

# **Информационная безопасность**

## **Лабораторная работа №5**

Матюшкин Денис Владимирович (НПИбд-02-21)

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Теоретическое введение</b>	<b>6</b>
<b>3</b>	<b>Ход работы</b>	<b>7</b>
3.1	Создание программы . . . . .	7
3.2	Исследование Sticky-бита . . . . .	12
<b>4</b>	<b>Выводы</b>	<b>15</b>
	<b>Список литературы</b>	<b>16</b>

## Список иллюстраций

3.1	simpleid.c . . . . .	7
3.2	Выполнение программы simpleid . . . . .	8
3.3	Выполнение программы simpleid2 . . . . .	8
3.4	Выполнение команд . . . . .	8
3.5	Проверка атрибутов и запуск программы . . . . .	9
3.6	Установка SetGID-бита . . . . .	9
3.7	Проверка атрибутов и запуск программы . . . . .	9
3.8	readfile.c . . . . .	10
3.9	Смена владельца и установка SetUID . . . . .	10
3.10	Проверка . . . . .	11
3.11	Работа в директории /tmp . . . . .	12
3.12	Работа с файлом file01.txt . . . . .	13
3.13	Снятие Sticky-бит с директории /tmp . . . . .	13
3.14	Повторение действий с file01.txt . . . . .	14
3.15	Возвращение Sticky-бит в директорию /tmp . . . . .	14

## **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

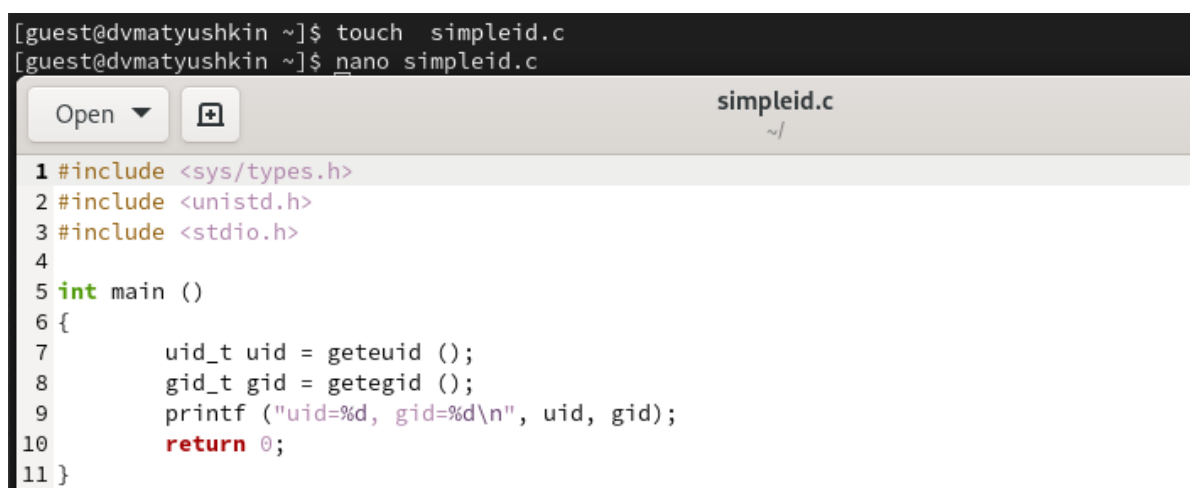
VirtualBox (Oracle VM VirtualBox) — программный продукт виртуализации для операционных систем Windows, Linux, FreeBSD, macOS, Solaris/OpenSolaris, ReactOS, DOS и других [1].

Rocky Linux — дистрибутив Linux, разработанный Rocky Enterprise Software Foundation. Предполагается, что это будет полный бинарно-совместимый выпуск, использующий исходный код операционной системы Red Hat Enterprise Linux (RHEL) [2].

## 3 Ход работы

### 3.1 Создание программы

1. Войдите в систему от имени пользователя guest.
2. Создайте программу simpleid.c (рис. 3.1).



