

Лабораторная работа №5

Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

Тарутин Кристина Олеговна

Содержание

1	Цель работы	1
2	Выполнение лабораторной работы.....	1
3	Выводы	1
	Список литературы	6

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

1.. Войдите в систему от имени пользователя guest. 2. Создайте программу simpleid.c(рис. [-@fig:001]).

```
simpleid.c [-M--] 1 L:[ 1+11 12/ 12] *(175 / 175b) <EOF> [*] [X]
#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;
}
```

{#fig:001 width=70%}

3. Скомпилируйте программу и убедитесь, что файл программы создан: `gcc simpleid.c -o simpleid`
4. Выполните программу `simpleid`: `./simpleid`
5. Выполните системную программу `id`: `id` и сравните полученный вами результат с данными предыдущего пункта задания (рис. [-@fig:002]).

```
[guest@user ~]$ gcc simpleid.c -o simpleid
[guest@user ~]$ ./simpleid
uid=1001, gid=1001
[guest@user ~]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfi
ned_r:unconfined_t:s0-s0:c0.c1023
[guest@user ~]$
```

{#fig:002 width=70%}

6. Усложните программу, добавив вывод действительных идентификато- ров. Получившуюся программу назовите `simpleid2.c`(рис. [-@fig:003]).

```
simpleid2.c  [----]  1 L:[  1+17  18/ 18] *(296 / 296b) <EOF>  [*] [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, egid=%d\n", e_uid, e_gid);
    printf ("real_uid=%d, real_gid=%d\n", real_uid, real_gid);

    return 0;
}
```

{#fig:003 width=70%}

7. Скомпилируйте и запустите `simpleid2.c`: `gcc simpleid2.c -o simpleid2 ./simpleid2` (рис. [-@fig:004]).

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

{#fig:004 width=70%} 8. От имени суперпользователя выполните команды `chown root:guest /home/guest/simpleid2` `chmod u+s /home/guest/simpleid2` 9. Используйте

sudo или повысьте временно свои права с помощью su. Поясните, что делают эти команды. 10. Выполните проверку правильности установки новых атрибутов и смены владельца файла simpleid2: ls -l simpleid2 11. Запустите simpleid2 и id: ./simpleid2 id Сравните результаты. 12. Прodelайте тоже самое относительно SetGID-бита (рис. [-@fig:005]).

```
[root@user guest]# chown root:guest /home/guest/simpleid2
[root@user guest]# chmod u+s /home/guest/simpleid2
[root@user guest]# ls -l simpleid2
-rwxr-xr-x. 1 guest guest 25960 apr 13 22:20 simpleid2
[root@user guest]# ./simpleid2
e_uid=0, egid=0
real_uid=0, real_gid=0
[root@user guest]# id
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

{#fig:005 width=70%} 13. Создайте программу readfile.c (рис. [-@fig:006]).

```
readfile.c [----] 29 L: [ 2+18 20/ 23] *(363 / 395b) 0098 0x062 [*] [X]
#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;
}
```

{#fig:006 width=70%} 15. Смените владельца у файла readfile.c (или любого другого текстового файла в системе) и измените права так, чтобы только суперпользователь (root) мог прочитать его, а guest не мог. 16. Проверьте, что пользователь guest не может прочитать файл readfile.c. 17. Смените у программы readfile владельца и установите SetU'D-бит. 18. Проверьте, может ли программа readfile прочитать файл readfile.c? (рис. [-@fig:007]).

