

Отчёт по лабораторной работе №5

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

Булаев Максим Александрович НПИбд-01-19

Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
2.1	Подготовка	5
2.2	Изучение механики SetUID	6
2.3	Исследование Sticky-бита	9
3	Выводы	13
	Список литературы	14

List of Figures

2.1	подготовка к работе	5
2.2	программа simpleid	6
2.3	результат программы simpleid	6
2.4	программа simpleid2	7
2.5	результат программы simpleid2	8
2.6	программа readfile	8
2.7	результат программы readfile	9
2.8	исследование Sticky-бита	12

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 Подготовка

1. Для выполнения части заданий требуются средства разработки приложений. Проверил наличие установленного компилятора gcc командой `gcc -v`: компилятор обнаружен.
2. Чтобы система защиты SELinux не мешала выполнению заданий работы, отключил систему запретов до очередной перезагрузки системы командой `setenforce 0`.
3. Команда `getenforce` вывела `Permissive`.

```
[guest@mabulaev ~]$ gcc -v
Using built-in specs.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/usr/libexec/gcc/x86_64-redhat-linux/11/lto-wrapper
OFFLOAD_TARGET_NAMES=nvptx-none
OFFLOAD_TARGET_DEFAULT=1
Target: x86_64-redhat-linux
Configured with: ../configure --enable-bootstrap --enable-host-pie --enable-host-bind-n
ow --enable-languages=c,c++,fortran,lto --prefix=/usr --mandir=/usr/share/man --infodir
=/usr/share/info --with-bugurl=https://bugs.rockylinux.org/ --enable-shared --enable-th
reads=posix --enable-checking=release --enable-multilib --with-system-zlib --enable-_c
xa_atexit --disable-libunwind-exceptions --enable-gnu-unique-object --enable-linker-bui
ld-id --with-gcc-major-version-only --with-linker-hash-style=gnu --enable-plugin --enab
le-initfini-array --without-isl --enable-offload-targets=nvptx-none --without-cuda-driv
er --enable-gnu-indirect-function --enable-cet --with-tune=generic --with-arch_64=x86-6
4-v2 --with-arch_32=x86-64 --build=x86_64-redhat-linux --with-build-config=bootstrap-lt
o --enable-link-serialization=1
Thread model: posix
Supported LTO compression algorithms: zlib zstd
gcc version 11.2.1 20220127 (Red Hat 11.2.1-9) (GCC)
[guest@mabulaev ~]$ getenforce
Permissive
```

Figure 2.1: подготовка к работе

2.2 Изучение механики SetUID

1. Вошёл в систему от имени пользователя guest.
2. Написал программу simpleid.c.

```
1 #include <sys/types.h>
2 #include <unistd.h>
3 #include <stdio.h>
4
5 int
6 main ()
7 {
8     uid_t uid = geteuid ();
9     gid_t gid = getegid ();
10    printf ("uid=%d, gid=%d\n", uid, gid);
11    return 0;
12 }
```

Figure 2.2: программа simpleid

3. Скомпилировал программу и убедился, что файл программы создан: gcc simpleid.c -o simpleid
4. Выполнил программу simpleid командой ./simpleid
5. Выполнил системную программу id с помощью команды id. uid и gid совпадает в обеих программах

```
[guest@mabulaev lab5]$ touch simpleid.c
[guest@mabulaev lab5]$ gedit simpleid.c
[guest@mabulaev lab5]$ gcc simpleid.c -o simpleid
[guest@mabulaev lab5]$ ./simpleid
uid=1001, gid=1001
[guest@mabulaev lab5]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@mabulaev lab5]$
```

Figure 2.3: результат программы simpleid

6. Усложнил программу, добавив вывод действительных идентификаторов.

```

1 #include <sys/types.h>
2 #include <unistd.h>
3 #include <stdio.h>
4
5 int
6 main ()
7 {
8     uid_t real_uid = getuid ();
9     uid_t e_uid = geteuid ();
10
11     gid_t real_gid = getgid ();
12     gid_t e_gid = getegid ();
13
14     printf ("e_uid=%d, e_gid=%d\n", e_uid, e_gid);
15     printf ("real_uid=%d, real_gid=%d\n", real_uid, real_gid);
16     return 0;
17 }

```

Figure 2.4: программа simpleid2

7. Скомпилировал и запустил simpleid2.c:

```

gcc simpleid2.c -o simpleid2
./simpleid2

```

8. От имени суперпользователя выполнил команды:

```

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

```

9. Использовал su для повышения прав до суперпользователя.

