Отчет по лабораторной работе №5

Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

Исаханян Эдуард Тигранович

2022 Sep 17th

Содержание

1 Цель работы **5**

2 Выполнение лабораторной работы **6**
2.1 Создание программы 6
2.2 Исследование Sticky-бита 10

3 Выводы **12**

Список литературы **13**

List of Tables

List of Figures

2.1 Программа simpleid.c 6
2.2 Компиляция и выполнение программы simpleid 6
2.3 Программа simpleid2.c 7
2.4 Компиляция и выполнение программы simpleid2 7
2.5 Смена пользователя. Установка SetUID-бита. Выполнение программы simpleid2 7
2.6 Установка SetGID-бита. Выполнение программы simpleid2 8
2.7 Программа readfile.c 8
2.8 Работа с программой readfile.c 8
2.9 Установка SetUID-бита на программу readfile 9
2.10 Программа readfile читает readfile.c 9
2.11 Программа readfile читает /etc/shadow 9
2.12 Исследование Sticky-бита от имени guest 10
2.13 Работа с file01.txt от имени guest2 при наличии Sticky-бита 10
2.14 Снятие Sticky-бита с /tmp 11
2.15 Работа с file01.txt от имени guest2 без Sticky-бита 11
2.16 Возвращение Sticky-бита на /tmp 11

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 Создание программы

Войдем в систему от имени пользователя guest. создадим программу simpleid.c по шаблону из методички. (рис. 2.1)

```
#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;
}
```

Figure 2.1: Программа simpleid.c

Скомпилируем программу и убедимся, что файл программы создан, после чего выполним программу и сравним с id. (рис. 2.2)

```
[guest@edikisakhayan dir2]$ gcc simpleid.c -o simpleid
[guest@edikisakhayan dir2]$ ./simpleid
uid=1001, gid=1001
[guest@edikisakhayan dir2]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfi
ned_r:unconfined_t:s0-s0:c0.c1023
```

Figure 2.2: Компиляция и выполнение программы simpleid

Полученный результат совпадает с id.

Усложним программу, добавив вывод действительных идентификаторов согласно шаблону из методички. (рис. 2.3)

```
#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 2.3: Программа simpleid2.c

Скомпилируем и запустим программу simpleid2.c. (рис. 2.4)

```
[guest@edikisakhayan dir2]$ gcc simpleid2.c -o simpleid2
[guest@edikisakhayan dir2]$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
```

Figure 2.4: Компиляция и выполнение программы simpleid2

От имени суперпользователя выполним команды: chown root:guest /home/guest/simpl и chmod u+s /home/guest/simpl. Выполним проверку правильности установки новых атрибутов и смены владельца файла simpl и запустим программу. (рис. 2.5)

```
[guest@edikisakhayan dir2]$ su
Пароль:
[root@edikisakhayan dir2]# chown root:guest /home/guest/dir2/simpleid2
[root@edikisakhayan dir2]# chmod u+s simpleid2
[root@edikisakhayan dir2]# ls -l simpleid2
-rwsrwxr-x. 1 root guest 8616 сен 17 19:34 simpleid2
[root@edikisakhayan dir2]# su guest
[guest@edikisakhayan dir2]$ ./simpleid2
e_uid=0, e_gid=1001
real_uid=1001, real_gid=1001
[guest@edikisakhayan dir2]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfi
ned_r:unconfined_t:s0-s0:c0.c1023
```

Figure 2.5: Смена пользователя. Установка SetUID-бита. Выполнение программы simpleid2

Проделаем то же самое относительно SetGID-бита. (рис. 2.6)

Figure 2.6: Установка SetGID-бита. Выполнение программы simpleid2

Создадим программу readfile.c по шаблону из методички. (рис. 2.7)

Figure 2.7: Программа readfile.c

Откомпилируем программу и сменим владельца и изменим права так, чтобы только суперпользователь (root) мог прочитать его, а guest не мог, и проверим это. (рис. 2.8)

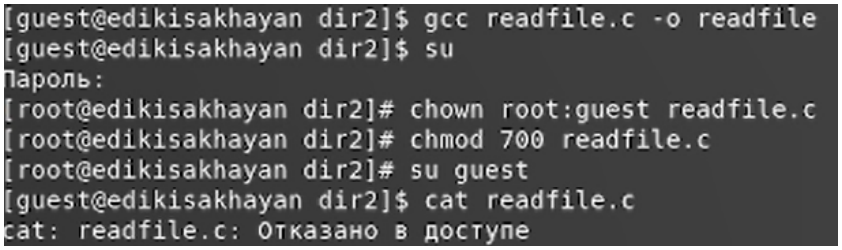

Figure 2.8: Работа с программой readfile.c

Установим SetUID-бит. (рис. 2.9)

