Лабораторная работа №5

Краснова Диана Владимировна

Содержание

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

2.1 Изучение механики SetUID

1. Вошли в систему от имени пользователя guest.
2. Написали программу simpleid.c.
3. Скомпилировали программу и убедились, что файл программы создан: gcc simpleid.c -o simpleid

программа simpleid

4. Выполнили программу simpleid командой ./simpleid
5. Выполнили системную программу id с помощью команды id. uid и gid совпадает в обеих программах

```
[guest@dvkrasnova ~]$ id
uid=1002(guest) gid=1002(guest) groups=1002(guest) context=unconfined_u:unconfin
ed_r:unconfined_t:s0-s0:c0.c1023
[guest@dvkrasnova ~]$ ./simpleid
uid=1002, gid=1002
[guest@dvkrasnova ~]$
```

результат программы simpleid

6. Усложнили программу, добавив вывод действительных идентификаторов.

```
#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;
}
```

программа simpleid2

7. Скомпилировали и запустили simpleid2.c:

```
gcc simpleid2.c -o simpleid2
./simpleid2
```

8. От имени суперпользователя выполнили команды:

```
chown root:guest /home/guest/simpleid2
chmod u+s /home/guest/simpleid2
```

```
[guest@dvkrasnova ~]$ su
Password:
[root@dvkrasnova guest]# chown root:guest /home/guest/simpleid2
[root@dvkrasnova guest]# chmod u+s /home/guest/simpleid2
[root@dvkrasnova guest]# ls -l simpleid2
-rwsrwxr-x. 1 root guest 18312 Oct  6 14:30 simpleid2
[root@dvkrasnova guest]#
```

изменение прав

9. Использовали su для повышения прав до суперпользователя

10. Выполнили проверку правильности установки новых атрибутов и смены владельца файла simpleid2:

```
ls -l simpleid2
```

11. Запустили simpleid2 и id:

```
./simpleid2
id
```

```
[guest@dvkrasnova ~]$ ./simpleid2
e_uid=0, e_gid=1002
real_uid=1002, real_gid=1002
[guest@dvkrasnova ~]$ id
uid=1002(guest) gid=1002(guest) groups=1002(guest) context=unconfined_u:unconfin
ed_r:unconfined_t:s0-s0:c0.c1023
[guest@dvkrasnova ~]$
```

simpleid2

Результат выполнения программ теперь немного отличается

12. Проделали тоже самое относительно SetGID-бита.

13. Написали программу readfile.c

```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;
}
```

программа readfile

14. Откомпилировали её.

```
gcc readfile.c -o readfile
```

15. Сменили владельца у файла readfile.c и изменили права так, чтобы только суперпользователь (root) мог прочитать его, а guest не мог.

```
chown root:guest /home/guest/readfile.c
chmod 700 /home/guest/readfile.c
```

16. Проверили, что пользователь guest не может прочитать файл readfile.c.

17. Сменили у программы readfile владельца и установили SetU'D-бит.

18. Проверили, может ли программа readfile прочитать файл readfile.c

19. Проверили, может ли программа readfile прочитать файл /etc/shadow

```
guest@dvkrasnova ~]$ su
Password:
[root@dvkrasnova guest]# chown root:guest /home/guest/simpleid2
[root@dvkrasnova guest]# chmod u+s /home/guest/simpleid2
[root@dvkrasnova guest]# ls -l simpleid2
-rwsrwxr-x. 1 root guest 18312 Oct  6 14:30 simpleid2
[root@dvkrasnova guest]# chown root:guest /home/guest/readfile.c
[root@dvkrasnova guest]# chmod 700 /home/guest/readfile.c
[root@dvkrasnova guest]# chown root:root readfile
[root@dvkrasnova guest]# chmod -rwx readfile.c
[root@dvkrasnova guest]# chmod u+s readfile
[root@dvkrasnova guest]#
```

результат программы readfile

```
[guest@dvkrasnova ~]$ ./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@dvkrasnova ~]$ ./readfile /etc/shadow
```

результат программы readfile

2.2 Исследование Sticky-бита

1. Выяснили, установлен ли атрибут Sticky на директории /tmp:

```
ls -l / | grep tmp
```

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

```
echo "test" > /tmp/file01.txt
```

