

Лабораторная работа №5

Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов.

Дугаева Светлана Анатольевна

Содержание

Цель работы	4
Выполнение лабораторной работы	5
Исследование Sticky-бита	12
Выводы	15

Список иллюстраций

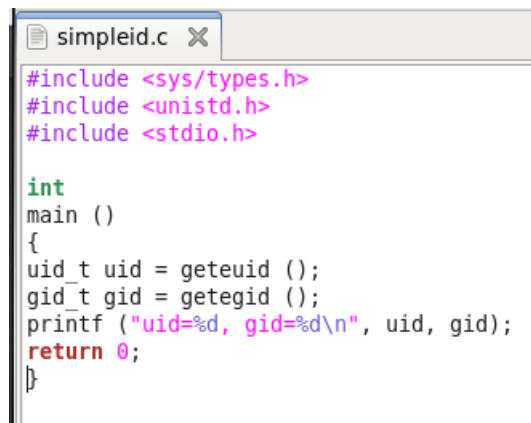
0.1	Программа simpleid.c	5
0.2	Программа simpleid2.c	6
0.3	Пункты 3-5 и 7	6
0.4	Пункты 8-12	7
0.5	Программа readfile.c	8
0.6	Пункты 14-17	9
0.7	Чтение файла readfile.c	10
0.8	Чтение файла /etc/shadow	11
0.1	Пункты 1-11	13
0.2	Пункты 12-13	14
0.3	Установка атрибута t	14

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.

Выполнение лабораторной работы

1. Вошла в систему от имени пользователя guest.
2. Создала программу simpleid.c (рис. @fig:001):



```
#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. 0.1: Программа simpleid.c

3. Скомпилировала программу и убедилась, что файл программы создан.
4. Выполнила программу simpleid.
5. Выполните системную программу id, выведенные данные совпадают(id выводит больше информации) .
6. Усложнила программу, добавив вывод действительных идентификаторов и назвала её simpleid2.c (рис. @fig:002):

```
simpleid2.c X
#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. 0.2: Программа simpleid2.c

7. Скомпилировала и запустила simpleid2.c

Действия из пунктов 3-5 и 7 приведены на (рис. @fig:003):

```
[guest@localhost ~]$ gcc simpleid.c -o simpleid
[guest@localhost ~]$ ls
simpleid  simpleid.c~  Документы  Картинки  Общедоступные  Шаблоны
file2  simpleid.c  Видео      Загрузки  Музыка         Рабочий стол
[guest@localhost ~]$ ./simpleid
uid=501, gid=501
[guest@localhost ~]$ id
uid=501(guest) gid=501(guest) группы=501(guest) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@localhost ~]$ rm simpleid
[guest@localhost ~]$ ls
simpleid.c  Видео      Загрузки  Музыка         Рабочий стол
file2  simpleid.c~  Документы  Картинки  Общедоступные  Шаблоны
[guest@localhost ~]$ gcc simpleid.c -o simpleid
[guest@localhost ~]$ ./simpleid
e_uid=501, e_gid=501
real_uid=501, real_gid=501
[guest@localhost ~]$ gcc simpleid2.c -o simpleid2
[guest@localhost ~]$ ./simpleid2
e_uid=501, e_gid=501
real_uid=501, real_gid=501
[guest@localhost ~]$ su
```

