

# **Лабораторная работа №5**

**Основы информационной безопасности**

Волгин Иван Алексеевич

# Содержание

1	Цель работы	5
2	Теоретическое введение	6
3	Подготовка к выполнению работы	7
4	Выполнение лабораторной работы	8
5	Выводы	13

## Список иллюстраций

3.1	Проверка установки ПО . . . . .	7
4.1	Вход в систему от другого пользователя . . . . .	8
4.2	Создание программы . . . . .	8
4.3	Заполнение программы . . . . .	9
4.4	Компиляция и запуск программы . . . . .	9
4.5	Команда id . . . . .	9
4.6	Компиляция и запуск программы . . . . .	10
4.7	Поменяла владельца программы . . . . .	10
4.8	ls -l . . . . .	10
4.9	Сравнение результатов . . . . .	10
4.10	Создание программы . . . . .	11
4.11	Компиляция программы . . . . .	11
4.12	Смена владельца . . . . .	11
4.13	Проверка . . . . .	12

## **Список таблиц**

# 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 Теоретическое введение

Setuid, Setgid и Sticky Bit - это специальные типы разрешений позволяют задавать расширенные права доступа на файлы или каталоги.

### 3 Подготовка к выполнению работы

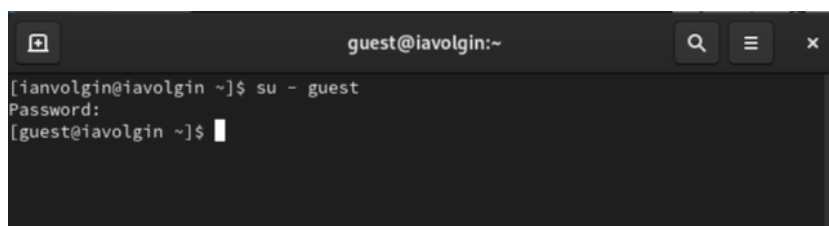
Я проверил, установлен ли у меня gcc командой **yum install gcc** (рис. 3.1). Помимо этого, я отключил систему запретов до очередной перезагрузки системы командой **setenforce 0**. После этого команда **getenforce** выводит **Permissive**.

```
[ianvolgin@iavolgin ~]$ yum install gcc
Error: This command has to be run with superuser privileges (under the root use
r on most systems).
[ianvolgin@iavolgin ~]$ sudo yum install gcc
[sudo] password for ianvolgin:
Last metadata expiration check: 1:59:06 ago on Thu 11 Apr 2024 02:41:35 PM MSK.
Package gcc-11.4.1-2.1.el9.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[ianvolgin@iavolgin ~]$ setenforce 0
setenforce: security_setenforce() failed: Permission denied
[ianvolgin@iavolgin ~]$ sudo setenforce 0
[ianvolgin@iavolgin ~]$ getenforce
Permissive
[ianvolgin@iavolgin ~]$
[ianvolgin@iavolgin ~]$
```

Рис. 3.1: Проверка установки ПО

## 4 Выполнение лабораторной работы

Для начала я захожу в систему от имени пользователя guest (рис. 4.1).



```
guest@iavolgin:~  
[ianvolgin@iavolgin ~]$ su - guest  
Password:  
[guest@iavolgin ~]$
```

Рис. 4.1: Вход в систему от другого пользователя

Далее создаю программу simpleid.c и заполняю ее (рис. 4.2), (рис. 4.3).



```
guest@iavolgin:~  
[ianvolgin@iavolgin ~]$ su - guest  
Password:  
[guest@iavolgin ~]$ pwd  
/home/guest  
[guest@iavolgin ~]$ touch simpleid.c  
[guest@iavolgin ~]$ ls  
dir1 simpleid.c  
[guest@iavolgin ~]$
```

Рис. 4.2: Создание программы



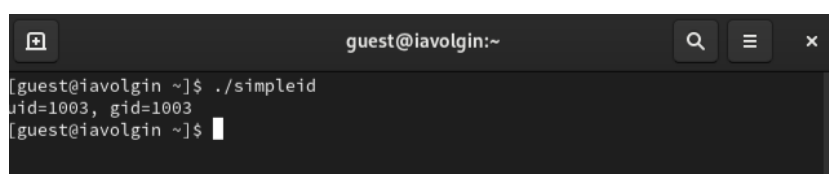


```
GNU nano 5.6.1 simpleid.c
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>

int
main ()
{
    uid_t uid = geteuid ();
    gid_t gid = getegid ();
    printf ("uid=%d, gid=%d\n", uid, gid);
    return 0;
}
```