3. Просмотрели атрибуты у только что созданного файла и разрешили 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@dvkrasnova ~]$ ls -l / | grep tmp
drwxrwxrwt.  16 root root 4096 Oct  6 14:32 tmp
[guest@dvkrasnova ~]$ echo "test" > /tmp/file01.txt
[guest@dvkrasnova ~]$ ls -l /tmp/file01.txt
-rw-rw-r--. 1 guest guest 5 Oct  6 14:37 /tmp/file01.txt
[guest@dvkrasnova ~]$ chmod o+rw /tmp/file01.txt
[guest@dvkrasnova ~]$ ls -l /tmp/file01.txt
-rw-rw-rw-. 1 guest guest 5 Oct  6 14:37 /tmp/file01.txt
[guest@dvkrasnova ~]$
```

команды

Первоначально все группы имели право на чтение, а запись могли осуществлять все, кроме «остальных пользователей».

4. От пользователя (не являющегося владельцем) попробовали прочитать файл /file01.txt:

```
cat /file01.txt
```

5. От пользователя попробовали дозаписать в файл /file01.txt слово test3 командой:

```
echo "test2" >> /file01.txt
```

6. Проверили содержимое файла командой:

```
cat /file01.txt
```

В файле теперь записано:

```
Test
Test2
```

7. От пользователя попробовали записать в файл /tmp/file01.txt слово test4, стерев при этом всю имеющуюся в файле информацию командой. Для этого воспользовалась командой echo “test3” > /tmp/file01.txt

8. Проверили содержимое файла командой

```
cat /tmp/file01.txt
```

9. От пользователя попробовали удалить файл /tmp/file01.txt командой rm /tmp/file01.txt, однако получила отказ.

10. От суперпользователя командой выполнили команду, снимающую атрибут t (Sticky-бит) с директории /tmp:

```
chmod -t /tmp
```

Покинули режим суперпользователя командой exit.

11. От пользователя проверили, что атрибута t у директории /tmp нет:

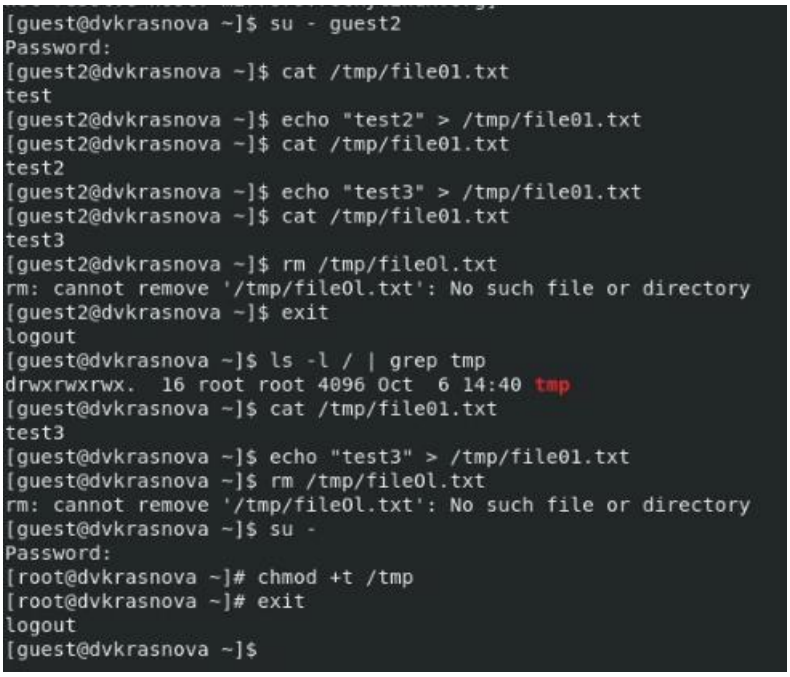
```
ls -l / | grep tmp
```

12. Повторили предыдущие шаги. Получилось удалить файл

13. Удалось удалить файл от имени пользователя, не являющегося его владельцем.

14. Повысили свои права до суперпользователя и вернули атрибут t на директорию /tmp :

```
su
chmod +t /tmp
exit
```



исследование Sticky-бита

3 Выводы

Изучили механизмы изменения идентификаторов, применения SetUID- и Sticky-битов. Получили практические навыки работы в консоли с дополнительными атрибутами. Также мы рассмотрели работу механизма смены идентификатора процессов пользователей и влияние бита Sticky на запись и удаление файлов.

Список литературы

1. КОМАНДА CHATTR В LINUX
2. chattr