

Отчёт по лабораторной работе №5

дисциплина: Информационная безопасность

Морозова Ульяна

Содержание

1	Цель работы	4
2	Подготовка к лабораторной работе	5
3	Выполнение лабораторной работы	6
3.1	Создание программ	6
3.2	Исследование Sticky-бита	10
4	Выводы	13

Список иллюстраций

2.1	gcc -v	5
3.1	Создание файла	6
3.2	Выполнение программы simpleid	6
3.3	Выполнение команды id	6
3.4	simpleid2.c	7
3.5	Запуск simpleid2	7
3.6	Изменение атрибута s	7
3.7	Выполнение команд	7
3.8	readfile.c	8
3.9	Компиляция программы	8
3.10	Смена владельца	8
3.11	Проверка	8
3.12	SetU'D-бит	8
3.13	Чтение файлов	9
3.14	Чтение файлов	9
3.15	Чтение файлов	10
3.16	tmp	10
3.17	Изменение прав	10
3.18	Чтение файла	11
3.19	Работа с файлом	11
3.20	Убираем атрибут	11
3.21	Проверка	11
3.22	Повторение команд	11
3.23	Возврат атрибута	12

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 Подготовка к лабораторной работе

Перед тем как начнем выполнять задания убедимся, что у нас установлен компилятор gcc (рис. 2.1).

```
[ukmorofova@localhost ~]$ 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 11.4.1 20231218 (Red Hat 11.4.1-3) (GCC)
```

Рис. 2.1: gcc -v

3 Выполнение лабораторной работы

3.1 Создание программ

1. Зайдем систему от имени пользователя ukmorozeva и создадим файл simpleid.c (рис. 3.1).

```
[ukmorozeva@localhost ~]$ touch simpleid.c
[ukmorozeva@localhost ~]$ ls
Desktop  Downloads  Pictures  simpleid.c  Videos
Documents Music      Public   Templates
[ukmorozeva@localhost ~]$ gedit simpleid.c
```

Рис. 3.1: Создание файла

2. Скомпилируем программу и убедимся, что исполняемый файл был создан, затем выполним программу simpleid и сравним ее с выполнением команды id (рис. 3.2 - рис. 3.3)

```
[ukmorozeva@localhost ~]$ gcc simpleid.c -o simpleid
[ukmorozeva@localhost ~]$ ls
Desktop  Downloads  Pictures  simpleid  Templates
Documents Music      Public   simpleid.c  Videos
[ukmorozeva@localhost ~]$ ./simpleid
uid=1000, gid=1000
```

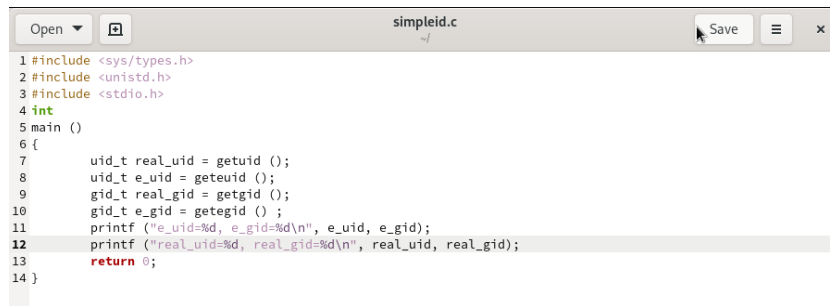
Рис. 3.2: Выполнение программы simpleid

```
[ukmorozeva@localhost ~]$ ./simpleid
uid=1000, gid=1000
[ukmorozeva@localhost ~]$ id
uid=1000(ukmorozeva) gid=1000(ukmorozeva) groups=1000(ukmorozeva),10(wheel) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Рис. 3.3: Выполнение команды id

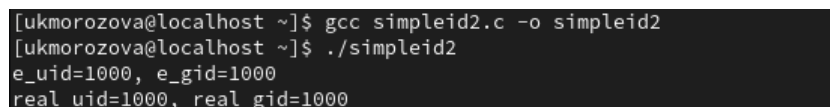
Как видно, команды выводят одинаковую информацию.

3. Усложним программу (рис. 3.4) скомпилируем и запустим ее (рис. 3.5).