```
[root@user guest]# chown root:guest /home/guest/readfile.c
[root@user guest]# chmod 700 /home/guest/readfile.c
[root@user guest]# exit
exit
[guest@user ~]$ cat readfile.c
cat: readfile.c: Отказано в доступе
[guest@user ~]$ su
Пароль:
su: Сбой при проверке подлинности
[guest@user ~]$ su
Пароль:
[root@user guest]# chown root:guest /home/guest/readfile.c
[root@user guest]# chmod u+s /home/guest/readfile.c
[root@user guest]# exit
exit
```

{#fig:007 width=70%)}

1. Выясните, установлен ли атрибут Sticky на директории /tmp, для чего выполните команду `ls -l / | grep tmp`
2. От имени пользователя guest создайте файл file01.txt в директории /tmp со словом test: `echo "test" > /tmp/file01.txt`
3. Просмотрите атрибуты у только что созданного файла и разрешите чтение и запись для категории пользователей «все остальные»: `ls -l /tmp/file01.txt chmod o+rw /tmp/file01.txt ls -l /tmp/file01.txt` (рис. [-@fig:008]).

```
[guest@user ~]$ ls -l / | grep tmp
drwxrwxrwt. 25 root root 4096 anp 13 22:34 tmp
[guest@user ~]$ echo "test" > /tmp/file01.txt
[guest@user ~]$ ls -l /tmp/file01.txt
-rw-r--r--. 1 guest guest 5 anp 13 22:36 /tmp/file01.txt
[guest@user ~]$ chmod o+rw /tmp/file-1.txt
chmod: невозможно получить доступ к '/tmp/file-1.txt': Нет такого файла или каталога
[guest@user ~]$ chmod o+rw /tmp/file01.txt
```

{#fig:008 width=70%)}

4. От пользователя guest2 (не являющегося владельцем) попробуйте прочитать файл /tmp/file01.txt: `cat /tmp/file01.txt`
5. От пользователя guest2 попробуйте дозаписать в файл /tmp/file01.txt слово test2 командой `echo "test2" > /tmp/file01.txt` Удалось ли вам выполнить операцию?
6. Проверьте содержимое файла командой `cat /tmp/file01.txt`
7. От пользователя guest2 попробуйте записать в файл /tmp/file01.txt слово test3, стерев при этом всю имеющуюся в файле информацию командой `echo "test3" > /tmp/file01.txt` Удалось ли вам выполнить операцию?
8. Проверьте содержимое файла командой `cat /tmp/file01.txt` (рис. [-@fig:009]).

```
[guest2@user guest]$ echo "test2" >> /tmp/file01.txt
[guest2@user guest]$ cat /tmp/file01.txt
test
test2
[guest2@user guest]$ echo "test3" > /tmp/file01.txt
[guest2@user guest]$ cat /tmp/file01.txt
test3
```

{#fig:009 width=70%})

9. От пользователя guest2 попробуйте удалить файл /tmp/file01.txt командой rm /tmp/file01.txt. Удалось ли вам удалить файл?
10. Повысьте свои права до суперпользователя следующей командой su - и выполните после этого команду, снимающую атрибут t (Sticky-бит) с директории /tmp: chmod -t /tmp
11. Покиньте режим суперпользователя командой exit
12. От пользователя guest2 проверьте, что атрибута t у директории /tmp нет: ls -l / | grep tmp
13. Повторите предыдущие шаги. Какие наблюдаются изменения? (рис. [-@fig:010]).

```
[guest2@user guest]$ rm /tmp/file01.txt
rm: невозможно удалить '/tmp/file01.txt': Операция не позволена
[guest2@user guest]$ su
Пароль:
su: Сбой при проверке подлинности
[guest2@user guest]$ su
Пароль:
[root@user guest]# chmod -t /tmp
[root@user guest]# exit
exit
[guest2@user guest]$ ls -l / | grep tmp
drwxrwxrwx. 25 root root 4096 апр 13 22:41 tmp
[guest2@user guest]$ rm /tmp/file01.txt
```

{#fig:010 width=70%})

15. Повысьте свои права до суперпользователя и верните атрибут t на директорию /tmp: su - chmod +t /tmp exit (рис. [-@fig:011]).

```
[guest2@user guest]$ su
Пароль:
[root@user guest]# chmod +t /tmp
[root@user guest]# exit
exit
[guest2@user guest]$
```

{#fig:011 width=70%})

3 Выводы

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

Список литературы