Рис. 4.3: Заполнение программы

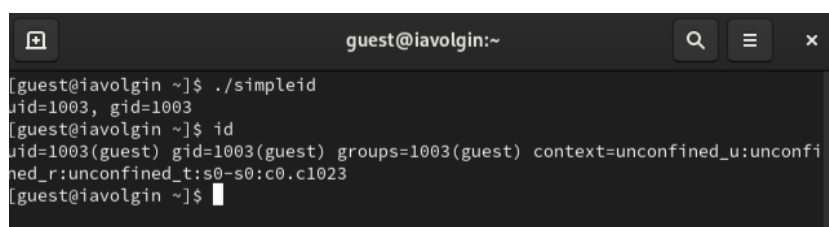
После этого скомпилировал файл через **gcc simpleid.c -o simpleid** и выполнил программу simpleid (рис. 4.4).



```
[guest@iavolgin ~]$ ./simpleid
uid=1003, gid=1003
[guest@iavolgin ~]$
```

Рис. 4.4: Компиляция и запуск программы

Выполнил системную программу id. Результаты похожи. Gid и uid одинаковые, но команда id дает больше информации (рис. 4.5).



```
[guest@iavolgin ~]$ ./simpleid
uid=1003, gid=1003
[guest@iavolgin ~]$ id
uid=1003(guest) gid=1003(guest) groups=1003(guest) context=unconfined_u:unconfi
ned_r:unconfined_t:s0-s0:c0.c1023
[guest@iavolgin ~]$
```

Рис. 4.5: Команда id

Затем усложнил программу, добавив вывод действительных идентификаторов, скомпилировал и запустил simpleid2.c (рис. 4.6).

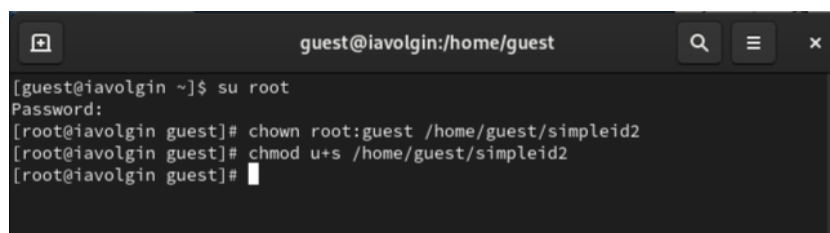
```

[guest@iavolgin ~]$ gcc simpleid2.c -o simpleid2
[guest@iavolgin ~]$ ./simpleid2
e_uid=1003, e_gid=1003
real_uid=1003, real_gid=1003
[guest@iavolgin ~]$

```

Рис. 4.6: Компиляция и запуск программы

От имени суперпользователя выполнил команды **chown root:guest /home/guest/simpleid2**, **chmod u+s /home/guest/simpleid2** (рис. 4.7).



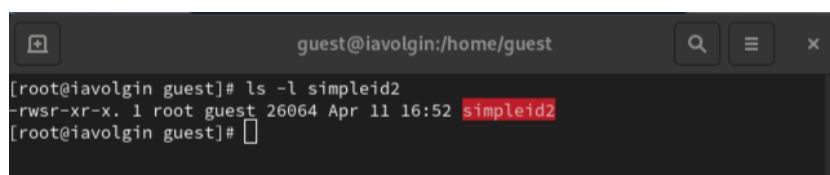
```

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

```

Рис. 4.7: Поменяла владельца программы

Далее выполнил проверку правильности установки новых атрибутов и смены владельца файла simpleid2 (рис. 4.8).