Рис. 0.3: Пункты 3-5 и 7

8. От имени суперпользователя выполнила команды по смене владельца и изме-

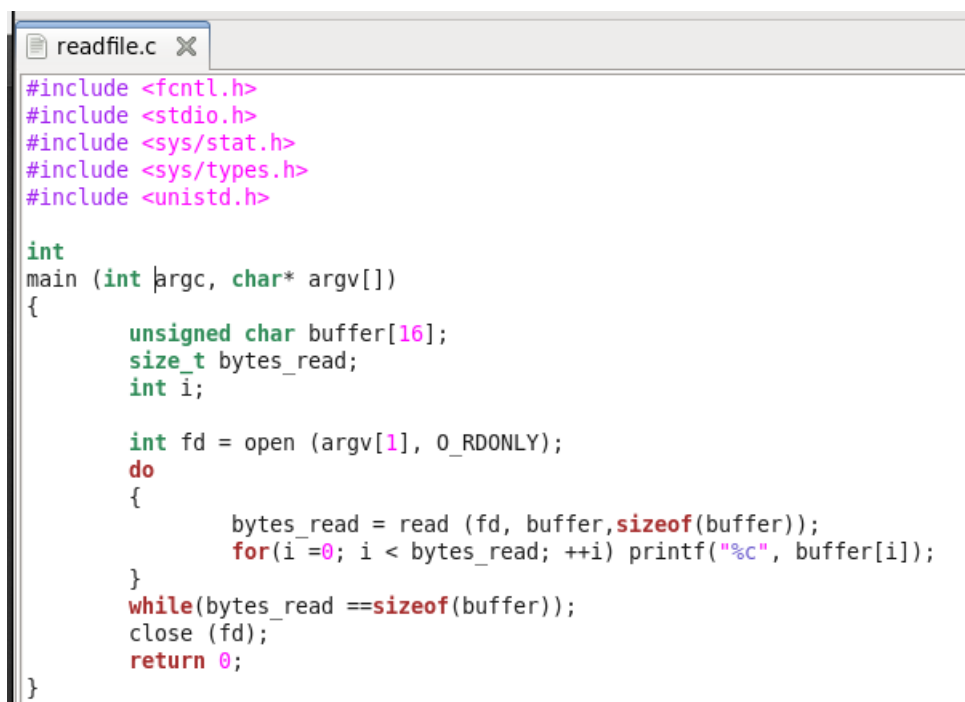
- нению прав на файл simpleid2.
9. Временно повысила свои права с помощью su. Команда chown позволяет изменить владельца файла, а команда chmod позволяет поменять права на файл.
 10. Выполнила проверку правильности установки новых атрибутов и смены владельца файла simpleid2.
 11. Запустила simpleid2 и id. Выведенные результаты не совпадают, т.к. мы уже изменили владельца файла на суперпользователя.
 12. Прodelала тоже самое относительно SetGID-бита. В этот раз данные полностью совпали.

Действия из пунктов 8-12 приведены на (рис. @fig:004):

```
[root@localhost guest]# chown root:guest /home/guest/simpleid2
[root@localhost guest]# chmod u+s /home/guest/simpleid2
[root@localhost guest]# ls -l simpleid2
-rwsrwxr-x. 1 root guest 5143 Ноя 10 18:29 simpleid2
[root@localhost guest]# su guest
[guest@localhost ~]$ ./simpleid2
e_uid=0, e_gid=501
real_uid=501, real_gid=501
[guest@localhost ~]$ id
uid=501(guest) gid=501(guest) группы=501(guest) контекст=unconfined_u:unconfined
_r:unconfined_t:s0-s0:c0.c1023
[guest@localhost ~]$ su
Пароль:
[root@localhost guest]# chmod u-s /home/guest/simpleid2
[root@localhost guest]# chmod g+s /home/guest/simpleid2
[root@localhost guest]# su guest
[guest@localhost ~]$ ls -l simpleid2
-rwxrwsr-x. 1 root guest 5143 Ноя 10 18:29 simpleid2
[guest@localhost ~]$ ./simpleid2
e_uid=501, e_gid=501
real_uid=501, real_gid=501
[guest@localhost ~]$ id
uid=501(guest) gid=501(guest) группы=501(guest) контекст=unconfined_u:unconfined
_r:unconfined_t:s0-s0:c0.c1023
```

Рис. 0.4: Пункты 8-12

13. Создала программу readfile.c (рис. @fig:005):



```
readfile.c X
#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. 0.5: Программа readfile.c

14. Откомпилировала её.
15. Сменила владельца у файла readfile.c (или любого другого текстового файла в системе) и изменила права так, чтобы только суперпользователь (root) мог прочитать его, а guest не мог.
16. Проверила, что пользователь guest не может прочитать файл readfile.c.
17. Сменила у программы readfile владельца и установила SetUID-бит.

Действия из пунктов 14-17 приведены на (рис. @fig:006):


```
[guest@localhost ~]$ gcc readfile.c -o readfile
readfile.c:7: ошибка: expected ')' before 'char'
[guest@localhost ~]$ gcc readfile.c -o readfile
[guest@localhost ~]$ su
Пароль:
[root@localhost guest]# chown root:guest /home/guest/readfile
[root@localhost guest]# chmod 700 /home/guest/readfile.c
[root@localhost guest]# chown root:guest /home/guest/readfile.c
[root@localhost guest]# su guest
[guest@localhost ~]$ cat readfile.c
cat: readfile.c: Отказано в доступе
[guest@localhost ~]$ su
Пароль:
[root@localhost guest]# chown root:guest /home/guest/readfile
[root@localhost guest]# chmod u+s /home/guest/readfile
[root@localhost guest]# su guest
[guest@localhost ~]$ ./readfile readfile.c
#include <fcntl.h>
#include <stdio.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>

int
```

Рис. 0.6: Пункты 14-17

18. Теперь программа readfile может прочитать файл readfile.c (рис. @fig:007):