```
[root@edikisakhayan dir2]# chown root:guest readfile
[root@edikisakhayan dir2]# chmod u+s readfile
```

Figure 2.9: Установка SetUID-бита на программу readfile

Проверим, может ли программа readfile прочитать файл readfile.c. (рис. 2.10)

```
[root@edikisakhayan dir2]# ./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 2.10: Программа readfile читает readfile.c

Проверим, может ли программа readfile прочитать файл /etc/shadow. (рис. 2.11)

```
chrony:!!:19245::::::
unbound:!!:19245::::::
qemu:!!:19245::::::
tss:!!:19245::::::
usbmuxd:!!:19245::::::
geoclue:!!:19245::::::
gluster:!!:19245::::::
gdm:!!:19245::::::
rpcuser:!!:19245::::::
nfsnobody:!!:19245::::::
gnome-initial-setup:!!:19245::::::
sshd:!!:19245::::::
avahi:!!:19245::::::
postfix:!!:19245::::::
ntp:!!:19245::::::
tcpdump:!!:19245::::::
edikisakhanyan:$6$m8bLiQqIgRCJUuyR$25jN.QZ9uYrarmAQEcWb5/TZYsKzT3uDZJtALh9BfOTu3
OGqmUW6yvWjrPGXvkdIYHpekIjCekA/6OZFkdlwG1::0:99999:7:::
vboxadd:!!:19245::::::
guest:$6$Er1dSDDk$BYqxG.Hz3V3qA6PS3D4Fpnc9HTjbLc5SQp8awW3RHXOIV4y22eWpaty51Y07Ui
kIc45BJGCL9zGeWj7rC1pam.:19248:0:99999:7:::
guest2:$6$pkIBHO.M$5NCX7imaoO6CB2Q4wFbflb1/oKnJPwhJtPUOgtoLKVsXEq6aj3mflqJS4HBAA
LyBqTFgHYLqxgAwF1BKVOP.G.:19252:0:99999:7:::
```

Figure 2.11: Программа readfile читает /etc/shadow

2.2 Исследование Sticky-бита

Выясним, установлен ли атрибут Sticky на директории /tmp, после чего от имени пользователя guest создал файл file01.txt в директории со словом test. Просмотрим атрибуты у только что созданного файла и разрешим чтение и запись для категории пользователей «все остальные». (рис. 2.12)

```
[root@edikisakhayan dir2]# su guest
[guest@edikisakhayan dir2]$ ls -l /| grep tmp
drwxrwxrwt.  32 root root 4096 сен 17 19:46 tmp
[guest@edikisakhayan dir2]$ echo "test" > /tmp/file01.txt
[guest@edikisakhayan dir2]$ ls -l /tmp/file01.txt
-rw-rw-r--. 1 guest guest 5 сен 17 19:48 /tmp/file01.txt
[guest@edikisakhayan dir2]$ chmod o+rw /tmp/file01.txt
[guest@edikisakhayan dir2]$ ls -l /tmp/file01.txt
-rw-rw-rw-. 1 guest guest 5 сен 17 19:48 /tmp/file01.txt
```

