

Защита лабораторной работы №5

Информационная безопасность

Данзанова С. З.

2024

Российский университет дружбы народов, Москва, Россия

Докладчик

.....: {.columns align=center} ::: {.column width="70%"}

- Данзанова Саяна Зоригтоевна
- Студентка группы НПИбд-01-20
- Студ. билет 1032217624
- Российский университет дружбы народов

Цель лабораторной 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

1. Дополнительные атрибуты файлов Linux

В Linux существует три основных вида прав — право на чтение (read), запись (write) и выполнение (execute), а также три категории пользователей, к которым они могут применяться — владелец файла (user), группа владельца (group) и все остальные (others). Но, кроме прав чтения, выполнения и записи, есть еще три дополнительных атрибута. [1]

2. Компилятор GCC

GCC - это свободно доступный оптимизирующий компилятор для языков C, C++. Собственно программа gcc это некоторая надстройка над группой компиляторов, которая способна анализировать имена файлов, передаваемые ей в качестве аргументов, и определять, какие действия необходимо выполнить. Файлы с расширением .cc или .C рассматриваются, как файлы на языке C++, файлы с расширением .c как программы на языке C, а файлы с расширением .o считаются объектными. [2]

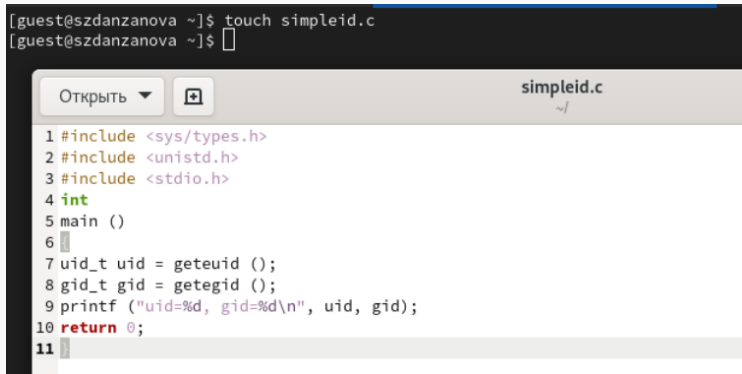
5.2.1. Подготовка лабораторного стенда

```
[guest@szdanzanova ~]$ su
Пароль:
[root@szdanzanova guest]# yum install gcc
Extra Packages for Enterprise Linux 9 - x86_64    52 kB/s | 38 kB    00:00
Extra Packages for Enterprise Linux 9 - x86_64    2.0 MB/s | 23 MB    00:11
Extra Packages for Enterprise Linux 9 openh264    3.5 kB/s | 993 B    00:00
packages for the GitHub CLI                      17 kB/s | 3.0 kB    00:00
packages for the GitHub CLI                      5.7 kB/s | 2.7 kB    00:00
Rocky Linux 9 - BaseOS                           7.7 kB/s | 4.1 kB    00:00
Rocky Linux 9 - BaseOS                           1.7 MB/s | 2.3 MB    00:01
Rocky Linux 9 - AppStream                        11 kB/s | 4.5 kB    00:00
Rocky Linux 9 - AppStream                        1.9 MB/s | 8.0 MB    00:04
Rocky Linux 9 - Extras                          8.6 kB/s | 2.9 kB    00:00
Пакет gcc-11.4.1-3.el9.x86_64 уже установлен.
Зависимости разрешены.
Отсутствуют действия для выполнения.
Выполнено!
[root@szdanzanova guest]# setenforce 0
[root@szdanzanova guest]# getenforce
Permissive
```

Рис. 1: (Установка gss)

5.3.1 Создание программы

Создали программу simpleid.c



The image shows a terminal window at the top with the command `touch simpleid.c` being executed. Below the terminal is a code editor window titled `simpleid.c`. The code in the editor is as follows:

```
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;
11 }
```

Рис. 2: (simpleid.c)

