

Отчёт по лабораторной работе №5

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

Кондрашина Мария Сергеевна

Содержание

1	Цель работы	5
2	Теоретические сведения	6
3	Выполнение лабораторной работы. Создание программы	7
4	Исследование Sticky-бита	14
5	Выводы	17
6	Список литературы	18

List of Figures

3.1	Подготовка лабораторного стенда	7
3.2	Программа simpleid.c	8
3.3	Пункт 1-5	8
3.4	Программа simpleid2.c	9
3.5	Компилирование и запуск simpleid2.c	9
3.6	SetUID-бит	10
3.7	SetGID-бит	10
3.8	Создание файла readfile.c	10
3.9	Программа readfile.c	11
3.10	Смена прав так, чтобы только суперпользователь мог прочесть файл	11
3.11	Проверка	11
3.12	readfile SetUID-бит	12
3.13	Проверка чтения файла readfile.c	12
3.14	Проверка чтения файла /etc/shadow	13
4.1	Атрибут Sticky на директории /tmp и файл file01.txt	14
4.2	guest2 и file01	15
4.3	Снятие атрибута t (Sticky-бит) с директории /tmp	15
4.4	Взаимодействие guest2 с файлом, когдан атрибута t (Sticky-бит) снят с директории /tmp	16
4.5	Возвращение атрибута t (Sticky-бит) на директорию /tmp	16

List of Tables

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 Теоретические сведения

Setuid, Setgid и Sticky Bit - это специальные типы разрешений позволяют задавать расширенные права доступа на файлы или каталоги.

Setuid – это бит разрешения, который позволяет пользователю запускать исполняемый файл с правами владельца этого файла. Другими словами, использование этого бита позволяет нам поднять привилегии пользователя в случае, если это необходимо.

Принцип работы Setgid похож на setuid с отличием, что файл будет запускаться пользователем от имени группы, которая владеет файлом.

В случае, если Sticky Bit установлен для папки, то файлы в этой папке могут быть удалены только их владельцем, тоже самое верно и для файлов.[2]

3 Выполнение лабораторной работы.

Создание программы

1. Подготовка лабораторного стенда (fig. 3.1)

```
[guest@mskondrashina ~]$ su root
Password:
[root@mskondrashina guest]# yum install gcc
Rocky Linux 9 - BaseOS                6.1 kB/s | 3.6 kB      00:00
Rocky Linux 9 - BaseOS                1.4 MB/s | 1.7 MB      00:01
Rocky Linux 9 - AppStream              5.8 kB/s | 3.6 kB      00:00
Rocky Linux 9 - AppStream              2.9 MB/s | 6.0 MB      00:02
Rocky Linux 9 - Extras                 2.9 kB/s | 2.9 kB      00:01
Package gcc-11.2.1-9.4.el9.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@mskondrashina guest]# setenforce 0
[root@mskondrashina guest]# getenforce
Permissive
```

Figure 3.1: Подготовка лабораторного стенда

2. Вошла в систему от имени пользователя guest, создала программу simpleid.c(fig. 3.2). Скомпилировала программу и выполнила ее, также выполнила системную программу id т сравнила вывод двух программ - вывелись одинаковые значения.(fig. 3.3)

Figure 3.2: Программа simpleid.c

```
[root@mskondrashina guest]# su guest
[guest@mskondrashina ~]$ touch simpleid.c
[guest@mskondrashina ~]$ gcc simpleid.c -o simpleid
[guest@mskondrashina ~]$ ./simpleid
uid=1001, gid=1001
[guest@mskondrashina ~]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfi
ned_t:s0-s0:c0.c1023
```

Figure 3.3: Пункт 1-5

3. Усложнила программу, добавив вывод действительных идентификаторов тила (fig. 3.4), скомпилировала и запустила (fig. 3.5)

Figure 3.4: Программа simpleid2.c

Figure 3.5: Компилирование и запуск simpleid2.c

4. От имени суперпользователя выполнила команды: `chown root:guest /home/guest/simpleid2`, `chmod u+s /home/guest/simpleid2`. Также выполнила проверку правильности установки новых атрибутов и смены владельца файла `simpleid2`: `ls -l simpleid2`, запустила `simpleid2` и `id`, результаты которых оказались одинаковыми.(fig. 3.6)

```
[guest@mskondrashina ~]$ su root
Password:
[root@mskondrashina guest]# chown root:guest /home/guest/simpleid2
[root@mskondrashina guest]# chmod u+s /home/guest/simpleid2
[root@mskondrashina guest]# ls -l simpleid2
-rwsrwxr-x. 1 root guest 26008 Oct  4 14:40 simpleid2
[root@mskondrashina guest]# ./simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@mskondrashina guest]# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Figure 3.6: SetUID-бит

5. Проделала тоже самое относительно SetGID-бита.(fig. 3.7)