The image shows a terminal window at the top with the following commands and output:

```
[guest@dvmatyushkin ~]$ touch simpleid.c
[guest@dvmatyushkin ~]$ nano simpleid.c
```

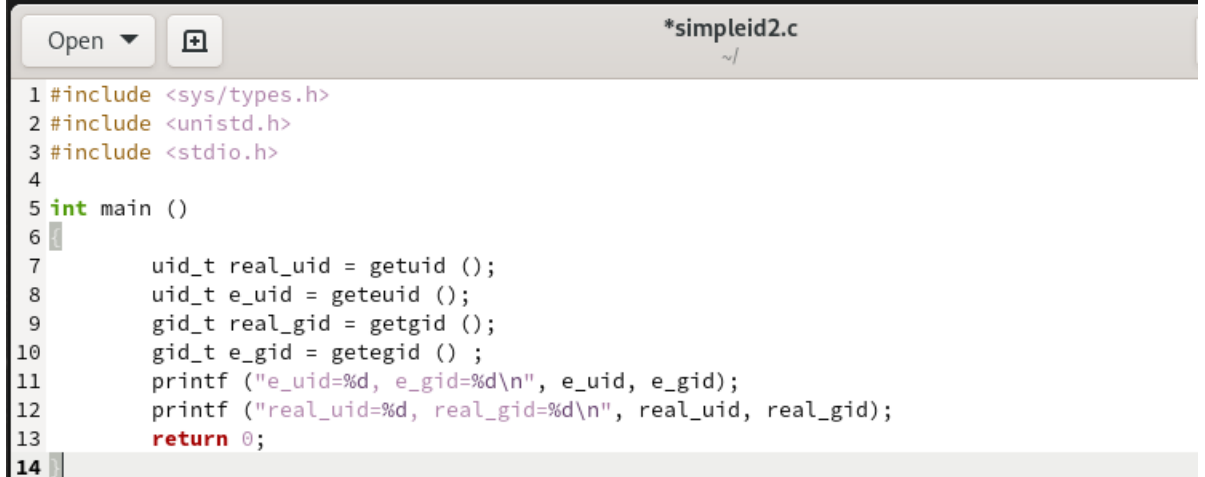
Below the terminal is a screenshot of the nano text editor editing the file simpleid.c. The code in the editor is as follows:

```
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 }
```

Рис. 3.1: simpleid.c

3. Скомпилируйте программу и убедитесь, что файл программы создан. Выполните программу simpleid и выполните системную программу id (рис. 3.2). Полученные результаты совпадают.

```
[guest@dvmatyushkin ~]$ ./simpleid
uid=1001, gid=1001
[guest@dvmatyushkin ~]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0
[guest@dvmatyushkin ~]$ touch simpleid2.c
[guest@dvmatyushkin ~]$
```



```
1 #include <sys/types.h>
2 #include <unistd.h>
3 #include <stdio.h>
4
5 int 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, real_gid);
13    return 0;
14 }
```

Рис. 3.2: Выполнение программы simpleid

4. Усложните программу, добавив вывод действительных идентификаторов (рис. 3.2).
5. Скомпилируйте и запустите simpleid2.c (рис. 3.3).

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

Рис. 3.3: Выполнение программы simpleid2

6. От имени суперпользователя выполните команды (рис. 3.4). Эти команды меняют владельца файла и устанавливают SetUID бит.

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

Рис. 3.4: Выполнение команд



7. Выполните проверку правильности установки новых атрибутов и смены владельца файла simpleid2. Запустите simpleid2 и id (рис. 3.5).

```
[guest@dvmatyushkin ~]$ ls -l simpleid2
-rwsr-xr-x. 1 root guest 17720 Oct  3 15:58 simpleid2
[guest@dvmatyushkin ~]$ ./simpleid2
e_uid=0, e_gid=1001
real_uid=1001, real_gid=1001
[guest@dvmatyushkin ~]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0
[guest@dvmatyushkin ~]$
```

Рис. 3.5: Проверка атрибутов и запуск программы

8. Прodelайте тоже самое относительно SetGID-бита (рис. 3.6 и 3.7).

```
[root@dvmatyushkin guest]# chmod g+s /home/guest/simpleid2
[root@dvmatyushkin guest]#
```

