Отчёт по лабораторной работе №5. Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

Предмет: информационная безопасность

Александр Сергеевич Баклашов

Содержание

List of Figures

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. [1]

2 Теоретическое введение

Информационная безопасность – это защищенность информации и поддерживающей инфраструктуры от случайных или преднамеренных воздействий естественного или искусственного характера, чреватых нанесением ущерба владельцам или пользователям информации и поддерживающей инфраструктуры.

chmod (от англ. change mode) — команда для изменения прав доступа к файлам и каталогам, используемая в Unix-подобных операционных системах. Входит в стандарт POSIX, в Coreutils. [3]

3 Выполнение лабораторной работы

3.1 Создание программы

1. Войдите в систему от имени пользователя guest. (рис. 3.1)

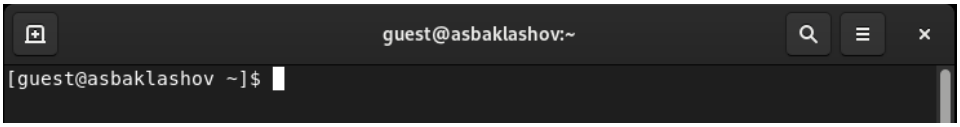

Figure 3.1: guest

2. Создайте программу simpleid.c: (рис. 3.2)

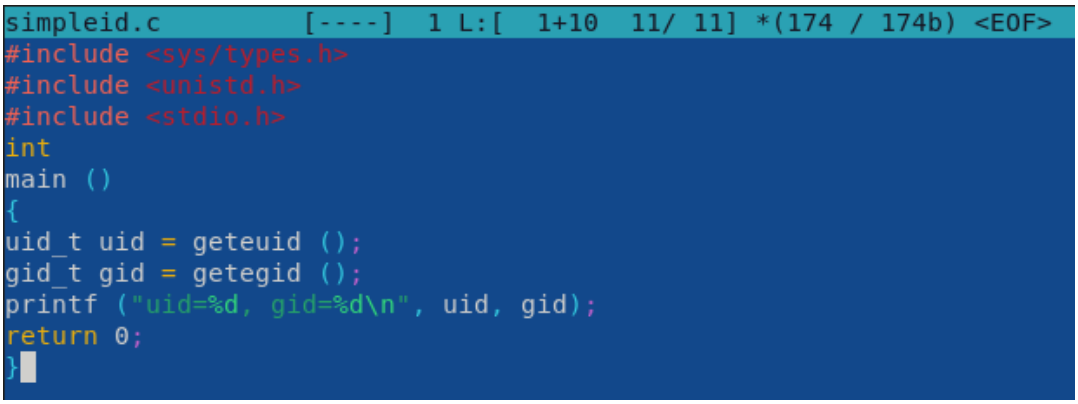

Figure 3.2: simpleid.c

3. Скомплилируйте программу и убедитесь, что файл программы создан (рис. 3.3)

```
[guest@asbaklashov Programs]$ gcc simpleid.c -o simpleid
[guest@asbaklashov Programs]$
```

Figure 3.3: Скомплилируем

4. Выполните программу simpleid

Выполните системную программу id и сравните полученный вами результат с данными предыдущего пункта задания. (рис. 3.4)

```
[guest@asbaklashov Programs]$ ./simpleid
uid=1001, gid=1001
[guest@asbaklashov Programs]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfin
ed_r:unconfined_t:s0-s0:c0.c1023
[guest@asbaklashov Programs]$
```

Figure 3.4: id

Результаты совпадают.

5. Усложните программу, добавив вывод действительных идентификаторов. (рис. 3.5)

```
simpleid2.c        [----]  1 L:[  1+13  14/ 14] *(302 / 302b) <EOF>
#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;
}
```

Figure 3.5: Усложним программу

Получившуюся программу назовите simpleid2.c

6. Скомпилируйте и запустите simpleid2.c (рис. 3.6)

```
[guest@asbaklashov Programs]$ gcc simpleid2.c -o simpleid2
[guest@asbaklashov Programs]$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
```

Figure 3.6: simpleid2.c

7. От имени суперпользователя выполните команды.

chown root:guest /home/guest/simpleid2

chmod u+s /home/guest/simpleid2 (рис. 3.7)

```
[guest@asbaklashov Programs]$ su
Password:
[root@asbaklashov Programs]# chown root:guest /home/guest/simpleid2
chown: cannot access '/home/guest/simpleid2': No such file or directory
[root@asbaklashov Programs]# chown root:guest /home/guest/Programs/simpleid2
[root@asbaklashov Programs]# chmod u+s /home/guest/Programs/simpleid2
```

Figure 3.7: chmod

Используйте sudo или повысьте временно свои права с помощью su. Поясните, что делают эти команды.

