

Лабораторная работа №5

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

Джангиров Илгар

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Исследование Sticky-бита	11
4	Выводы	14

Список иллюстраций

2.1	1	6
2.2	Исходный файл	6
2.3	Результат	7
2.4	Результат	7
2.5	Изменение	7
2.6	Изменение	8
2.7	7	8
2.8	7	9
2.9	7	9
2.10	7	10
3.1	7	11
3.2	7	11
3.3	7	12
3.4	7	12
3.5	7	12
3.6	7	12
3.7	7	13
3.8	7	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 лабораторной работы

1. Войдите в систему от имени пользователя guest.
2. Создайте программу simpleid.c:

```
[guest@localhost ~]$ vim readfile.c
[guest@localhost ~]$ gcc readfile.c -o readfile
[guest@localhost ~]$ sudo chown root:root readfile.c

Мы полагаем, что ваш системный администратор изложил вам основы
безопасности. Как правило, всё сводится к трём следующим правилам:

№1) Уважайте частную жизнь других.
№2) Думайте, прежде что-то вводить.
№3) С большой властью приходит большая ответственность.

[sudo] пароль для guest:
guest is not in the sudoers file. This incident will be reported.
[guest@localhost ~]$ sudo chmod 700 readfile.c
[sudo] пароль для guest:
guest is not in the sudoers file. This incident will be reported.
[guest@localhost ~]$ cat readfile.c
```

Рис. 2.1: 1

```
guest@localhost:~
#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);
    retur 0;
}
~
```

Рис. 2.2: Исходный файл

3. Скомпилируйте программу и убедитесь, что файл программы создан: `gcc simpleid.c -o simpleid`
4. Выполните программу `simpleid`: `./simpleid`

```
[guest@localhost ~]$ gcc simpleid.c -o simpleid
cc1: фатальная ошибка: simpleid.c: Нет такого файла или каталога
компиляция прервана.
[guest@localhost ~]$ ./simpleid

-bash: ./simpleid: Нет такого файла или каталога
[guest@localhost ~]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@localhost ~]$
```

Рис. 2.3: Результат

5. Выполните системную программу `id`: `id` и сравните полученный вами результат с данными предыдущего пункта задания.

```
cc1: фатальная ошибка: simpleid2.c: Нет такого файла или каталога
компиляция прервана.
[guest@localhost ~]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@localhost ~]$
```

Рис. 2.4: Результат

6. Усложните программу, добавив вывод действительных идентификаторов:

```
finid_t:s0-s0:c0.c1023
[guest@localhost ~]$ vim simplified2.
[guest@localhost ~]$ gcc simpleid2.c -o simpleid2
cc1: фатальная ошибка: simpleid2.c: Нет такого файла или каталога
компиляция прервана.
[guest@localhost ~]$ ./simplified2
-bash: ./simplified2: Нет такого файла или каталога
[guest@localhost ~]$ chown root:guest /home/guest/simpleid2
chown: невозможно получить доступ к '/home/guest/simpleid2': Нет такого файла или каталога
[guest@localhost ~]$
```

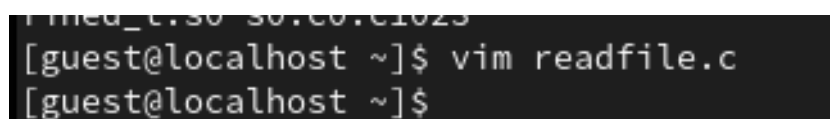
Рис. 2.5: Изменение



```
guest@localhost:~  
#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. 2.6: Изменение

7. Скомпилируйте и запустите simpleid2.c: `gcc simpleid2.c -o simpleid2`
`./simpleid2`



```
guest@localhost:~$ vim readfile.c  
guest@localhost:~$
```

Рис. 2.7: 7

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`:


```
[guest@localhost ~]$ sudo chown root:guest ./readfile
[sudo] пароль для guest:
guest is not in the sudoers file. This incident will be reported.
[guest@localhost ~]$ sudo chmod u+s ./readfile
[sudo] пароль для guest: 
```

Рис. 2.8: 7

11. Запустите simpleid2 и id: ./simpleid2 id Сравните результаты.
12. Прodelайте тоже самое относительно SetGID-бита.
13. Создайте программу readfile.c:

```
guest@localhost:~
#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. 2.9: 7

14. Откомпилируйте её. gcc readfile.c -o readfile

```
[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;
}
[guest@localhost ~]$
```

Рис. 2.10: 7

3 Исследование Sticky-бита

1. Выясните, установлен ли атрибут Sticky на директории /tmp, для чего выполните команду `ls -l / | grep tmp`

```
[guest@localhost ~]$ ls -l / | grep tmp
drwxrwxrwt. 17 root root 4096 апр 12 21:07 tmp
[guest@localhost ~]$
```

