

Лабораторная работа 5

**Дискреционное разграничение прав в Linux. Исследование влияния
дополнительных атрибутов**

Лушин Артём Андреевич

Содержание

1	Цель работы	4
2	Выполнение лабораторной работы. Созданте программы	5
3	Исследование Sticky-бита.	13
4	ВАЖНОЕ ПРИМЕЧАНИЕ	16
5	Выводы	17

Список иллюстраций

2.1	Первая программа	5
2.2	Компиляция первой программы	5
2.3	Запуск первой программы	6
2.4	Вторая программа	6
2.5	Запуск второй программы	6
2.6	Изменение прав для root	7
2.7	Проверка работы для root	7
2.8	Установка SetUID-бита	8
2.9	Программа readfile	8
2.10	Компиляция readfile	9
2.11	Проверка на root и guest пользователях	10
2.12	Смена владельца	11
2.13	Запуск с guest	11
2.14	Запуск с root	12
3.1	Проверка наличия атрибута	13
3.2	Выдача прав для файла	13
3.3	Проверка от второго пользователя	14
3.4	Проверка без атрибута	14
3.5	Возвращение атрибута	15

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 Выполнение лабораторной работы.

Создате программы

1) Я создал файл “simpleid.c” и внёс в него программу.

```
1 #include <sys/types.h>
2 #include <unistd.h>
3 #include <stdio.h>
4
5 int
6 main ()
7 {
8     uid_t uid = geteuid ();
9     gid_t gid = getegid ();
10    printf ("uid=%d, gid=%d/n", uid, gid);
11    return 0;
12 }
```

Рис. 2.1: Первая программа

2) Скомпилировал программу и убедился, что файл создан правильно.

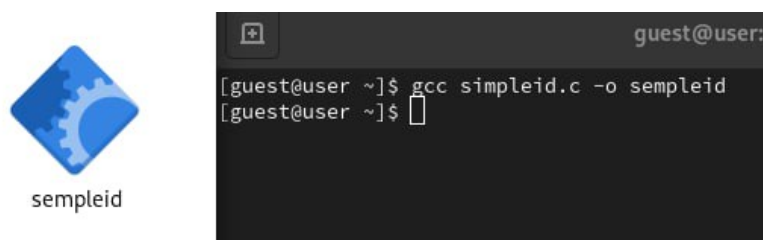


Рис. 2.2: Компиляция первой программы

- 3) Запустил программу и посмотрел, как она работает. Затем прописал команду “id”, чтобы сравнить данные. Все данные сходятся.

```
[guest@user ~]$ ./simpleid
uid=1001, gid=1001/n[guest@user ~]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfi
ned_r:unconfined_t:s0-s0:c0.c1023
[guest@user ~]$
```

Рис. 2.3: Запуск первой программы

- 4) Создал второй файл и назвал его “simpleid2.c”. Усложнил первую программу и внёс ее в файл.

```
1 #include <sys/types.h>
2 #include <unistd.h>
3 #include <stdio.h>
4
5 int
6 main ()
7 {
8     uid_t real_uid = getuid();
9     uid_t e_uid = geteuid ();
10
11     gid_t real_gid = getgid ();
12     gid_t e_gid = getegid ();
13
14     printf ("e_uid=%d, e_gid=%d\n", e_uid, e_gid);
15     printf ("real_uid=%d, real_gid=%d\n", real_uid, real_gid);
16
17     return 0;
18 }
```

Рис. 2.4: Вторая программа

- 5) Скомпилировал и посмотрел вторую программу. Проверил как она работает.

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

Рис. 2.5: Запуск второй программы

- 6) От имени суперпользователя я выполнил команды и временно повысил свои права. Команды сменили пользователя файла на root и установили SetUID-бит. Я запустил файл от имени root-пользователя и проверил сходство с командой “id”.

```
[guest@user ~]$ su
Password:
[root@user guest]# chown root:guest /home/guest/simpleid2
[root@user guest]# chmod u+s /home/guest/simpleid2
[root@user guest]# ls -l
total 64
drwxr-xr-x. 2 guest guest  6 Feb 16 20:57 Desktop
drwxrwx---. 2 guest guest 18 Feb 16 21:22 dir1
drwxr-xr-x. 2 guest guest  6 Feb 16 20:57 Documents
drwxr-xr-x. 2 guest guest  6 Feb 16 20:57 Downloads
drwxr-xr-x. 2 guest guest  6 Feb 16 20:57 Music
drwxr-xr-x. 2 guest guest 53 Feb 17 03:12 Pictures
drwxr-xr-x. 2 guest guest  6 Feb 16 20:57 Public
-rwxr-xr-x. 1 guest guest 25900 Feb 17 03:12 simpleid
-rwsr-xr-x. 1 root  guest 26004 Feb 17 03:17 simpleid2
-rw-r--r--. 1 guest guest  306 Feb 17 03:16 simpleid2.c
-rw-r--r--. 1 guest guest  306 Feb 17 03:16 simpleid.c
drwxr-xr-x. 2 guest guest  6 Feb 16 20:57 Templates
drwxr-xr-x. 2 guest guest  6 Feb 16 20:57 Videos
[root@user guest]# lsattr simpleid2
----- simpleid2
[root@user guest]# ./simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@user guest]# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@user guest]#
```