1ая команда меняет владельца, 2ая - атрибуты

8. Выполните проверку правильности установки новых атрибутов и смены владельца файла simpleid2 (рис. 3.8)

```
[root@asbaklashov Programs]# ls -l simpleid2
-rwsrwxr-x. 1 root guest 26008 Oct  4 18:02 simpleid2
[root@asbaklashov Programs]#
```

Figure 3.8: ls

9. Запустите simpleid2 и id (рис. 3.9)

```
[root@asbaklashov Programs]# ./simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@asbaklashov Programs]# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfi
ned_t:s0-s0:c0.c1023
[root@asbaklashov Programs]#
```

Figure 3.9: id

Результаты совпадают.

10. Проделайте тоже самое относительно SetGID-бита (рис. 3.10)

```
[root@asbaklashov Programs]# chmod g+s /home/guest/Programs/simpleid2
[root@asbaklashov Programs]# ls -l simpleid2
-rwsrwsr-x. 1 root guest 26008 Oct  4 18:02 simpleid2
[root@asbaklashov Programs]# ./simpleid2
e_uid=0, e_gid=1001
real_uid=0, real_gid=0
[root@asbaklashov Programs]# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfi
ned_t:s0-s0:c0.c1023
[root@asbaklashov Programs]# exit
exit
[guest@asbaklashov Programs]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfin
ed_r:unconfined_t:s0-s0:c0.c1023
[guest@asbaklashov Programs]$
```

Figure 3.10: SetGID

11. Создайте программу readfile.c (рис. 3.11)

```
readfile.c          [----]  1 L:[  1+21  22/ 22] *(402 / 402b) <EOF>
#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 3.11: readfile.c

12. Откомпилируйте её (рис. 3.12)

```
[guest@asbaklashov Programs]$ gcc readfile.c -o readfile
[guest@asbaklashov Programs]$
```

Figure 3.12: Compile

13. 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 (рис. 3.13)

```
[guest@asbaklashov Programs]$ ls -l
total 96
-rwxrwxr-x. 1 guest guest 25952 Oct  4 18:15 readfile
-rw-rw-r--. 1 guest guest   402 Oct  4 18:15 readfile.c
-rwxrwxr-x. 1 guest guest 25904 Oct  4 17:58 simpleid
-rwsrwsr-x. 1 root  guest 26008 Oct  4 18:02 simpleid2
-rw-rw-r--. 1 guest guest   302 Oct  4 18:01 simpleid2.c
-rw-rw-r--. 1 guest guest   174 Oct  4 17:57 simpleid.c
[guest@asbaklashov Programs]$ su
Password:
[root@asbaklashov Programs]# chown root:guest /home/guest/Programs/readfile.c
[root@asbaklashov Programs]# chmod 700 readfile.c
[root@asbaklashov Programs]# ls -l simpleid2
-rwsrwsr-x. 1 root guest 26008 Oct  4 18:02 simpleid2
[root@asbaklashov Programs]# ls -l readfile.c
-rwx------. 1 root guest 402 Oct  4 18:15 readfile.c
[root@asbaklashov Programs]#
```

Figure 3.13: Смена владельца

14. Проверьте, что пользователь guest не может прочитать файл readfile.c. (рис. 3.14)

```
[guest@asbaklashov Programs]$ cat readfile.c
cat: readfile.c: Permission denied
[guest@asbaklashov Programs]$
```

Figure 3.14: Проверка

15. Смените у программы readfile владельца и установите SetU'D-бит (рис. 3.15)

```
[root@asbaklashov Programs]# chown root:guest /home/guest/Programs/readfile
[root@asbaklashov Programs]# chmod u+s /home/guest/Programs/readfile
[root@asbaklashov Programs]# ls -l
total 96
-rwsrwxr-x. 1 root  guest 25952 Oct  4 18:15 readfile
-rwx------. 1 root  guest   402 Oct  4 18:15 readfile.c
-rwxrwxr-x. 1 guest guest 25904 Oct  4 17:58 simpleid
-rwsrwsr-x. 1 root  guest 26008 Oct  4 18:02 simpleid2
-rw-rw-r--. 1 guest guest   302 Oct  4 18:01 simpleid2.c
-rw-rw-r--. 1 guest guest   174 Oct  4 17:57 simpleid.c
[root@asbaklashov Programs]#
```

Figure 3.15: Смена владельца

16. Проверьте, может ли программа readfile прочитать файл readfile.c? (рис. 3.16)