Рис. 3.1: 7

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`

```
[guest@localhost ~]$ echo "test" > /tmp/file01.txt
[guest@localhost ~]$ ls -l /tmp/file01.txt
-rw-r--r--. 1 guest guest 5 апр 12 21:08 /tmp/file01.txt
[guest@localhost ~]$ chmod o+rw /tmp/file01.txt
[guest@localhost ~]$ ls -l /tmp/file01.txt
-rw-r--rw-. 1 guest guest 5 апр 12 21:08 /tmp/file01.txt
[guest@localhost ~]$
```

Рис. 3.2: 7

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

```
[guest@localhost ~]$ echo "test2" > /tmp/file01.txt
[guest@localhost ~]$ echo /tmp/file01.txt
/tmp/file01.txt
[guest@localhost ~]$ cat /tmp/file01.txt
test2
[guest@localhost ~]$
```

Рис. 3.3: 7

```
[guest2@localhost ~]$ rm /tmp/file01.txt
rm: невозможно удалить '/tmp/file01.txt': Нет такого файла или каталога
[guest2@localhost ~]$
```

Рис. 3.4: 7

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

```
пароль.
[guest2@localhost ~]$ echo "test3" > /tmp/file01.txt
-bash: /tmp/file01.txt: Отказано в доступе
[guest2@localhost ~]$
```

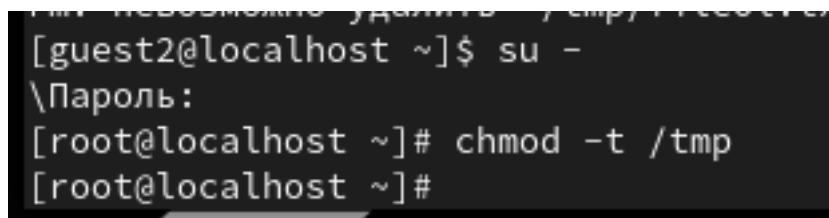
Рис. 3.5: 7

8. Проверьте содержимое файла командой cat /tmp/file01.txt

```
[guest2@localhost ~]$ cat /tmp/file01.txt
test2
[guest2@localhost ~]$
```

Рис. 3.6: 7

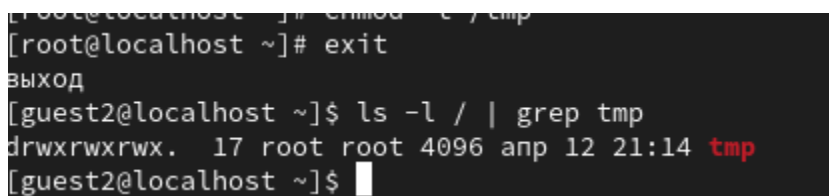
9. От пользователя guest2 попробуйте удалить файл /tmp/file01.txt командой `rm /tmp/file01.txt`. Удалось ли вам удалить файл?
10. Повысьте свои права до суперпользователя следующей командой `su -` и выполните после этого команду, снимающую атрибут `t` (Sticky-бит) с директории /tmp: `chmod -t /tmp`



```
[guest2@localhost ~]$ su -
\Пароль:
[root@localhost ~]# chmod -t /tmp
[root@localhost ~]#
```

Рис. 3.7: 7

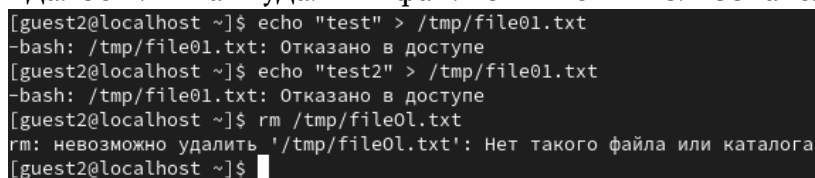
11. Покиньте режим суперпользователя командой `exit`



```
[root@localhost ~]# exit
выход
[guest2@localhost ~]$ ls -l / | grep tmp
drwxrwxrwx. 17 root root 4096 апр 12 21:14 tmp
[guest2@localhost ~]$
```

Рис. 3.8: 7

12. От пользователя guest2 проверьте, что атрибута `t` у директории /tmp нет: `ls -l / | grep tmp`
13. Повторите предыдущие шаги. Какие наблюдаются изменения?
14. Удалось ли вам удалить файл от имени пользователя, не являющегося



```
[guest2@localhost ~]$ echo "test" > /tmp/file01.txt
-bash: /tmp/file01.txt: Отказано в доступе
[guest2@localhost ~]$ echo "test2" > /tmp/file01.txt
-bash: /tmp/file01.txt: Отказано в доступе
[guest2@localhost ~]$ rm /tmp/file01.txt
rm: невозможно удалить '/tmp/file01.txt': Нет такого файла или каталога
[guest2@localhost ~]$
```

4 Выводы

В результате выполнения работы я выполнил цели работы ::: {#refs} :::