```
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, real_gid);
13    return 0;
14 }
```

Рис. 3.4: simpleid2.c



```
[ukmorofova@localhost ~]$ gcc simpleid2.c -o simpleid2
[ukmorofova@localhost ~]$ ./simpleid2
e_uid=1000, e_gid=1000
real_uid=1000, real_gid=1000
```

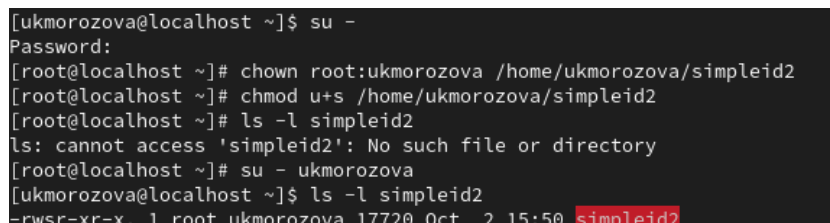
Рис. 3.5: Запуск simpleid2

4. От имени суперпользователя выполняем следующие команды

`chown root:guest /home/guest/simpleid2`

`chmod u+s /home/guest/simpleid2`

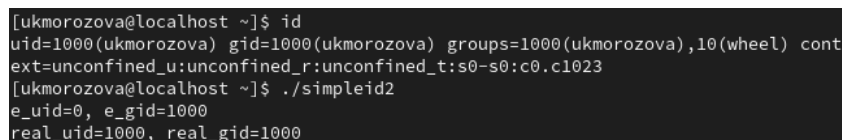
и проверяем правильность выполнения команд (рис. 3.6).



```
[ukmorofova@localhost ~]$ su -
Password:
[root@localhost ~]# chown root:ukmorofova /home/ukmorofova/simpleid2
[root@localhost ~]# chmod u+s /home/ukmorofova/simpleid2
[root@localhost ~]# ls -l simpleid2
ls: cannot access 'simpleid2': No such file or directory
[root@localhost ~]# su - ukmorofova
[ukmorofova@localhost ~]$ ls -l simpleid2
-rwsr-xr-x. 1 root ukmorofova 17720 Oct  2 15:50 simpleid2
```

Рис. 3.6: Изменение атрибута s

Запускаем программу simpleid2 и команду id (рис. 3.7).



```
[ukmorofova@localhost ~]$ id
uid=1000(ukmorofova) gid=1000(ukmorofova) groups=1000(ukmorofova),10(wheel) cont
ext=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[ukmorofova@localhost ~]$ ./simpleid2
e_uid=0, e_gid=1000
real_uid=1000, real_gid=1000
```


Рис. 3.7: Выполнение команд

5. Создаем программу readfile.c (рис. 3.8) и откомпилируем ее (рис. 3.9)



```
1 #include <fcntl.h>
2 #include <stdio.h>
3 #include <sys/stat.h>
4 #include <sys/types.h>
5 #include <unistd.h>
6 int
7 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 }
22
```

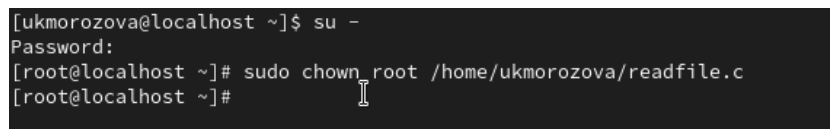
Рис. 3.8: readfile.c



```
[ukmorofova@localhost ~]$ gcc readfile.c -o readfile
[ukmorofova@localhost ~]$ ls
Desktop  Downloads  Pictures  readfile  simpleid  simpleid2.c  Templates
Documents Music      Public   readfile.c simpleid2 simpleid.c  Videos
```

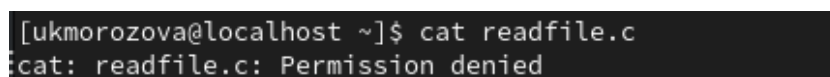
Рис. 3.9: Компиляция программы

Меняем владельца файла, что ukmorofova не мог прочитать его (рис. 3.10), проверяем (рис. 3.11).