Рис. 2.6: Изменение прав для root

```
[root@user guest]# su guest
[guest@user ~]$ ls
Desktop dir1 Documents Downloads Music Pictures Public simpleid simpleid2 simpleid2.c simpleid.c Templates Videos
[guest@user ~]$ ./simpleid2
e_uid=0, e_gid=1001
real_uid=1001, real_gid=1001
[guest@user ~]$ su root
Password:
[root@user guest]# ./simpleid
bash: ./simpleid: No such file or directory
[root@user guest]# ./simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@user guest]# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@user guest]#
```

Рис. 2.7: Проверка работы для root

```

root@user guest]# chmod g+s /home/guest/simpleid2
root@user guest]# ./simpleid2
a_uid=0, e_gid=1001
real_uid=0, real_gid=0
root@user guest]# su guest
guest@user ~]$ ./simpleid2
a_uid=0, e_gid=1001
real_uid=1001, real_gid=1001
guest@user ~]$ ls
bash: ls: command not found...
guest@user ~]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
guest@user ~]$ su
password:
root@user guest]# ./simpleid2
a_uid=0, e_gid=1001
real_uid=0, real_gid=0
root@user guest]# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
root@user guest]#

```

Рис. 2.8: Установка SetUID-бита

7) Я создал файл “readfile.c”. Внёс туда программу.

```

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. 2.9: Программа readfile

8) Скомпилировал программу readfile.


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

Рис. 2.10: Компиляция readfile

- 9) Я выдал программе “readfile” права так, чтобы root пользователь мог прочитать файл, а простой пользователь нет.


```

[guest@user ~]$ su root
Password:
[root@user guest]# chown root:guest /home/guest/readfile
[root@user guest]# chmod g+s /home/guest/readfile
[root@user guest]# ls -l
total 96
drwxr-xr-x. 2 guest guest    6 Feb 16 20:57 Desktop
drwxrwx---. 2 guest guest   18 Feb 16 21:22 dir1
drwxr-xr-x. 2 guest guest    6 Feb 16 20:57 Documents
drwxr-xr-x. 2 guest guest    6 Feb 16 20:57 Downloads
drwxr-xr-x. 2 guest guest    6 Feb 16 20:57 Music
drwxr-xr-x. 2 guest guest  100 Feb 17 03:30 Pictures
drwxr-xr-x. 2 guest guest    6 Feb 16 20:57 Public
-rwx--S---. 1 root  guest 26008 Feb 17 03:27 readfile

```

Рис. 2.12: Смена владельца

- 11) Попытался запустить программу и прочитать два файла с простого пользователя, но программа выдала ошибку. А если запускать с аккаунта root, то программа запускается нормально и работает. Связано это с тем, что владельцем программы является root-пользователь, а у других пользователей нет доступа и прав на использование программы.

```

[guest@user ~]$ ./readfile readfile.c
bash: ./readfile: Permission denied
[guest@user ~]$ ./readfile /etc/shadow
bash: ./readfile: Permission denied
[guest@user ~]$

```

Рис. 2.13: Запуск с guest

```

[guest@user ~]$ su root
Password:
[root@user guest]# ./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;
    }

    [root@user guest]# ./readfile /etc/shadow
root:$6$z9VjQ038u..kG8BU54CqQLENVVgQMPK.VhTmv59AaAFHtsKs5qJ7tvyrlEWYcCdaW.981bumGYhu4o3JyE16Xkt.aRF/23ZhcZES51::0:99999:7:::
bin::19469:0:99999:7:::
daemon::19469:0:99999:7:::
adm::19469:0:99999:7:::
lp::19469:0:99999:7:::
sync::19469:0:99999:7:::
shutdown::19469:0:99999:7:::
halt::19469:0:99999:7:::
mail::19469:0:99999:7:::
operator::19469:0:99999:7:::
games::19469:0:99999:7:::
ftp::19469:0:99999:7:::
nobody::19469:0:99999:7:::
system-coredump::19766:::
dbus::19766:::
polkitd::19766:::
avahi::19766:::
rtkit::19766:::
pipewire::19766:::
sssd::19766:::
libstoragemgmt::19766:::
systemd-oom::19766:::
cups::19766:::
geoclue::19766:::
cockpit-ws::19766:::
cockpit-ws/instances::19766:::
flatpak::19766:::
colord::19766:::
clefts::19766:::
setroubleshoot::19766:::
gdm::19766:::
pesign::19766:::
gnome-initial-setup::19766:::
ssh::19766:::
chrony::19766:::
dnsmasq::19766:::
tcpdump::19766:::
aatus-fhm:65507QMA2edcSVr4u56L3/jgk2qHCsGyAHxb11ZENV5ELKPFVTuq6Ldhr fHRee fWswBwmHlJ3xnlK4R4JzQNrRf1eKfUNn6FCUBb10y.:10:99999:7:::
vboxadd::19767:::
guest:465JRGWwYJ805Ut6TetoOP1j8t..IfxL3mNSXHVqdmuZwh1E..4r3p4u3LxvKthchuqSAEeEfLOCqAl8p9r7..9FYq6VrphTbMazuZkRA.:19768:0:99999:7:::
guest2:56Gfs3Lw474c0gduK5i82gQ2adob6gwb/rq6G6fGrXaCpfD2lHg6p4nPhjmbGbrwHJLxLDWvyLeqAthcyCFmpR7f8jzh2fnnw/vry1.:19768:0:99999:7:::
[root@user guest]#

```

Рис. 2.14: Запуск с root

3 Исследование Sticky-бита.

- 1) Я выяснил, установлен ли атрибут Sticky (t) на директории “/tmp”. Атрибут установлен.