```
[root@mskondrashina guest]# chmod g+s /home/guest/simpleid2
[root@mskondrashina guest]# ls -l simpleid2
-rwsrwsr-x. 1 root guest 26008 Oct  4 14:40 simpleid2
[root@mskondrashina guest]# ./simpleid2
e_uid=0, e_gid=1001
real_uid=0, real_gid=0
[root@mskondrashina guest]# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Figure 3.7: SetGID-бит

6. Создала программу readfile.c (fig. 3.8)(fig. 3.9)

```
[root@mskondrashina guest]# su guest
[guest@mskondrashina ~]$ touch readfile.c
[guest@mskondrashina ~]$ su root
```

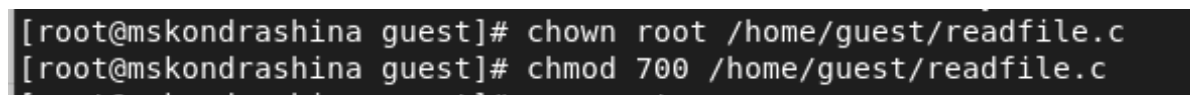
Figure 3.8: Создание файла readfile.c



```
1 #include <fcntl.h>
2 #include <stdio.h>
3 #include <sys/stat.h>
4 #include <sys/types.h>
5 #include <unistd.h>
6
7 int
8 main (int argc, char* argv[])
9 {
10     unsigned char buffer[16];
11     size_t bytes_read;
12     int i;
13
14     int fd = open (argv[1], O_RDONLY);
15     do
16     {
17         bytes_read = read (fd, buffer, sizeof(buffer));
18         for(i = 0; i < bytes_read; ++i) printf("%c", buffer[i]);
19     }
20
21     while (bytes_read == sizeof (buffer));
22     close(fd);
23     return 0;
24 }
```

Figure 3.9: Программа readfile.c

7. Сменила владельца у файла readfile.c и изменила права так, чтобы только суперпользователь (root) мог прочитать его, а guest не мог.(fig. 3.10)



```
[root@mskondrashina guest]# chown root /home/guest/readfile.c
[root@mskondrashina guest]# chmod 700 /home/guest/readfile.c
```

Figure 3.10: Смена прав так, чтобы только суперпользователь мог прочесть файл

8. Проверила, что пользователь guest не может прочитать файл readfile.c (fig. 3.11)



```
[root@mskondrashina guest]# su guest
[guest@mskondrashina ~]$ cat readfile.c
cat: readfile.c: Permission denied
```

Figure 3.11: Проверка

9. Сменила у программы readfile владельца и установила SetUID-бит.
(fig. 3.12)

```
[guest@mskondrashina ~]$ su root
Password:
[root@mskondrashina guest]# chown root:guest /home/guest/readfile.c
[root@mskondrashina guest]# chmod u+s /home/guest/readfile.c
```

Figure 3.12: readfile SetUID-бит

10. Проверила, может ли программа readfile прочитать файл readfile.c - Да, может. (fig. 3.13)

```
[root@mskondrashina guest]# gcc readfile.c -o readfile
[root@mskondrashina 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;
}
```

Figure 3.13: Проверка чтения файла readfile.c

11. Проверила, может ли программа readfile прочитать файл /etc/shadow. - Да, может. (fig. 3.14)

```
[root@mskondrashina guest]# ./readfile /etc/shadow
root:$6$BjQnQGS/.VKfTpWT$Yg5ZUZMLxL8k.pcct9CngiuWlaZEsrXWysFjZNTa8v0TNg0dsbwno3laZXTXey9XtcXo6b.1WAn/wZob
uYjjP.::0:99999:7:::
bin:!:19123:0:99999:7:::
daemon:!:19123:0:99999:7:::
adm:!:19123:0:99999:7:::
lp:!:19123:0:99999:7:::
sync:!:19123:0:99999:7:::
shutdown:!:19123:0:99999:7:::
halt:!:19123:0:99999:7:::
mail:!:19123:0:99999:7:::
operator:!:19123:0:99999:7:::
games:!:19123:0:99999:7:::
ftp:!:19123:0:99999:7:::
nobody:!:19123:0:99999:7:::
systemd-coredump:!!:19241:::::
dbus:!!:19241:::::
polkitd:!!:19241:::::
rtkit:!!:19241:::::
sssd:!!:19241:::::
avahi:!!:19241:::::
pipewire:!!:19241:::::
libstoragemgmt:!!:19241:::::
tss:!!:19241:::::
geoclue:!!:19241:::::
cockpit-ws:!!:19241:::::
cockpit-wsinstance:!!:19241:::::
setroubleshoot:!!:19241:::::
flatpak:!!:19241:::::
colord:!!:19241:::::
```

