edzhibitskaya guest]$ cat /tmp/file01.txt
test
[guest2@edzhibitskaya guest]$
```

Рис. 2.11: Чтение от guest2

```
[guest2@edzhibitskaya guest]$ rm /tmp/file01.txt
rm: remove write-protected regular file '/tmp/file01.txt'? y
rm: cannot remove '/tmp/file01.txt': Operation not permitted
[guest2@edzhibitskaya guest]$ su -
```

Рис. 2.12: Попытка удаления

Получив необходимые полномочия, уберем sticky-бит, повторим те же действия и уже удалим файл(рис. [-fig. 2.13).

```
[guest2@edzhibitskaya guest]$ su -
Password:
[root@edzhibitskaya ~]# chmod -t /tmp
[root@edzhibitskaya ~]# exit
logout
[guest2@edzhibitskaya guest]$ ls -l / | grep tmp
drwxrwxrwx. 18 root root 4096 Mar  6 15:18 tmp
[guest2@edzhibitskaya guest]$ echo "test2" > /tmp/file01.txt
bash: /tmp/file01.txt: Permission denied
[guest2@edzhibitskaya guest]$ echo "test3" > /tmp/file01.txt
bash: /tmp/file01.txt: Permission denied
[guest2@edzhibitskaya guest]$ rm /tmp/file01.txt
rm: remove write-protected regular file '/tmp/file01.txt'? y
```

Рис. 2.13: Удаление файла без sticky-бита

Вернем бит и завершим выполнение работырис. [-fig. 2.14).

```
[guest2@edzhibitskaya guest]$ su -  
Password:  
[root@edzhibitskaya ~]# chmod +t /tmp  
[root@edzhibitskaya ~]# exit  
logout  
[guest2@edzhibitskaya guest]$ ls -l / | grep tmp  
drwxrwxrwt. 20 root root 4096 Mar  6 15:19 tmp  
[guest2@edzhibitskaya guest]$
```

Рис. 2.14: Возвращение sticky-бит

## 3 Выводы

В ходе работы были изучены механизмы изменения идентификаторов, применения SetUID- и Sticky-битов а также влияние бита Sticky на запись и удаление файлов.

# **Список литературы**

ТУИС