Лабораторная работа 5

Попов Дмитрий Павлович, НФИбд-01-19

Содержание

List of Figures

List of Tables

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №5

дисциплина: Информационная безопасность

Преподователь: Кулябов Дмитрий Сергеевич

Студент: Попов Дмитрий Павлович

Группа: НФИбд-01-19

МОСКВА

2022 г.

1 Цель работы

Изучение механизмов изменения идентификаторов, применения SetUID и Sticky-битов.

2 Выполнение лабораторной работы

1. Создал программу simpleid.c (Рис fig. 2.1).

```
[guest@dpopov ~]$ touch simpleid.c
[guest@dpopov ~]$ nano simpleid.c
[guest@dpopov ~]$ cat 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;
}
```

Figure 2.1: simpleid

2. Скомпилировал и выполнил программу. Сравнил с id. Как видим, результат работы команд - одинаковый (Рис fig. 2.2).

```
[guest@dpopov ~]$ gcc simpleid.c -o simpleid
[guest@dpopov ~]$ ./simpleid
uid=1001, gid=1001
[guest@dpopov ~]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfin
ed r:unconfined t:s0-s0:c0.c1023
```

Figure 2.2: compile and run

3. Усложнил программу, добавив вывод действительных идентификаторов (Рис fig. 2.3).

```
[guest@dpopov ~]$ cat simpleid2.c
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>

int
main ()
{
        uid_t real_uid = getuid ();
        gid_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;
}
```

Figure 2.3: simpleid2.c

4. Скомпилировал и запустил simpleid2.c (Рис fig. 2.4).

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

Figure 2.4: simpleid2

5. От имени суперпользователя выполнил команды (Рис fig. 2.5)

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

Figure 2.5: chmod

6. Запустил simpleid2 и id (Рис fig. 2.6)

```
[guest@dpopov ~]$ ls -l simpleid2
-rwsrwxr-x. 1 root guest 26008 Oct  8 18:13 simpleid2
[guest@dpopov ~]$ ./simpleid2
e_uid=0, e_gid=1001
real_uid=1001, real_gid=1001
[guest@dpopov ~]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfin
ed_r:unconfined_t:s0-s0:c0.c1023
```

Figure 2.6: simpleid2 run

7. Создал программу readfile.c: (Рис fig. 2.7)

```
[guest@dpopov ~]$ cat 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;
```

Figure 2.7: readfile.c

8. Сменил владельца у файла readfile.c и изменил права так, чтобы только суперпользователь (root) мог прочитать его, а guest не мог (Рис fig. 2.8).

```
[root@dpopov ~]# chown root:guest /home/guest/readfile.c
[root@dpopov ~]# chmod 700 /home/guest/readfile.c
```

Figure 2.8: chown

9. guest не может прочитать файл readfile.c (Рис fig. 2.9)

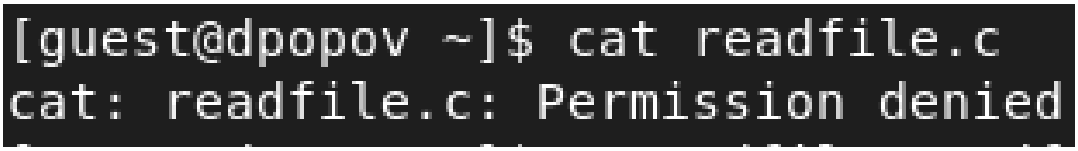

Figure 2.9: can not read

10. Сменил у программы readfile владельца и установил SetU'D-бит (Рис fig. 2.10)

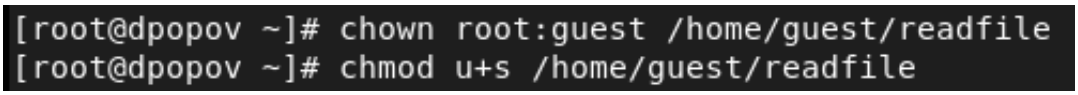

Figure 2.10: readfile

11. Проверил прочитать файл readfile и /etc/shadow (Рис fig. 2.11 и fig. 2.12)