5.3.1 Создание программы

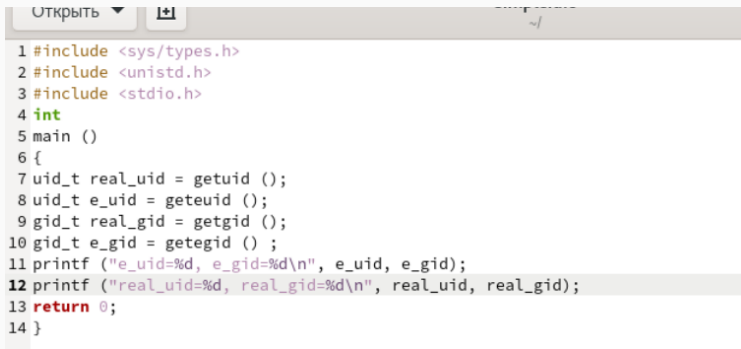
Скомпилировали и выполнили программу simpleid. Затем выполнили системную программу id и сравнили полученные результаты

```
[guest@szdanzanova ~]$ touch simpleid.c
[guest@szdanzanova ~]$ gcc simpleid.c -o simpleid
[guest@szdanzanova ~]$ ./simpleid
uid=1001, gid=1001
[guest@szdanzanova ~]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfi
ned_r:unconfined_t:s0-s0:c0.c1023
```

Рис. 3: (3-5 пункты задания лабораторной)

5.3.1 Создание программы

Усложнили программу, добавив вывод действительных идентификаторов



```
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. 4: (simpleid2.c)

5.3.1 Создание программы

Скомпилировали и выполнили программу simpleid2

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

Рис. 5: (7 пункт задания лабораторной)

5.3.1 Создание программы

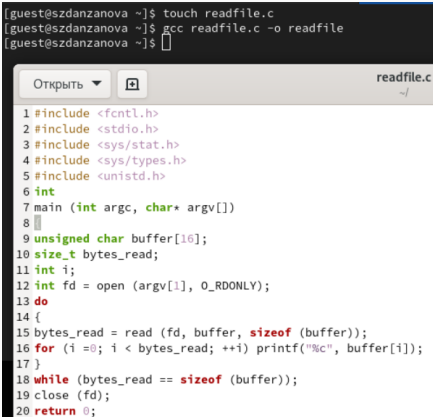
От имени суперпользователя выполнили команды и проверили правильность установки новых атрибутов и смены владельца файла. Запустили `simpleid2` и `id`. Сравнили результаты. Проделали то же самое относительно SetGID-бита

```
[root@szdanzanova guest]# chown root:guest /home/guest/simpleid2
[root@szdanzanova guest]# chmod u+s /home/guest/simpleid2
[root@szdanzanova guest]# ls -l simpleid2
-rwsr-xr-x. 1 root guest 17720 окт  5 19:13 simpleid2
[root@szdanzanova guest]# ./simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@szdanzanova guest]# id
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@szdanzanova guest]# chown root:guest /home/guest/simpleid2
[root@szdanzanova guest]# chmod g+s /home/guest/simpleid2
[root@szdanzanova guest]# ls -l simpleid2
-rwxr-sr-x. 1 root guest 17720 окт  5 19:13 simpleid2
[root@szdanzanova guest]# ./simpleid2
e_uid=0, e_gid=1001
real_uid=0, real_gid=0
[root@szdanzanova guest]# id
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

5.3.1 Создание программы

Скомпилировали программу readfile.c

```
[guest@szdanzanova ~]$ touch readfile.c
[guest@szdanzanova ~]$ gcc readfile.c -o readfile
[guest@szdanzanova ~]$
```



```
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 }
```

Рис. 7: (readfile.c)

5.3.1 Создание программы

Сменили владельца у файла и изменили права так, чтобы только суперпользователь мог прочитать его, а guest не мог

```
[guest@szdanzanova ~]$ su
Пароль:
[root@szdanzanova guest]# chown root:guest readfile
[root@szdanzanova guest]# chown 700 readfile
[root@szdanzanova guest]# chown root:guest readfile
[root@szdanzanova guest]# chown -r readfile.c
chown: неверный ключ - «r»
По команде «chown --help» можно получить дополнительную информацию.
[root@szdanzanova guest]# chmod -r readfile.c
[root@szdanzanova guest]# chmod u+s readfile
```

Рис. 8: (chmod)

