

Отчет по лабораторной работе №5

Дискреционное разграничение прав в Linux. Расширенные атрибуты

Низамова Альфия Айдаровна

Содержание

1	Цель работы	5
2	Выполнение	6
2.1	Создание программы	6
2.2	Исследование Sticky-бита	10
3	Вывод	14

Список иллюстраций

Рис. 1	6
Рис. 2	6
Рис. 3	7
Рис. 4	7
Рис. 5	8
Рис. 6	8
Рис. 7	8
Рис. 8	9
Рис. 9	9
Рис. 10	10
Рис. 11	10
Рис. 12	11
Рис. 13	11
Рис. 14	12
Рис. 15	12
Рис. 16	12
Рис. 17	13

Список таблиц

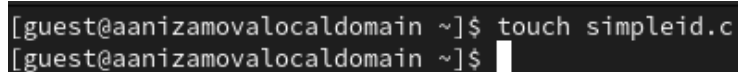
1 Цель работы

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

2.1 Создание программы

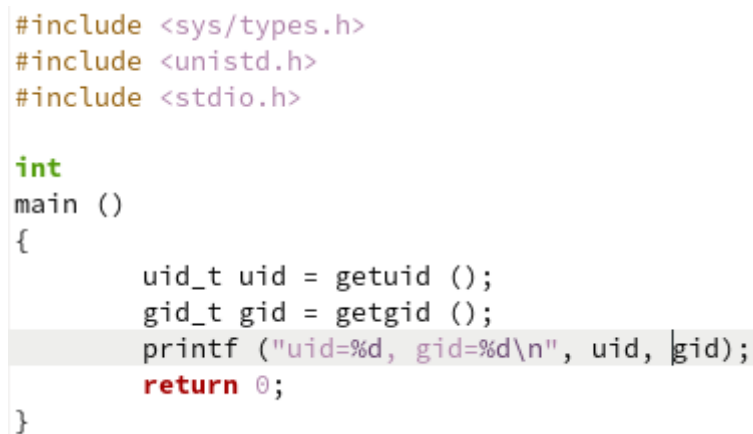
1. Войдем в систему от имени пользователя guest (рис.1)



```
[guest@aanizamovalocaldomain ~]$ touch simpleid.c
[guest@aanizamovalocaldomain ~]$
```

Рис. 1

2. Создадим программу simpleid.c (рис.2)



```
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>

int
main ()
{
    uid_t uid = getuid ();
    gid_t gid = getgid ();
    printf ("uid=%d, gid=%d\n", uid, gid);
    return 0;
}
```

Рис. 2

3. Скомпилируем программу и убедимся, что файл программы создан (рис.3)
4. Выполним программу simpleid (рис.3)

5. Выполним системную программу `id` и сравним полученный результат с данными предыдущего пункта задания (рис.3)

```
[guest@aanizamovalocaldomain ~]$ gcc simpleid.c
[guest@aanizamovalocaldomain ~]$ ./simpleid
uid=1001, gid=1001
[guest@aanizamovalocaldomain ~]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Рис. 3

6. Усложним программу, добавив вывод действительных идентификаторов. Получившуюся программу назовем `simpleid2.c` (рис.4)

```
#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. 4

7. Скомпилируем и запустим `simpleid2.c` (рис.5)
8. От имени суперпользователя выполним заданные команды (рис.5)
9. Используем `sudo` или повысим временно свои права с помощью `su` (рис.5)

```
[guest@aanizamovalocaldomain ~]$ gcc simpleid2.c -o simpleid2
[guest@aanizamovalocaldomain ~]$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
[guest@aanizamovalocaldomain ~]$ su root
Password:
[root@aanizamovalocaldomain guest]# chown root:guest /home/guest/simpleid2
[root@aanizamovalocaldomain guest]# chmod u+s /home/guest/simpleid2
```

Рис. 5

10. Выполним проверку правильности установки новых атрибутов и смены владельца файла simpleid2 (рис.6)

11. Запустим simpleid2 и id. Сравним результаты. (рис.6)

```
[guest@aanizamovalocaldomain ~]$ su root
Password:
[root@aanizamovalocaldomain guest]# ls -l simpleid2
-rwsr-xr-x. 1 root guest 26064 Oct  5 02:14 simpleid2
[root@aanizamovalocaldomain guest]# ./simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@aanizamovalocaldomain guest]# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfi
ned_t:s0-s0:c0.c1023
```