```
[guest@user ~]$ ls -l / | grep tmp
drwxrwxrwt. 17 root root 4096 Feb 17 03:36 tmp
[guest@user ~]$
```

Рис. 3.1: Проверка наличия атрибута

- 2) От пользователя “guest” я создал файл “file01.txt” в директории “/tmp”. Вписал в файл слово “test”. И дал права на чтение и запись для категории “все остальные (o)”.

```
[guest@user ~]$ echo "test" > /tmp/file01.txt
[guest@user ~]$ ls -l /tmp/file01.txt
-rw-r--r--. 1 guest guest 5 Feb 17 03:38 /tmp/file01.txt
[guest@user ~]$ chmod 0+rw /tmp/file01.txt
chmod: invalid mode: '0+rw'
Try 'chmod --help' for more information.
[guest@user ~]$ chmod o+rw /tmp/file01.txt
[guest@user ~]$ ls -l /tmp/file01.txt
-rw-r--rw-. 1 guest guest 5 Feb 17 03:38 /tmp/file01.txt
[guest@user ~]$
```

Рис. 3.2: Выдача прав для файла

- 3) От пользователя “guest2”, который не является владельцем, я попробовал прочитать файл. Я могу прочитать файл. Но не могу дописывать содержимое, вписывать новое или удалять этот файл.

```
[guest@user ~]$ su guest2
Password:
[guest2@user guest]$ cat /tmp/file01.txt
test
[guest2@user guest]$ echo "test2" >> /tmp/file01.txt
bash: /tmp/file01.txt: Permission denied
[guest2@user guest]$ echo "test2" >> /tmp/file01.txt
bash: /tmp/file01.txt: Permission denied
[guest2@user guest]$ cat /tmp/file01.txt
test
[guest2@user guest]$ echo "test3" > /tmp/file01.txt
bash: /tmp/file01.txt: Permission denied
[guest2@user guest]$ rm /tmp/file01.txt
rm: remove write-protected regular file '/tmp/file01.txt'? y
rm: cannot remove '/tmp/file01.txt': Operation not permitted
[guest2@user guest]$
```

Рис. 3.3: Проверка от второго пользователя

- 4) я отключил атрибут “t” у директории “/tmp”. Попробовал повторить все предыдущие действия. Я так же не смог вписать в файл данные или дописать их. Но смог прочитать файл и удалить его.

```
[guest2@user tmp]$ ls
file01.txt
[guest2@user tmp]$ echo "test2" >> file01.txt
bash: file01.txt: Permission denied
[guest2@user tmp]$ echo "test2" >> file01.txt
bash: file01.txt: Permission denied
[guest2@user tmp]$ cat file01.txt
test
[guest2@user tmp]$ rm file01.txt
rm: remove write-protected regular file 'file01.txt'? y
[guest2@user tmp]$ ls
[guest2@user tmp]$
```

Рис. 3.4: Проверка без атрибута

- 5) Чтобы в дальнейшем у меня не было проблем в работе с директорией “/tmp” я вернул атрибут на директорию, используя суперпользователя.

```
guest2@user tmp]$ su -  
Password:  
[root@user ~]# chmod +t /tmp  
[root@user ~]# exit  
logout  
guest2@user tmp]$ ls -l / | grep tmp  
drwxrwxrwt. 19 root root 4096 Feb 17 03:56 tmp  
guest2@user tmp]$
```

Рис. 3.5: Возвращение атрибута

4 ВАЖНОЕ ПРИМЕЧАНИЕ

По итогам лабораторной работы я понял, что Sticky-бит создан для защиты файла от удаления. Даже не смотря на то, что я дал права на запись и чтение файлов для категории “все остальные”, я не смог вписать в файл данные с пользователя “guest2”. А не смог я это сделать, так как этот аккаунт у меня находится в группе с “guest”. То есть я не дал права на вписывание для категории “группа”, но дал права для категории “все остальные”. Из-за этого Sticky-бит не влиял на возможность записи, а влиял только на возможность удаления. А изменять файл я не мог, так как мой аккаунт находился не в той группе. Если бы я использовал другой аккаунт, который не находится в группе, результаты бы были другие.

5 Выводы

Я изучил механизмы изменения идентификатора, применил SetUID-бит и Sticky-бит. Получил практические навыки работы в консоли с дополнительными атрибутами. Рассмотрел работы механизма смены идентификатора процессов пользователя, а так же влияние бита Sticky на запись и удаление файлов.