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

```

ls -l simpleid2

```

11. Запустил simpleid2 и id:

```

./simpleid2
id

```

Результат выполнения программ теперь немного отличается.

12. Проделал тоже самое относительно SetGID-бита.

```
[root@mabulaev guest]# chown root:guest /home/guest/simpleid2
[root@mabulaev guest]# chown u+s /home/guest/simpleid2
chown: invalid user: 'u+s'
[root@mabulaev guest]# chmod u+s /home/guest/simpleid2
[root@mabulaev guest]# su
[root@mabulaev guest]# ls -l simpleid2
-rwsrwxr-x. 1 root guest 26008 Oct  8 15:16 simpleid2
[root@mabulaev guest]# ./simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@mabulaev guest]# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@mabulaev guest]# chmod u+g /home/guest/simpleid2
[root@mabulaev guest]# ./simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@mabulaev guest]# ls -l simpleid2
-rwsrwxr-x. 1 root guest 26008 Oct  8 15:16 simpleid2
[root@mabulaev guest]# chmod u-g /home/guest/simpleid2
[root@mabulaev guest]# chmod g+s /home/guest/simpleid2
[root@mabulaev guest]# ./simpleid2
e_uid=0, e_gid=1001
real_uid=0, real_gid=0
[root@mabulaev guest]# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@mabulaev guest]# touch
```

Figure 2.5: результат программы simpleid2

13. Написал программу 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     int fd = open (argv[1], O_RDONLY);
14     do
15     {
16         bytes_read = read (fd, buffer, sizeof (buffer));
17         for (i = 0; i < bytes_read; ++i) printf("%c", buffer[i]);
18     }
19     while (bytes_read == sizeof (buffer));
20     close (fd);
21     return 0;
22 }
```

Figure 2.6: программа readfile

14. Откомпилировал её.

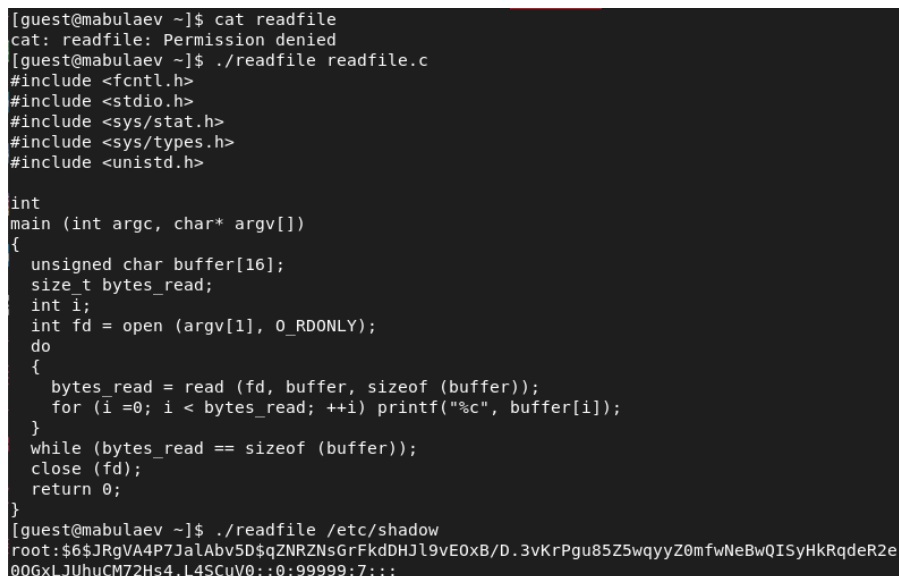

