Лабораторная работа 5

Щепелева Марина Евгеньевна, НФИбд-01-19

Содержание

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: Щепелева Марина Евгеньевна

Группа: НФИбд-01-19

МОСКВА

2022 г.

Цель работы

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

Выполнение лабораторной работы

1. Создала программу simpleid.c.

```
[guest@shepeleva lab5]$ 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;
}
```

simpleid

2. Скомпилировала и выполнила программу. Сравнил с id. Как видим, результат работы команд - одинаковый.

```
[guest@shepeleva lab5]$ gcc simpleid.c -o simpleid
[guest@shepeleva lab5]$ ls
readfile.c  simpleid  sinpleid2.c  simpleid.c
[guest@shepeleva lab5]$ ./simpleid
uid=1001, gid=1001
[guest@shepeleva lab5]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfin
ed_r:unconfined_t:s0-s0:c0.c1023
```

compile and run

3. Усложнил программу, добавив вывод действительных идентификаторов.

```
#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;
}
```

simpleid2.c

4. Скомпилировала и запустила simpleid2.c.

```
[guest@shepeleva lab5]$ gcc simpleid2.c -o simpleid2
[guest@shepeleva lab5]$ ls
readfile.c  simpleid  sinpleid2  simpleid2.c  sinpleid.c
[guest@shepeleva lab5]$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
```

simpleid2

5. От имени суперпользователя выполнила команды

```
[root@shepeleva ~]# chown root:guest /home/guest/lab5/simpleid2
[root@shepeleva ~]# chmod u+s /home/guest/lab5/simpleid2
```

chmod

6. Запустила simpleid2 и id

```
[guest@shepeleva lab5]$ ./simpleid2
e_uid=0, e_gid=1001
real_uid=1001, real_gid=1001
[guest@shepeleva lab5]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfi
ned_r:unconfined_t:s0-s0:c0.c1023
```

simpleid2 run

7. Создала программу readfile.c:

readfile.c

readfile.c

8. Сменила владельца у файла readfile.c и изменила права так, чтобы только суперпользователь (root) мог прочитать его, а guest не мог.

```
[root@shepeleva ~]# chown root:guest /home/guest/lab5/readfile.c
[root@shepeleva ~]# chmod 700 /home/guest/lab5/readfile.c
```

chown

9. guest не может прочитать файл readfile.c

![cant read](img/9.png "cant read")

10. Сменила у программы readfile владельца и установила SetU'D-бит

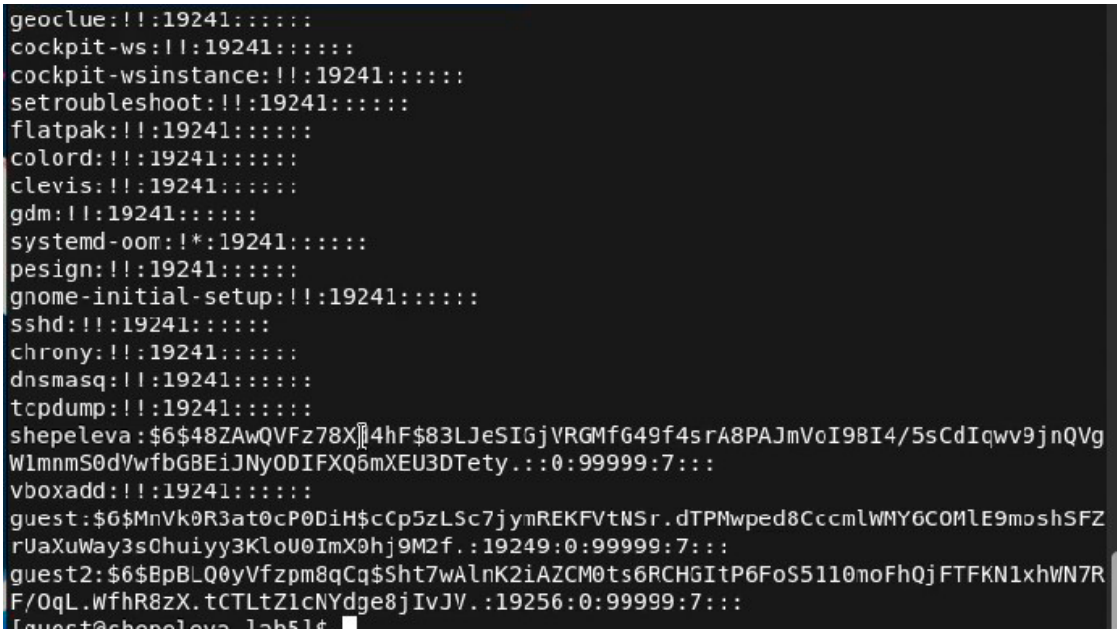
```
[root@shepeleva ~]# chown root:guest /home/guest/lab5/readfile
[root@shepeleva ~]# chmod u+s /home/guest/lab5/readfile
```

readfile

11. Проверила прочитать файл readfile и /etc/shadow

```
[guest@shepeleva lab5]$ ./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;
}
```

readfile read



/etc/shadow read

12. readfile удалось прочитать, а /etc/shadow - нет

13. Проверила sticky бит на категории tmp. Создала файл в tmp от guest и посмотерла атрибуты.

sticky

14. От guest2 попробовала выполнить различные операции

```
[guest@shepeleva tmp]$ su - guest2
Password:
[guest2@shepeleva ~]$ cat /tmp/file01.txt
test
[guest2@shepeleva ~]$ echo "test2" > /tmp/file01.txt
[guest2@shepeleva ~]$ cat /tmp/file01.txt
test2
[guest2@shepeleva ~]$ echo "test3" > /tmp/file01.txt
[guest2@shepeleva ~]$ cat /tmp/file01.txt
test3
[guest2@shepeleva ~]$ rm /tmp/file01.txt
rm: cannot remove '/tmp/file01.txt': Operation not permitted
```

guest2 file01

15. Не удалось выполнить только rm

16. Сняла атрибут t (Sticky-бит) сдиректории /tmp

```
[root@shepeleva ~]# chmod -t /tmp
[root@shepeleva ~]# exit
logout
[guest2@shepeleva ~]$ ls -l / | grep tmp
drwxrwxrwx.  14 root root 4096 Oct  8 16:24 tmp
[guest2@shepeleva ~]$
```

-t

17. Повторила предыдущие шаги.

```
[guest2@shepeleva ~]$ ls -l / | grep tmp
drwxrwxrwx.  14 root root 4096 Oct  8 16:24 tmp
[guest2@shepeleva ~]$ echo "test2" > /tmp/file01.txt
[guest2@shepeleva ~]$ cat /tmp/file01.txt
test2
[guest2@shepeleva ~]$ echo "test3" >> /tmp/file01.txt
[guest2@shepeleva ~]$ cat /tmp/file01.txt
test2
test3
[guest2@shepeleva ~]$ rm /tmp/file01.txt
```

guest2 file01 try 2

Вывод

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

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