Рис. 3.6: Установка SetGID-бита

```
[guest@dvmatyushkin ~]$ ls -l simpleid2
-rwsr-sr-x. 1 root guest 17720 Oct  3 15:58 simpleid2
[guest@dvmatyushkin ~]$ ./simpleid2
e_uid=0, e_gid=1001
real_uid=1001, real_gid=1001
[guest@dvmatyushkin ~]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0
[guest@dvmatyushkin ~]$
```

Рис. 3.7: Проверка атрибутов и запуск программы

9. Создайте программу readfile.c. Откомпилируйте её (рис. 3.8).

```
[guest@dvmatyushkin ~]$ touch readfile.c

Open ▾ + *readfile.c ~/

1 #include <fcntl.h>
2 #include <stdio.h>
3 #include <sys/stat.h>
4 #include <sys/types.h>
5 #include <unistd.h>
6
7 int main (int argc, char* argv[])
8 {
9     unsigned char buffer[16];
10    size_t bytes_read;
11    int i;
12    int fd = open (argv[1], O_RDONLY);
13    do
14    {
15        bytes_read = read (fd, buffer, sizeof (buffer));
16        for (i = 0; i < bytes_read; ++i) printf("%c", buffer[i]);
17    }
18    while (bytes_read == sizeof (buffer));
19    close (fd);
20    return 0;
21 }
```

Рис. 3.8: readfile.c

10. Смените владельца у файла readfile.c (или любого другого текстового файла в системе) и измените права так, чтобы только суперпользователь (root) мог прочитать его, а guest не мог (рис. 3.9).

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

Рис. 3.9: Смена владельца и установка SetUID

11. Проверьте, что пользователь guest не может прочитать файл readfile.c (рис. 3.10).

```
[guest@dvmatyushkin ~]$ cat readfile.c
cat: readfile.c: Permission denied
[guest@dvmatyushkin ~]$ ls -l readfile.c
-rwx-----. 1 guest2 guest 417 Oct  3 16:03 readfile.c
[guest@dvmatyushkin ~]$ ./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;
}

[guest@dvmatyushkin ~]$ ./readfile /etc/shadow
*****v@*****<">*****>@*****>*****
*****e*****<*****>*****>*****
z*****<*****>*****>*****
0J@e8   p@
*****Uq2S*****x86_64./readfile/etc/shadowSHELL=/bin/bashSESS
```

Рис. 3.10: Проверка

12. Смените у программы readfile владельца и установите SetUID-бит (рис. 3.9).
13. Проверьте, может ли программа readfile прочитать файл readfile.c (рис. 3.10).
14. Проверьте, может ли программа readfile прочитать файл /etc/shadow (рис. 3.10).

Команды выполняются, т.к. Setuid – это бит разрешения, который позволяет пользователю запускать исполняемый файл с правами владельца этого файла.

## 3.2 Исследование Sticky-бита

1. Выясните, установлен ли атрибут Sticky на директории /tmp (рис. 3.11).
2. От имени пользователя guest создайте файл file01.txt в директории /tmp со словом test (рис. 3.11).
3. Просмотрите атрибуты у только что созданного файла и разрешите чтение и запись для категории пользователей «все остальные» (рис. 3.11).

```
[guest@dvmatyushkin ~]$ ls -l / | grep tmp
drwxrwxrwt. 17 root root 4096 Oct  3 16:11 tmp
[guest@dvmatyushkin ~]$ echo "test" > /tmp/file01.txt
[guest@dvmatyushkin ~]$ ls -l /tmp/file01.txt
-rw-r--r--. 1 guest guest 5 Oct  3 16:13 /tmp/file01.txt
[guest@dvmatyushkin ~]$ chmod o+rw /tmp/file01.txt
[guest@dvmatyushkin ~]$ ls -l /tmp/file01.txt
-rw-r--rw-. 1 guest guest 5 Oct  3 16:13 /tmp/file01.txt
[guest@dvmatyushkin ~]$
```