```
gcc readfile.c -o readfile
```

15. Сменил владельца у файла readfile.c и изменил 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. Проверил, что пользователь guest не может прочитать файл readfile.c.
17. Сменил у программы readfile владельца и установил SetU'D-бит.
18. Проверил, может ли программа readfile прочитать файл readfile.c
19. Проверил, может ли программа readfile прочитать файл /etc/shadow



```
[guest@mabulaev ~]$ cat readfile
cat: readfile: Permission denied
[guest@mabulaev ~]$ ./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@mabulaev ~]$ ./readfile /etc/shadow
root:$6$JRgVA4P7Ja1Abv5D$qZNRZnsGrFkdDHJl9vE0xB/D.3vKrPgu85Z5wqyyZ0mfWNeBwQISyHkRqdeR2e
00GxLJUhuCM72Hs4.L4SCuV0::0:99999:7:::
```

Figure 2.7: результат программы readfile

2.3 Исследование Sticky-бита

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

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

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

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

3. Просмотрел атрибуты у только что созданного файла и разрешил чтение и запись для категории пользователей «все остальные»:

```
ls -l /tmp/file01.txt  
chmod o+rw /tmp/file01.txt  
ls -l /tmp/file01.txt
```

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

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

```
cat /file01.txt
```

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

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

6. Проверил содержимое файла командой:

```
cat /file01.txt
```

В файле теперь записано:

```
Test  
Test2
```

7. От пользователя попробовал записать в файл /tmp/file01.txt слово test4, стеревав при этом всю имеющуюся в файле информацию командой. Для этого воспользовалась командой `echo "test3" > /tmp/file01.txt`

8. Проверил содержимое файла командой

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

9. От пользователя попробовал удалить файл /tmp/file01.txt командой `rm /tmp/file01.txt`, однако получил отказ.

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

```
chmod -t /tmp
```

Покинул режим суперпользователя командой `exit`.

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

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

12. Повторил предыдущие шаги. Получилось удалить файл.

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

14. Повысил свои права до суперпользователя и вернул атрибут t на директорию /tmp :

```
su
```

```
chmod +t /tmp
```

```
exit
```

```

[guest@mabulaev tmp]$ chmod o+rw /tmp/file01.txt
[guest@mabulaev tmp]$ ls -l /tmp/file01.txt
-rw-rw-rw-. 1 guest guest 5 Oct  8 15:51 /tmp/file01.txt
[guest@mabulaev tmp]$ su guest2
Password:
[guest2@mabulaev tmp]$ cat /tmp/file01.txt
test
[guest2@mabulaev tmp]$ echo "test2" > /tmp/file01.txt
[guest2@mabulaev tmp]$ cat /tmp/file01.txt
test2
[guest2@mabulaev tmp]$ rm /tmp/file01.txt
rm: cannot remove '/tmp/file01.txt': Operation not permitted
[guest2@mabulaev tmp]$ su -
Password:
[root@mabulaev ~]# chmod -t /tmp
[root@mabulaev ~]# exit
logout
[guest2@mabulaev tmp]$ ls -l / |grep tmp
drwxrwxrwx. 15 root root 4096 Oct  8 15:56 tmp
[guest2@mabulaev tmp]$ cat /tmp/file01.txt
test2
[guest2@mabulaev tmp]$ echo "test3" > /tmp/file01.txt
[guest2@mabulaev tmp]$ cat /tmp/file01.txt
test3
[guest2@mabulaev tmp]$ rm /tmp/file01.txt
[guest2@mabulaev tmp]$ su -
Password:
[root@mabulaev ~]# chmod +t /tmp

```

Figure 2.8: исследование Sticky-бита

3 Выводы

Таким образом, в ходе лабораторной работы я изучил механизмы изменения идентификаторов, применения SetUID- и Sticky-битов, получил практические навыки работы в консоли с дополнительными атрибутами, а также рассмотрел работу механизма смены идентификатора процессов пользователей и влияние бита Sticky на запись и удаление файлов.

Список литературы

1. КОМАНДА CHATTR В LINUX
2. chattr