```
[guest@asbaklashov Programs]$ ./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 3.16: Проверка

17. Проверьте, может ли программа readfile прочитать файл /etc/shadow? (рис. 3.17)

Figure 3.17: Проверка

Программа может прочитать оба файла.

3.2 Исследование Sticky-бита

1. Выясните, установлен ли атрибут Sticky на директории /tmp, для чего выполните команду ls -l / | grep tmp (рис. 3.18)

Figure 3.18: Sticky

Атрибут “t” установлен.

2. От имени пользователя guest создайте файл file01.txt в директории /tmp со словом test

Просмотрите атрибуты у только что созданного файла и разрешите чтение и запись для категории пользователей «все остальные». (рис. 3.19)

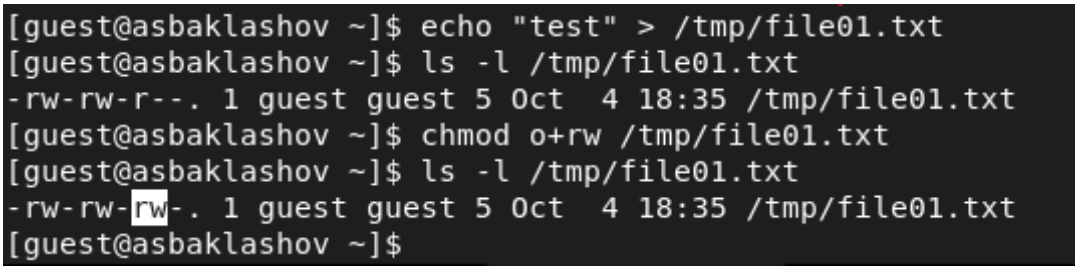

Figure 3.19: file01.txt

3. От пользователя guest2 (не являющегося владельцем) попробуйте прочитать файл /tmp/file01.txt

От пользователя guest2 попробуйте дозаписать в файл /tmp/file01.txt слово test2 командой

Проверьте содержимое файла командой cat /tmp/file01.txt (рис. 3.20)

```
[guest@asbaklashov ~]$ su guest2
Password:
[guest2@asbaklashov guest]$ cat /tmp/file01.txt
test
[guest2@asbaklashov guest]$ echo "test2" >> /tmp/file01.txt
[guest2@asbaklashov guest]$ cat /home/guest/dir1/file1
cat: /home/guest/dir1/file1: Permission denied
[guest2@asbaklashov guest]$ cat /tmp/file01.txt
test
test2
[guest2@asbaklashov guest]$
```

Figure 3.20: атрибуты

4. От пользователя guest2 попробуйте записать в файл /tmp/file01.txt слово test3, стерев при этом всю имеющуюся в файле информацию командой

Проверьте содержимое файла командой cat /tmp/file01.txt (рис. 3.21)

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

Figure 3.21: запись в файл

5. От пользователя guest2 попробуйте удалить файл /tmp/file01.txt командой rm /tmp/fileOl.txt (рис. 3.22)

```
[guest2@asbaklashov guest]$ rm /tmp/file01.txt
rm: cannot remove '/tmp/file01.txt': Operation not permitted
[guest2@asbaklashov guest]$
```

Figure 3.22: удаление файла

Удалить файл не удалось

6. Повысьте свои права до суперпользователя следующей командой su - и выполните после этого команду, снимающую атрибут t (Sticky-бит) с директории /tmp: chmod -t /tmp

Покиньте режим суперпользователя командой exit

От пользователя guest2 проверьте, что атрибута t у директории /tmp нет: ls -l / | grep tmp (рис. 3.23)