Рис. 3.11: Работа в директории /tmp

4. От пользователя guest2 (не являющегося владельцем) попробуйте прочитать файл /tmp/file01.txt (рис. 3.12).
5. От пользователя guest2 попробуйте дозаписать в файл /tmp/file01.txt слово test2 (рис. 3.12).
6. Проверьте содержимое файла командой (рис. 3.12).
7. От пользователя guest2 попробуйте записать в файл /tmp/file01.txt слово test3, стерев при этом всю имеющуюся в файле информацию (рис. 3.12).
8. Проверьте содержимое файла командой (рис. 3.12).
9. От пользователя guest2 попробуйте удалить файл /tmp/file01.txt (рис. 3.12).

```
[guest2@dvmatyushkin guest]$ echo "test2" > /tmp/file01.txt
bash: /tmp/file01.txt: Permission denied
[guest2@dvmatyushkin guest]$ cat /tmp/file01.txt
test
[guest2@dvmatyushkin guest]$ echo "test3" > /tmp/file01.txt
bash: /tmp/file01.txt: Permission denied
[guest2@dvmatyushkin guest]$ cat /tmp/file01.txt
test
[guest2@dvmatyushkin guest]$ rm /tmp/file01.txt
rm: cannot remove '/tmp/file01.txt': No such file or directory
```

Рис. 3.12: Работа с файлом file01.txt

Удалить файл нам не получилось.

10. Повысьте свои права до суперпользователя и выполните после этого команду, снимающую атрибут t (Sticky-бит) с директории /tmp. После покиньте режим суперпользователя (рис. 3.13).

```
[guest2@dvmatyushkin guest]$ su -
Password:
[root@dvmatyushkin ~]# chmod -t /tmp
[root@dvmatyushkin ~]# exit
logout
[guest2@dvmatyushkin guest]$ ls -l / | grep tmp
drwxrwxrwx. 17 root root 4096 Oct  3 16:16 tmp
[guest2@dvmatyushkin guest]$
```

Рис. 3.13: Снятие Sticky-бит с директории /tmp

11. От пользователя guest2 проверьте, что атрибута t у директории /tmp нет (рис. 3.13).
12. Повторите предыдущие шаги. Теперь можем все то же самое, но еще и удалять файл (рис. 3.14).

```
[guest2@dvmatyushkin tmp]$ cat /tmp/file01.txt
test
[guest2@dvmatyushkin tmp]$ echo "test2" > /tmp/file01.txt
bash: /tmp/file01.txt: Permission denied
[guest2@dvmatyushkin tmp]$ echo "test2" > /tmp/file01.txt
bash: /tmp/file01.txt: Permission denied
[guest2@dvmatyushkin tmp]$ cat /tmp/file01.txt
test
[guest2@dvmatyushkin tmp]$ echo "test3" > /tmp/file01.txt
bash: /tmp/file01.txt: Permission denied
[guest2@dvmatyushkin tmp]$ rm /tmp/file01.txt
rm: cannot remove '/tmp/file01.txt': No such file or directory
[guest2@dvmatyushkin tmp]$ rm /tmp/file01.txt
rm: remove write-protected regular file '/tmp/file01.txt'?
```

Рис. 3.14: Повторение действий с file01.txt

13. Повысьте свои права до суперпользователя и верните атрибут `t` на директорию `/tmp` (рис. 3.15).

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

Рис. 3.15: Возвращение Sticky-бит в директорию `/tmp`

## 4 Выводы

В ходе данной лабораторной работы мы изучили механизм изменения идентификаторов, применение SetUID- и Sticky-битов. Получили практические навыки работы в консоли с дополнительными атрибутами. Рассмотрели работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

## Список литературы

1. VirtualBox Documentation [Электронный ресурс]. Oracle, 2024. URL: <https://www.virtualbox.org/wiki/Documentation>.
2. Rocky Documentation [Электронный ресурс]. Rocky Enterprise Software Foundation, 2024. URL: <https://docs.rockylinux.org/ru/>.