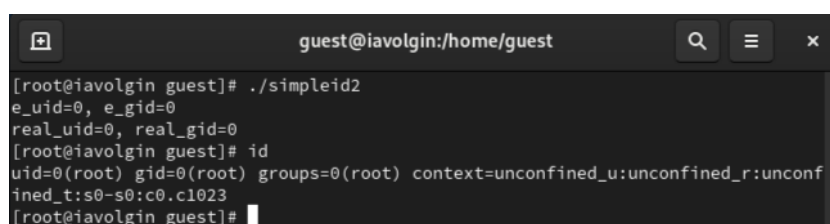
```

guest@iavolgin:/home/guest
[root@iavolgin guest]# ls -l simpleid2
-rwsr-xr-x. 1 root guest 26064 Apr 11 16:52 simpleid2
[root@iavolgin guest]#

```

Рис. 4.8: ls -l

Запустил simpleid2 и id. Результаты похожи. Gid и uid одинаковые, однако команда id дает больше информации (рис. 4.9).



```

guest@iavolgin:/home/guest
[root@iavolgin guest]# ./simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@iavolgin guest]# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@iavolgin guest]#

```

Рис. 4.9: Сравнение результатов

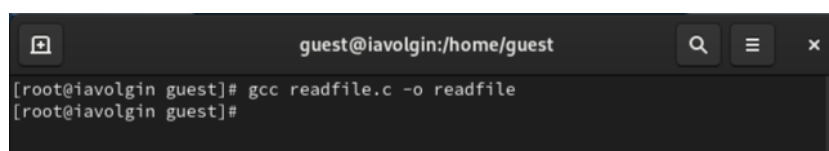
После этого создал программу readfile.c (рис. 4.10).



```
guest@iavolgin:/home/guest
GNU nano 5.6.1 readfile.c Modified
#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. 4.10: Создание программы

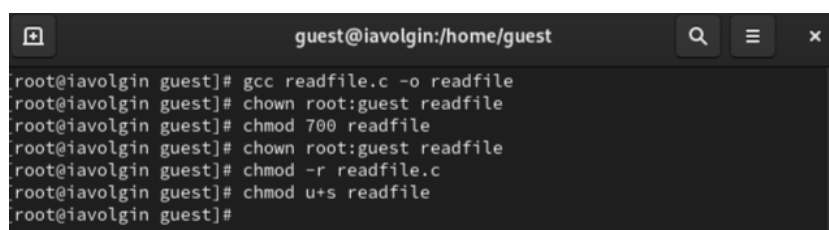
И откомпилировал её (рис. 4.11).



```
guest@iavolgin:/home/guest
[root@iavolgin guest]# gcc readfile.c -o readfile
[root@iavolgin guest]#
```

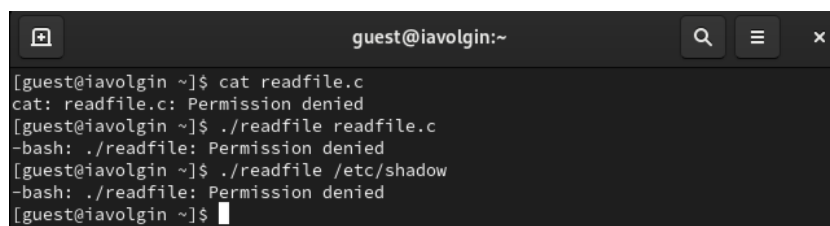
Рис. 4.11: Компиляция программы

Сменил владельца у файла readfile.c и изменил права так, чтобы только супер-пользователь (root) мог прочитать его (рис. 4.12) и проверил, может ли пользова-тель прочитать файл readfile.c. (рис. 4.13).



```
guest@iavolgin:/home/guest
[root@iavolgin guest]# gcc readfile.c -o readfile
[root@iavolgin guest]# chown root:guest readfile
[root@iavolgin guest]# chmod 700 readfile
[root@iavolgin guest]# chown root:guest readfile
[root@iavolgin guest]# chmod -r readfile.c
[root@iavolgin guest]# chmod u+s readfile
[root@iavolgin guest]#
```

Рис. 4.12: Смена владельца



```
guest@iavolgin:~  
[guest@iavolgin ~]$ cat readfile.c  
cat: readfile.c: Permission denied  
[guest@iavolgin ~]$ ./readfile readfile.c  
-bash: ./readfile: Permission denied  
[guest@iavolgin ~]$ ./readfile /etc/shadow  
-bash: ./readfile: Permission denied  
[guest@iavolgin ~]$
```

Рис. 4.13: Проверка

## 5 Выводы

Я изучил механизмы изменения идентификаторов, применения SetUID- и Sticky-битов. Получение практических навыков работы в консоли с дополнительными атрибутами.