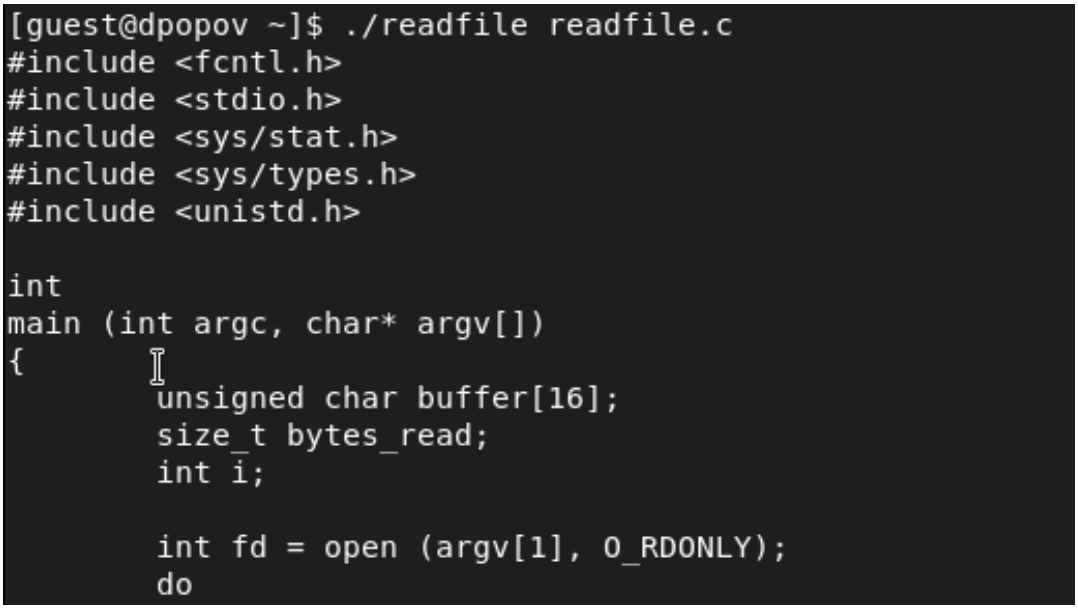

Figure 2.11: readfile read

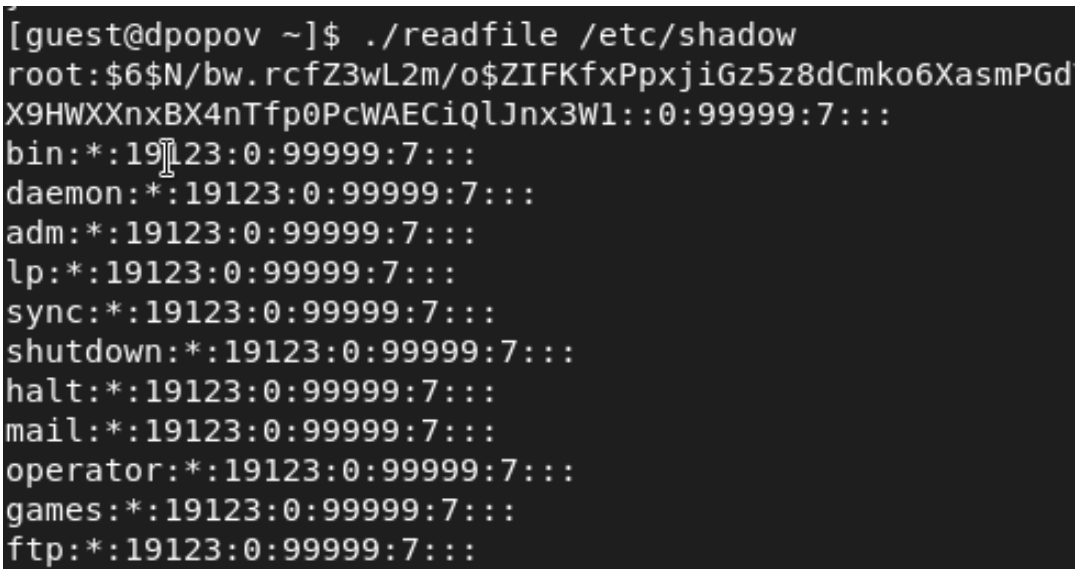

Figure 2.12: /etc/shadow read

12. readfile удалось прочитать, /etc/shadow - тоже удалось

13. Проверил sticky бит на категории tmp. Создал файл в tmp от guest и посмотерл атрибуты (Рис fig. 2.13).

Figure 2.13: sticky

14. От guest2 попробовал выполнить различные операции (Рис fig. 2.14)

```
[guest2@dpopov ~]$ echo "test" > /tmp/file01.txt
[guest2@dpopov ~]$ echo "test2" >> /tmp/file01.txt
[guest2@dpopov ~]$ cat /tmp/file01.txt
test
test2
[guest2@dpopov ~]$ echo "test3" > /tmp/file01.txt
[guest2@dpopov ~]$ cat /tmp/file01.txt
test3
```

Figure 2.14: guest2 file01

15. Не удалось выполнить только rm

16. Снял атрибут t (Sticky-бит) сдиректории /tmp (Рис fig. 2.15)

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

Figure 2.15: -t

17. Повторил предыдущие шаги. rm теперь работает (Рис fig. 2.16)

```
[guest2@dpopov ~]$ echo "test2" >> /tmp/file01.txt
[guest2@dpopov ~]$ cat /tmp/file01.txt
test3
test2
[guest2@dpopov ~]$ rm /tmp/file01.txt
[guest2@dpopov ~]$ su -
```

Figure 2.16: guest2 file01 try 2

3 Вывод

Выполнив данную лабораторную работу, я получение практические навыков работы в консоли с дополнительными атрибутами. Рассмотрел работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

4 Список литературы

1. Кулябов, Д.С. - Лабораторная работа № 5. Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов https://esystem.rudn.ru/pluginfile.php/1651889/mod_resource/content/2/005-lab_discret_sticky.pdf