5.3.1 Создание программы

Проверили, что guest не может прочитать файл. Сменили у программы readfile владельца и установили SetU'D-бит. Проверили, может ли программа readfile прочитать файл readfile.c, файл /etc/shadow

```
[guest@szdanzanova ~]$ cat readfile.c
cat: readfile.c: Отказано в доступе
[guest@szdanzanova ~]$ ./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@szdanzanova ~]$ ./readfile /etc/shadow
root:*$6$0S.X8qKi.B7YyXaY$sieLpkqyQVlckEygcbLYv6tu00BEa3GkaCrLIoJ8ud0hB99gLHb2Nz
piqrD6eYEF6FOY5cQAGVTMEGtK8cq20::0:99999:7:::
bin:*.19820:0:99999:7:::
daemon:*.19820:0:99999:7:::
adm:*.19820:0:99999:7:::
lp:*.19820:0:99999:7:::
sync:*.19820:0:99999:7:::
shutdown:*.19820:0:99999:7:::
halt:*.19820:0:99999:7:::
mail:*.19820:0:99999:7:::
operator:*.19820:0:99999:7:::
games:*.19820:0:99999:7:::
ftp:*.19820:0:99999:7:::
nobody:*.19820:0:99999:7:::
```

5.3.2. Исследование Sticky-бита

Выяснили, установлен ли атрибут Sticky на директории /tmp, создали файл file01.txt со словом test. Просмотрели атрибуты у только что созданного файла и разрешили чтение и запись для категории пользователей «все остальные»

```
[guest@szdanzanova ~]$ ls -l / | grep tmp
drwxrwxrwt. 18 root root 4096 окт  5 20:02 tmp
[guest@szdanzanova ~]$ echo "test" > /tmp/file01.txt
[guest@szdanzanova ~]$ ls -l /tmp/file01.txt
-rw-r--r--. 1 guest guest 5 окт  5 20:08 /tmp/file01.txt
[guest@szdanzanova ~]$ chmod o+rw /tmp/file01.txt
[guest@szdanzanova ~]$ ls -l /tmp/file01.txt
-rw-r--rw-. 1 guest guest 5 окт  5 20:08 /tmp/file01.txt
```

Рис. 10: (1-3 пункты)

5.3.2. Исследование Sticky-бита

От guest2 попробовали прочесть файл, дозаписать слово test2, затем записать слово test3, стерев при этом всю имеющуюся в файле информацию. Попробовали удалить файл. Этого сделать не удалось.

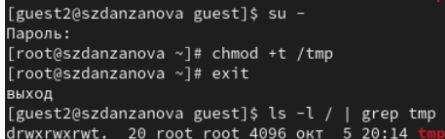
Повысили свои права до суперпользователя и сняли атрибут t с директории /tmp. От guest2 проверили, что атрибута t у директории /tmp нет

```
[guest@szdanzanova ~]$ su guest2
Пароль:
[guest2@szdanzanova guest]$ cat /tmp/file01.txt
test
[guest2@szdanzanova guest]$ echo "test2" > /tmp/file01.txt
bash: /tmp/file01.txt: Отказано в доступе
[guest2@szdanzanova guest]$ cat /tmp/file01.txt
test
[guest2@szdanzanova guest]$ rm /tmp/file01.txt
rm: удалить защищённый от записи обычный файл '/tmp/file01.txt'?
[guest2@szdanzanova guest]$ su -
Пароль:
[root@szdanzanova ~]# chmod -t /tmp
[root@szdanzanova ~]# exit
```


5.3.2. Исследование Sticky-бита

Повторили предыдущие шаги. При повторении всё получилось. Удалось удалить файл от имени пользователя, не являющегося его владельцем.

Повысили свои права до суперпользователя и вернули атрибут `t` на директорию `/tmp`



```
[guest2@szdanzanova guest]$ su -  
Пароль:  
[root@szdanzanova ~]# chmod +t /tmp  
[root@szdanzanova ~]# exit  
выход  
[guest2@szdanzanova guest]$ ls -l / | grep tmp  
drwxrwxrwt. 20 root root 4096 окт 5 20:14 tmp
```

Рис. 12: (Возвращение атрибута)

- Были изучены механизмы изменения идентификаторов и применения SetUID- и Sticky-битов. Получены практические навыки работы в консоли с дополнительными атрибутами. Были рассмотрены работа механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов

Список литературы. Библиография

[0] Методические материалы курса

[1] Дополнительные атрибуты: <https://tokmakov.msk.ru/blog/item/141>

[2] Компилятор GSS: <http://parallel.imm.uran.ru/freesoft/make/instrum.html>