```
[ukmorofova@localhost ~]$ su -
Password:
[root@localhost ~]# sudo chown root /home/ukmorofova/readfile.c
[root@localhost ~]#
```

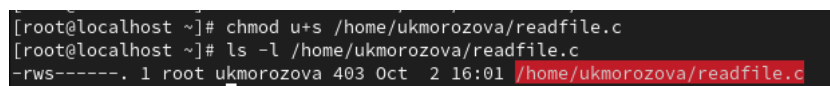
Рис. 3.10: Смена владельца



```
[ukmorofova@localhost ~]$ cat readfile.c
cat: readfile.c: Permission denied
```

Рис. 3.11: Проверка

Установим SetU'D-бит (рис. 3.12).



```
[root@localhost ~]# chmod u+s /home/ukmorofova/readfile.c
[root@localhost ~]# ls -l /home/ukmorofova/readfile.c
-rwsr--r--. 1 root ukmorofova 403 Oct  2 16:01 /home/ukmorofova/readfile.c
```

Рис. 3.12: SetU'D-бит

3.13 - рис. 3.15))



Рис. 3.13: Чтение файлов



Рис. 3.14: Чтение файлов

```
[ukmorofova@localhost ~]$ ./readfile simpleid.c
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>
int
main ()
{
    uid_t real_uid = getuid ();
    uid_t e_uid = geteuid ();
    gid_t real_gid = getgid ();
    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;
}
```

Рис. 3.15: Чтение файлов

3.2 Исследование Sticky-бита

1. Выясним, установлен ли атрибут Sticky на директории /tmp и создадим файл file01.txt в директории /tmp со словом test (рис. 3.16).

```
[ukmorofova@localhost ~]$ ls -l / | grep tmp
drwxrwxrwt. 20 root root 4096 Oct  2 16:09 tmp
[ukmorofova@localhost ~]$ echo "test" > /tmp/file01.txt
```

Рис. 3.16: tmp

3. Просмотрим атрибуты у только что созданного файла и разрешим чтение и запись для категории пользователей «все остальные» (рис. 3.17)

```
[ukmorofova@localhost ~]$ ls -l /tmp/file01.txt
-rw-r--r--. 1 ukmorofova ukmorofova 5 Oct  2 16:12 /tmp/file01.txt
[ukmorofova@localhost ~]$ chmod o+rw /tmp/file01.txt
[ukmorofova@localhost ~]$ ls -l /tmp/file01.txt
-rw-r--rw-. 1 ukmorofova ukmorofova 5 Oct  2 16:12 /tmp/file01.txt
```

Рис. 3.17: Изменение прав

4. От имени другого пользователя попробуем прочитать файл /tmp/file01.txt (рис. 3.18).

```
[ukmorofova@localhost ~]$ su - guest2
Password:
[guest2@localhost ~]$ cat /tmp/file01.txt
test
```

Рис. 3.18: Чтение файла

5. Попробуем дозаписать в файл /tmp/file01.txt слово test2 командой и проверим содержимое файла, затем попробуем удалить его (рис. 3.19).

```
[guest2@localhost ~]$ echo "test2" > /tmp/file01.txt
[guest2@localhost ~]$ cat /tmp/file01.txt
test2
[guest2@localhost ~]$ echo "test3" > /tmp/file01.txt
[guest2@localhost ~]$ cat /tmp/file01.txt
test3
[guest2@localhost ~]$ rm /tmp/file01.txt
rm: cannot remove '/tmp/file01.txt': Operation not permitted
```

Рис. 3.19: Работа с файлом

6. Повысим свои права до суперпользователя и выполним команду, снимающую атрибут t (Sticky-бит) с директории /tmp (рис. 3.20).

```
[root@localhost ~]# chmod -t /tmp
[root@localhost ~]# exit
logout
[guest2@localhost ~]$
```

Рис. 3.20: Убираем атрибут

Проверим выполнение команды от имени guest2 (рис. 3.21) и повторим шаги выше (рис. 3.22).

```
[guest2@localhost ~]$ ls -l / | grep tmp
drwxrwxrwx. 20 root root 4096 Oct  2 16:18 tmp
```

Рис. 3.21: Проверка

```
[guest2@localhost ~]$ echo "test4" > /tmp/file01.txt
[guest2@localhost ~]$ cat /tmp/file01.txt
test4
[guest2@localhost ~]$ rm /tmp/file01.txt
```

Рис. 3.22: Повторение команд

После всего возвращаем атрибут t (рис. 3.23).

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

Рис. 3.23: Возврат атрибута

4 Выводы

Изучили механизмы изменения идентификаторов, применили SetUID- и Sticky-битов.