
ЛР №5. Исследование влияния дополнительных атрибутов

Информационная безопасность

Доборщук Владимир Владимирович,
НФИбд-01-18

13 ноября 2021

Содержание

Цель работы	4
Теоретическое введение	5
Выполнение лабораторной работы	6
1. Создание программы	6
2. Исследование Sticky-бита	13
Заключение	16

Список иллюстраций

1	simpleid.c	6
2	simpleid2.c	7
3	readfile.c	8
4	Пункты 3-5	8
5	Пункт 7	9
6	Пункты 8-11	10
7	Пункт 12	11
8	Пункты 14-18	12
9	Пункт 19	13
10	Исследование Sticky-бита	14

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.

Задачи:

- изучить механизмы изменения идентификаторов;
- укрепление навыков работы в консоли;
- изучение влияния бита Sticky на запись и удаление файлов.

Теоретическое введение

Для выполнения данной лабораторной работы мы использовали данные источники, в виде описания лабораторной работы, а также свободные источники в интернете.

Выполнение лабораторной работы

1. Создание программы

Зайдя в терминал, мы сделали вошли в систему от лица `guest` и реализовали следующий набор скриптов:

```
#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;
}
```

Рис. 1: `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. 2: simpleid2.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;
}
```

Рис. 3: readfile.c

Все программы скомпилировали, последовательно выполнив пункты:

```
[root@wxcore wdo]# su - guest
Last login: Sat Nov 13 22:46:21 MSK 2021 on pts/0
[guest@wxcore ~]$ vim simpleid.c
[guest@wxcore ~]$ gcc simpleid.c -o simpleid
[guest@wxcore ~]$ ls
dir1 simpleid simpleid.c
[guest@wxcore ~]$ ./simpleid
uid=1001, gid=1001
[guest@wxcore ~]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest)
[guest@wxcore ~]$
```

Рис. 4: Пункты 3-5


```
[root@wxcore wdbo]# su - guest
Last login: Sat Nov 13 22:46:21 MSK 2021 on pts/0
[guest@wxcore ~]$ vim simpleid.c
[guest@wxcore ~]$ gcc simpleid.c -o simpleid
[guest@wxcore ~]$ ls
dir1 simpleid simpleid.c
[guest@wxcore ~]$ ./simpleid
uid=1001, gid=1001
[guest@wxcore ~]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest)
[guest@wxcore ~]$ cp simpleid.c simpleid2.c
[guest@wxcore ~]$ vim simpleid2.c
[guest@wxcore ~]$ gcc simpleid2.c -o simpleid2
[guest@wxcore ~]$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
[guest@wxcore ~]$ |
```

Рис. 5: Пункт 7

```
[guest@wxcore ~]$ exit
logout
[root@wxcore wdo]# chown root:guest /home/guest/simpleid2
[root@wxcore wdo]# chmod u+s /home/guest/simpleid2
[root@wxcore wdo]# ls -l /home/guest/
total 52
drwxrwxrwx 2 guest guest 4096 Oct 30 17:30 dir1
-rwxrwxr-x 1 guest guest 17544 Nov 13 22:51 simpleid
-rwsrwxr-x 1 root  guest 17648 Nov 13 22:54 simpleid2
-rw-rw-r-- 1 guest guest 311 Nov 13 22:54 simpleid2.c
-rw-rw-r-- 1 guest guest 180 Nov 13 22:51 simpleid.c
[root@wxcore wdo]# /home/guest
guest/ guest2/
[root@wxcore wdo]# /home/guest/simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@wxcore wdo]# su - guest
Last login: Sat Nov 13 22:51:44 MSK 2021 on pts/0
[guest@wxcore ~]$ ./simpleid2
e_uid=0, e_gid=1001
real_uid=1001, real_gid=1001
[guest@wxcore ~]$
```

Рис. 6: Пункты 8-11

```
[guest@wxcore ~]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest)
[guest@wxcore ~]$ exit
logout
[root@wxcore wdo]# chmod g+s /home/guest/simpleid2
[root@wxcore wdo]# su - guest
Last login: Sat Nov 13 22:56:39 MSK 2021 on pts/0
[guest@wxcore ~]$ ./simpleid2
e_uid=0, e_gid=1001
real_uid=1001, real_gid=1001
[guest@wxcore ~]$ ls -l
total 52
drwxrwxrwx 2 guest guest 4096 Oct 30 17:30 dir1
-rwxrwxr-x 1 guest guest 17544 Nov 13 22:51 simpleid
-rwsrwsr-x 1 root guest 17648 Nov 13 22:54 simpleid2
-rw-rw-r-- 1 guest guest 311 Nov 13 22:54 simpleid2.c
-rw-rw-r-- 1 guest guest 180 Nov 13 22:51 simpleid.c
[guest@wxcore ~]$
```

Рис. 7: Пункт 12