Рис. 6

12. Проделаем то же самое относительно SetGID-бита (рис.7)

```
[root@aanizamovalocaldomain guest]# su guest
[guest@aanizamovalocaldomain ~]$ ./simpleid2
e_uid=0, e_gid=1001
real_uid=1001, real_gid=1001
[guest@aanizamovalocaldomain ~]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfi
ned_r:unconfined_t:s0-s0:c0.c1023
[guest@aanizamovalocaldomain ~]$ su root
Password:
[root@aanizamovalocaldomain guest]# chmod g+s /home/guest/simpleid2
[root@aanizamovalocaldomain guest]# su guest
[guest@aanizamovalocaldomain ~]$ ./simpleid2
e_uid=0, e_gid=1001
real_uid=1001, real_gid=1001
```

Рис. 7

13. Создадим программу readfile.c (рис.8)

```
#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[i], 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. 8

14. Откомпилируем её (рис.9)

15. Сменим владельца у файла readfile.c и изменим права так, чтобы только суперпользователь (root) мог прочитать его, а guest не мог (рис.9)

16. Проверим, что пользователь guest не может прочитать файл readfile.c (рис.9)

```
[guest@aanizamovalocaldomain ~]$ gcc readfile.c -o readfile
[guest@aanizamovalocaldomain ~]$ su root
Password:
[root@aanizamovalocaldomain guest]# chown root /home/guest/readfile.c
[root@aanizamovalocaldomain guest]# chmod 700 /home/guest/readfile.c
[root@aanizamovalocaldomain guest]# su guest
[guest@aanizamovalocaldomain ~]$ cat readfile.c
cat: readfile.c: Permission denied
```

Рис. 9


```
[guest@aanizamovalocaldomain ~]$ ls -l / | grep tmp
drwxrwxrwt. 17 root root 4096 Oct  5 02:48 tmp
[guest@aanizamovalocaldomain ~]$ echo "test" > /tmp/file01.txt
[guest@aanizamovalocaldomain ~]$ ls -l /tmp/file01.txt
-rw-r--rw-. 1 guest guest 5 Oct  5 02:53 /tmp/file01.txt
[guest@aanizamovalocaldomain ~]$ chmod o+rw /tmp/file01.txt
[guest@aanizamovalocaldomain ~]$ ls -l /tmp/file01.txt
-rw-r--rw-. 1 guest guest 5 Oct  5 02:53 /tmp/file01.txt
[guest@aanizamovalocaldomain ~]$ su guest2
```

Рис. 12

4. От пользователя guest2 (не являющегося владельцем) попробуем прочитать файл /tmp/file01.txt (рис.13)
5. От пользователя guest2 попробуем дозаписать в файл /tmp/file01.txt слово test2. Выполнить операцию не удалось (рис.13)
6. Проверим содержимое файла командой (рис.13)

```
[guest@aanizamovalocaldomain ~]$ su guest2
Password:
[guest2@aanizamovalocaldomain guest]$ cat /tmp/file01.txt
test
[guest2@aanizamovalocaldomain guest]$ echo "test2" >> /tmp/file01.txt
bash: /tmp/file01.txt: Permission denied
[guest2@aanizamovalocaldomain guest]$ cat /tmp/file01.txt
test
```

Рис. 13

7. От пользователя guest2 попробуем записать в файл /tmp/file01.txt слово test3, стерев при этом всю имеющуюся в файле информацию. Выполнить операцию не удалось (рис.14)
8. Проверим содержимое файла командой (рис.14)
9. От пользователя guest2 попробуем удалить файл /tmp/file01.txt. Удалить файл не удалось (рис.14)

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

Рис. 14

10. Повысим свои права до суперпользователя и выполним после этого команду, снимающую атрибут t (Sticky-бит) с директории /tmp (рис.15)

11. Покинем режим суперпользователя (рис.15)

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

Рис. 15

12. От пользователя guest2 проверим, что атрибута t у директории /tmp нет (рис.16)

13. Повторим предыдущие шаги. Удалось удалить файл (рис.16)

14. удалить файл от имени пользователя, не являющегося его владельцем удалось. (рис.16)

```
[guest2@aanizamovalocaldomain root]$ ls -l / | grep tmp
drwxrwxrwx. 17 root root 4096 Oct  5 03:05 tmp
[guest2@aanizamovalocaldomain root]$ cat /tmp/file01.txt
test
[guest2@aanizamovalocaldomain root]$ echo "test2" >> /tmp/file01.txt
bash: /tmp/file01.txt: Permission denied
[guest2@aanizamovalocaldomain root]$ echo "test3" > /tmp/file01.txt
bash: /tmp/file01.txt: Permission denied
[guest2@aanizamovalocaldomain root]$ rm /tmp/file01.txt
rm: remove write-protected regular file '/tmp/file01.txt'? y
```

Рис. 16

15. Повысим свои права до суперпользователя и вернем атрибут t на директорию /tmp (рис.17)

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

Рис. 17

3 Вывод

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