Figure 2.12: Исследование Sticky-бита от имени guest

От имени пользователя попробовал разные операции над файлом file01.txt. (рис. 2.13)

```
[guest@edikisakhayan ~]$ su guest2
Пароль:
[guest2@edikisakhayan guest]$ cat /tmp/file01.txt
test
[guest2@edikisakhayan guest]$ echo "test2" >> /tmp/file01.txt
[guest2@edikisakhayan guest]$ cat /tmp/file01.txt
test
test2
[guest2@edikisakhayan guest]$ echo "test3" > /tmp/file01.txt
[guest2@edikisakhayan guest]$ cat /tmp/file01.txt
test3
[guest2@edikisakhayan guest]$ rm /tmp/file01.txt
rm: невозможно удалить «/tmp/file01.txt»: Операция не позволена
```

Figure 2.13: Работа с file01.txt от имени guest2 при наличии Sticky-бита

Повысим свои права до суперпользователя, и выполним после этого команду, снимающую атрибут t (Sticky-бит) с директории и покиним режим суперпользователя. (рис. 2.14)

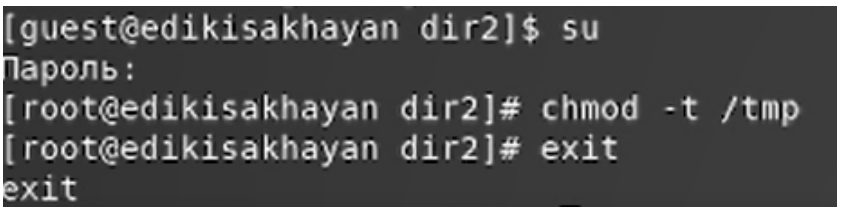

Figure 2.14: Снятие Sticky-бита с /tmp

Проверим те же операции после стнятия атрибута t. (рис. 2.15)

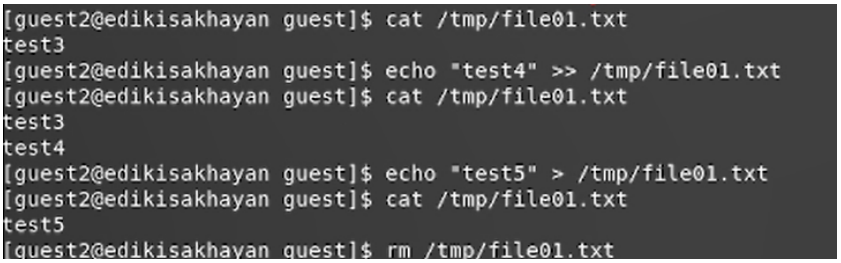

Figure 2.15: Работа с file01.txt от имени guest2 без Sticky-бита

Нам удалось удалить файл от имени пользователя, не являющегося его владельцем.

Повысим свои права до суперпользователя, и выполним после этого команду, добавляющее атрибут t (Sticky-бит) к директории и покиним режим суперпользователя. (рис. 2.16)

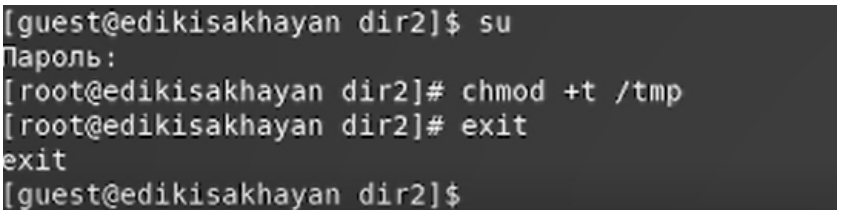

Figure 2.16: Возвращение Sticky-бита на /tmp

3 Выводы

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

Список литературы

1. Методические материалы к лабораторной работе, представленные на сайте “ТУИС РУДН” https://esystem.rudn.ru/

::: {#refs} :::