```

[root@localhost guest]# chmod u+s /home/guest/readfile
[root@localhost guest]# su guest
[guest@localhost ~]$ ./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;
}

```

Рис. 0.7: Чтение файла readfile.c

19. Также программа readfile может прочитать файл /etc/shadow. Это связано с тем, что мы установили SetUID-бит, и соответственно дали ей права владельца файла(суперпользователя) (рис. @fig:008):

```
[guest@localhost ~]$ ./readfile /etc/shadow
root:$6$7MKhNKHhY6IT6S5x$AGQwN2UXLHtBWw5tKkNtutWx0a/hqu5TjKm29xcDXn6CbVsXsyoN8hH
CVLRhpDoZQsZft/vGEguJJZfoNdYcsN.:18888:0:99999:7:::
bin:!:15980:0:99999:7:::
daemon:!:15980:0:99999:7:::
adm:!:15980:0:99999:7:::
lp:!:15980:0:99999:7:::
sync:!:15980:0:99999:7:::
shutdown:!:15980:0:99999:7:::
halt:!:15980:0:99999:7:::
mail:!:15980:0:99999:7:::
uucp:!:15980:0:99999:7:::
operator:!:15980:0:99999:7:::
games:!:15980:0:99999:7:::
gopher:!:15980:0:99999:7:::
ftp:!:15980:0:99999:7:::
nobody:!:15980:0:99999:7:::
dbus:!!:16654:::::::
vcsa:!!:16654:::::::
rtkit:!!:16654:::::::
avahi-autoipd:!!:16654:::::::
pulse:!!:16654:::::::
ssslauth:!!:16654:::::::
ntp:!!:16654:::::::
haldaemon:!!:16654:::::::
postfix:!!:16654:::::::
gdm:!!:16654:::::::
sshd:!!:16654:::::::
tcpdump:!!:16654:::::::
```

Рис. 0.8: Чтение файла /etc/shadow

Исследование Sticky-бита

1. Выяснила, установлен ли атрибут Sticky на директории /tmp.
2. От имени пользователя guest создала файл file01.txt в директории /tmp со словом test.
3. Просмотрела атрибуты у только что созданного файла и разрешила чтение и запись для категории пользователей «все остальные».
4. От пользователя guest2 (не являющегося владельцем) смогла прочитать файл /tmp/file01.txt.
5. От пользователя guest2 дозаписала в файл /tmp/file01.txt слово test2.
6. Проверила содержимое файла.
7. От пользователя guest2 записала в файл /tmp/file01.txt слово test3, стерев при этом всю имеющуюся в файле информацию.
8. Проверила содержимое файла.
9. От пользователя guest2 не смогла удалить файл /tmp/file01.txt.
10. Повысила свои права до суперпользователя и выполнила после этого команду, снимающую атрибут t (Sticky-бит) с директории /tmp.
11. Покинула режим суперпользователя командой exit.

Действия из пунктов 1-11 приведены на (рис. @fig:009):