Figure 3.14: Проверка чтения файла /etc/shadow

4 Исследование Sticky-бита

1. Выяснила, установлен ли атрибут Sticky на директории /tmp. От имени пользователя guest создала файл file01.txt в директории /tmp со словом test. Просмотрела атрибуты у только что созданного файла и разрешила чтение и запись для категории пользователей «все остальные» (fig. 4.1)

```
[root@mskondrashina guest]# ls -l / | grep tmp
drwxrwxrwt. 15 root root 4096 Oct  4 15:07 tmp
[root@mskondrashina guest]# su guest
[guest@mskondrashina ~]$ echo "test" > /tmp/file01.txt
[guest@mskondrashina ~]$ ls -l /tmp/file01.txt
-rw-rw-r--. 1 guest guest 5 Oct  4 15:12 /tmp/file01.txt
[guest@mskondrashina ~]$ chmod o+rw /tmp/file01.txt
[guest@mskondrashina ~]$ ls -l /tmp/file01.txt
-rw-rw-rw-. 1 guest guest 5 Oct  4 15:12 /tmp/file01.txt
```

Figure 4.1: Атрибут Sticky на директории /tmp и файл file01.txt

2. От пользователя guest2 попробовала прочитать файл /tmp/file01.txt, дозаписать в файл, записать в файл слово test3, стерев при этом всю имеющуюся в файле информацию, удалить файл.(fig. 4.2)

Все удалось кроме удаления файла.

```
[guest@mskondrashina ~]$ su guest2
Password:
[guest2@mskondrashina guest]$ cat /tmp/file01.txt
test
[guest2@mskondrashina guest]$ echo "test2" >> /tmp/file01.txt
[guest2@mskondrashina guest]$ cat /tmp/file01.txt
test
test2
[guest2@mskondrashina guest]$ echo "test3" > /tmp/file01.txt
[guest2@mskondrashina guest]$ cat /tmp/file01.txt
test3
[guest2@mskondrashina guest]$ rm /tmp/file01.txt
rm: cannot remove '/tmp/file01.txt': Operation not permitted
```

Figure 4.2: guest2 и file01

3. Повысила свои права до суперпользователя и выполнила после этого команду, снимающую атрибут t (Sticky-бит) с директории /tmp, покинула режим суперпользователя.(fig. 4.3)

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

Figure 4.3: Снятие атрибута t (Sticky-бит) с директории /tmp

4. От пользователя guest2 проверила, что атрибута t у директории /tmp нет, повторила предыдущие шаги.(fig. 4.4)

Удалось удалить файл от имени пользователя, не являющегося его владельцем

```
[guest2@mskondrashina guest]$ ls -l / | grep tmp
drwxrwxrwx. 15 root root 4096 Oct  4 15:16 tmp
[guest2@mskondrashina guest]$ cat /tmp/file01.txt
test3
[guest2@mskondrashina guest]$ echo "test2" >> /tmp/file01.txt
[guest2@mskondrashina guest]$ cat /tmp/file01.txt
test3
test2
[guest2@mskondrashina guest]$ echo "test3" > /tmp/file01.txt
[guest2@mskondrashina guest]$ cat /tmp/file01.txt
test3
[guest2@mskondrashina guest]$ rm /tmp/file01.txt
[guest2@mskondrashina guest]$
```

Figure 4.4: Взаимодействие guest2 с файлом, когдан атрибута t (Sticky-бит) снят с директории /tmp

5. Повысила свои права до суперпользователя и вернула атрибут t на директорию /tmp.(fig. 4.5)

```
[guest2@mskondrashina guest]$ su -
Password:
[root@mskondrashina ~]# chmod +t /tmp
[root@mskondrashina ~]# exit
logout
[guest2@mskondrashina guest]$ ls -l / | grep tmp
drwxrwxrwt. 17 root root 4096 Oct  4 15:19 tmp
[guest2@mskondrashina guest]$
```

Figure 4.5: Возвращение атрибута t (Sticky-бит) на директорию /tmp

5 Выводы

Выполнила лабораторную работу №5.

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

6 Список литературы

1. Методические материалы курса. “Информационная безопасность компьютерных сетей” Кулябов Д. С., Королькова А. В., Геворкян М. Н.
2. <https://ruvds.com/ru/helpcenter/suid-sgid-sticky-bit-linux/>