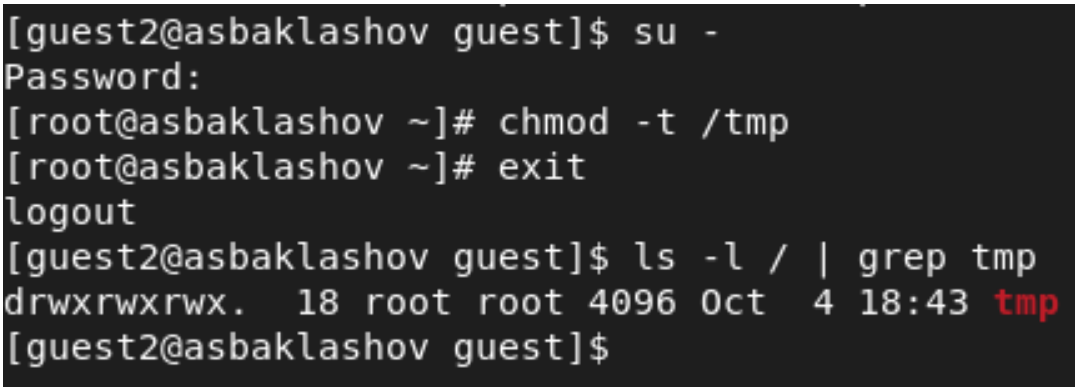

Figure 3.23: Снятие Sticky-бита

7. Повторим предыдущие шаги (рис. 3.24, рис. 3.25)

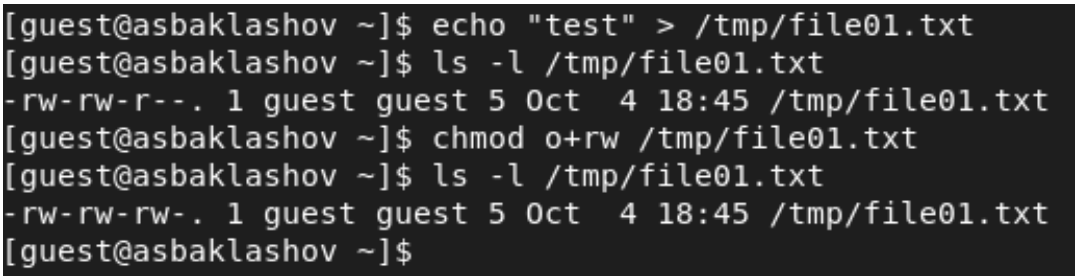

Figure 3.24: шаги

Figure 3.25: шаги

8. Повысьте свои права до суперпользователя и верните атрибут t на директорию /tmp (рис. 3.26)

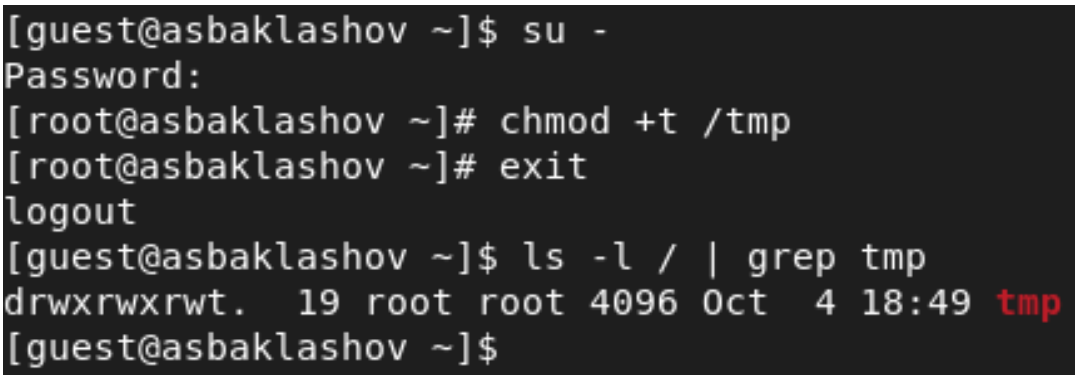

Figure 3.26: Возвращение Sticky-бита

4 Вывод

В результате выполнения работы я изучил механизмы изменения идентификаторов, применения SetUID- и Sticky-битов. Получил практические навыки работы в консоли с дополнительными атрибутами. Рассмотрел работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

5 Библиография

1. Лабораторная работа №5. Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов. - 7 с. [Электронный ресурс]. М. URL: Лабораторная работа №5 (Дата обращения: 08.10.2022).

2. Rocky Linux Documentation. [Электронный ресурс]. М. URL: Rocky Linux Documentation (Дата обращения: 08.10.2022).

3. Chmod. [Электронный ресурс]. М. URL: Chmod (Дата обращения: 08.10.2022).