```
[root@wxcore wdo]# su - guest
Last login: Sat Nov 13 23:01:46 MSK 2021 on pts/0
[guest@wxcore ~]$ ls -l
total 76
drwxrwxrwx 2 guest guest 4096 Oct 30 17:30 dir1
-rwsrwxr-x 1 root guest 17592 Nov 13 23:00 readfile
-rwx----- 1 root guest 417 Nov 13 22:59 readfile.c
-rwxrwxr-x 1 guest guest 17544 Nov 13 22:51 simpleid
-rwsrwsr-x 1 root guest 17648 Nov 13 22:54 simpleid2
-rw-rw-r-- 1 guest guest 311 Nov 13 22:54 simpleid2.c
-rw-rw-r-- 1 guest guest 180 Nov 13 22:51 simpleid.c
[guest@wxcore ~]$ cat readfile.c
cat: readfile.c: Permission denied
[guest@wxcore ~]$ exit
logout
[root@wxcore wdo]# chown root:guest /home/guest/readfile
[root@wxcore wdo]# chmod u+s /home/guest/readfile
[root@wxcore wdo]# ls -l /home/guest/readfile
-rwsrwxr-x 1 root guest 17592 Nov 13 23:00 /home/guest/readfile
[root@wxcore wdo]# su - guest
Last login: Sat Nov 13 23:04:20 MSK 2021 on pts/0
[guest@wxcore ~]$ ./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@wxcore ~]$
```

Рис. 8: Пункты 14-18

```
[guest@wxcore ~]$ ./readfile /etc/shadow
root:!locked::0:99999:7:::
bin:!:18264:0:99999:7:::
daemon:!:18264:0:99999:7:::
adm:!:18264:0:99999:7:::
lp:!:18264:0:99999:7:::
sync:!:18264:0:99999:7:::
shutdown:!:18264:0:99999:7:::
halt:!:18264:0:99999:7:::
mail:!:18264:0:99999:7:::
operator:!:18264:0:99999:7:::
games:!:18264:0:99999:7:::
ftp:!:18264:0:99999:7:::
nobody:!:18264:0:99999:7:::
dbus:!!:18274:::::
systemd-coredump:!!:18274:::::
systemd-resolve:!!:18274:::::
vvdoborschuk:$6$Zq/Dt116HUEZmWkY$JhdgeCN4HHYvuYLf7gZso4gbZ5ma30tCpnNImpKRW8V2KitAPIc51HWJY8x
RJbozKjbhPEQTZkP.ZkDog6H1.:18915:0:99999:7:::
tss:!!:18915:::::
unbound:!!:18915:::::
guest:$6$jxpsM.8t8SsPcb9X$Hk2zU/RwTYaAQvvdIDgaCP3UwLSS5FmwPIG0jZhFkwHA.FmaNkenH7kKJhhH2zKoLmx
GV82Fd.jX3cQwKfx1D/:18916:0:99999:7:::
guest2:$6$ZBkqE3JiSh5XkSHW$6gjIKM8g7qaeL8cZX98uyWV1hmC1s107KO/LdBnxyUEDq7EU7WM8/fk2bRoMKNu76B
LEAWw10HmW02nxM7r201:18916:0:99999:7:::
[guest@wxcore ~]$
```

Рис. 9: Пункт 19

Команды `sudo` и `su` позволяют использовать права суперпользователя, или же использовать его учетную запись.

Как только мы установили SetUID-бит, программа `readfile` получила возможность читать файлы, принадлежащие суперпользователю.

2. Исследование Sticky-бита

Все действия по исследованию Sticky-бита удалось поместить в один скриншот:

```
[guest@wxcore ~]$ ls -l / | grep tmp
drwxrwxrwt  6 root root  4096 Nov 13 23:07 tmp
[guest@wxcore ~]$ echo "test" > /tmp/file01.txt
[guest@wxcore ~]$ ls -l /tmp/file01.txt
-rw-rw-r-- 1 guest guest 5 Nov 13 23:08 /tmp/file01.txt
[guest@wxcore ~]$ chmod o+rw /tmp/file01.txt
[guest@wxcore ~]$ ls -l /tmp/file01.txt
-rw-rw-rw- 1 guest guest 5 Nov 13 23:08 /tmp/file01.txt
[guest@wxcore ~]$ su - guest2
Password:
Last login: Sat Oct 16 22:22:55 MSK 2021 on pts/2
[guest2@wxcore ~]$ cat /tmp/file01.txt
test
[guest2@wxcore ~]$ echo "test2" >> /tmp/file01.txt
[guest2@wxcore ~]$ cat /tmp/file01.txt
test
test2
[guest2@wxcore ~]$ echo "test2" > /tmp/file01.txt
[guest2@wxcore ~]$ cat /tmp/file01.txt
test2
[guest2@wxcore ~]$ echo "test3" > /tmp/file01.txt
[guest2@wxcore ~]$ cat /tmp/file01.txt
test3
[guest2@wxcore ~]$ rm /tmp/file01.txt
rm: cannot remove '/tmp/file01.txt': Operation not permitted
[guest2@wxcore ~]$ su -
Password:
su: Authentication failure
[guest2@wxcore ~]$ exit
logout
[guest@wxcore ~]$ exit
logout
[root@wxcore wdo]# chmod -t /tmp
[root@wxcore wdo]# su - guest2
Last login: Sat Nov 13 23:08:35 MSK 2021 on pts/0
[guest2@wxcore ~]$ ls -l / | grep /tmp
[guest2@wxcore ~]$ ls -l / | grep tmp
drwxrwxrwx  6 root root  4096 Nov 13 23:08 tmp
[guest2@wxcore ~]$ rm /tmp/file01.txt
[guest2@wxcore ~]$ exit
logout
[root@wxcore wdo]# chmod +t /tmp
[root@wxcore wdo]#
```

Рис. 10: Исследование Sticky-бита

В заключении, убрав Sticky-бит, мы смогли удалить файл чужого пользователя.

Заключение

В результате выполнения работы мы изучили механизмы изменения идентификаторов, применения SetUID- и Sticky-битов. Получили практические навыки работы в консоли с дополнительными атрибутами. Рассмотрели работу механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.