```

[guest@localhost ~]$ ls -l / | grep tmp
drwxrwxrwt. 29 root root 4096 Ноя 10 18:54 tmp
[guest@localhost ~]$ echo "test" > /tmp/file01.txt
[guest@localhost ~]$ ls -l /tmp/fole01.txt
ls: невозможно получить доступ к /tmp/fole01.txt: Нет такого файла или каталога
[guest@localhost ~]$ ls -l /tmp/file01.txt
-rw-rw-r--. 1 guest guest 9 Ноя 10 19:13 /tmp/file01.txt
[guest@localhost ~]$ chmod o+rw /tmp/file01.txt
[guest@localhost ~]$ ls -l /tmp/file01.txt
-rw-rw-rw-. 1 guest guest 9 Ноя 10 19:13 /tmp/file01.txt
[guest@localhost ~]$ su guest2
Пароль:
[guest2@localhost guest]$ cat /tmp/file01.txt
"test"
[guest2@localhost guest]$ echo "test2" >> /tmp/file01.txt
[guest2@localhost guest]$ cat /tmp/file01.txt
"test"
"test2"
[guest2@localhost guest]$ echo "tetst3" > /tmp/file01.txt
[guest2@localhost guest]$ cat /tmp/file01.txt
"tetst3"
[guest2@localhost guest]$ rm /tmp/file01.txt
rm: невозможно удалить «/tmp/file01.txt»: Операция не допускается
[guest2@localhost guest]$ su
Пароль:
[root@localhost guest]# chmod -t /tmp
[root@localhost guest]# exit
exit
[guest2@localhost guest]$ ls -l / | grep tmp

```

Рис. 0.1: Пункты 1-11

12. От пользователя guest2 проверила, что атрибута t у директории /tmp нет.
13. Повторила предыдущие шаги. Удалось выполнить все действия, в том числе и удаление файла.

Действия из пунктов 12-13 приведены на (рис. @fig:010):

```

[guest2@localhost guest]$ ls -l / | grep tmp
drwxrwxrwx. 29 root root 4096 Ноя 10 19:13 tmp
[guest2@localhost guest]$ echo "test" > /tmp/file01.txt
[guest2@localhost guest]$ cat /tmp/file1.txt
cat: /tmp/file1.txt: Нет такого файла или каталога
[guest2@localhost guest]$ cat /tmp/file01.txt
"test"
[guest2@localhost guest]$ echo "test" >> /tmp/file01.txt
[guest2@localhost guest]$ cat /tmp/file01.txt
"test"
"test"
[guest2@localhost guest]$ echo "test" >> /tmp/file02.txt
[guest2@localhost guest]$ cat /tmp/file02.txt
"test"
[guest2@localhost guest]$ rm /tmp/file02.txt
[guest2@localhost guest]$ ls /tmp
file01.txt      keyring-MGNCf0  orbit-gdm
gconfd-gdm     keyring-oBL7Ip  orbit-guest
gconfd-guest   keyring-OLChHR  pulse-1T2833879De3
keyring-3UUToa keyring-rGLVXU  pulse-80TnUafBh1EY
keyring-5KS5yA keyring-TUj35m  pulse-BDdk4lU8402N
keyring-90PLMU keyring-VuW3dm  pulse-uLV0NTNTB802
keyring-BadZjb keyring-YY3pCY  Temp-4ab2cea4-76d0-48e3-ae99-97895506eec9
keyring-dkFbFs keyring-ZTT25Z  Temp-ec754bcc-a6bd-4a35-977c-68e7d954127d
[guest2@localhost guest]$ rm /tmp/file01.txt
[guest2@localhost guest]$ ls /tmp
gconfd-gdm     keyring-oBL7Ip  orbit-guest
gconfd-guest   keyring-OLChHR  pulse-1T2833879De3
keyring-3UUToa keyring-rGLVXU  pulse-80TnUafBh1EY

```

Рис. 0.2: Пункты 12-13

14. Повысила свои права до суперпользователя и вернула атрибут `t` на директорию `/tmp`. (рис. @fig:011):

```

[guest2@localhost guest]$ rm /tmp/file01.txt
[guest2@localhost guest]$ ls /tmp
gconfd-gdm     keyring-oBL7Ip  orbit-guest
gconfd-guest   keyring-OLChHR  pulse-1T2833879De3
keyring-3UUToa keyring-rGLVXU  pulse-80TnUafBh1EY
keyring-5KS5yA keyring-TUj35m  pulse-BDdk4lU8402N
keyring-90PLMU keyring-VuW3dm  pulse-uLV0NTNTB802
keyring-BadZjb keyring-YY3pCY  Temp-4ab2cea4-76d0-48e3-ae99-97895506eec9
keyring-dkFbFs keyring-ZTT25Z  Temp-ec754bcc-a6bd-4a35-977c-68e7d954127d
keyring-MGNCf0 orbit-gdm
[guest2@localhost guest]$ su
Пароль:
[root@localhost guest]# chmod +t /tmp
[root@localhost guest]# exit
exit

```

Рис. 0.3: Установка атрибута `t`

Выводы

Изучила механизмы изменения идентификаторов, применения SetUID- и Sticky-битов. Получила практические навыки работы в консоли с дополнительными атрибутами. Рассмотрела работу механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.