

Отчёт по лабораторной работе №5

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

Боровиков Даниил Александрович НПИбд-01-22

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	10
3	Выводы	14
	Список литературы	15

List of Figures

2.1	подготовка к работе	6
2.2	программа simpleid	6
2.3	результат программы simpleid	7
2.4	программа simpleid2	7
2.5	результат программы simpleid2	8
2.6	программа readfile	9
2.7	результат программы readfile	10
2.8	исследование Sticky-бита	13

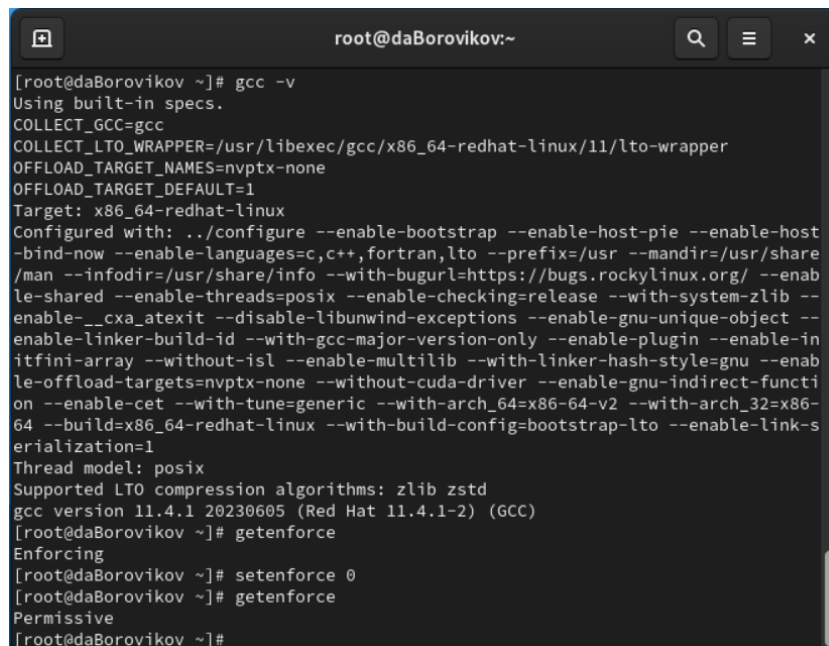
1 Цель работы

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. Проверили наличие установленного компилятора gcc командой `gcc -v`:
компилятор обнаружен.
2. Чтобы система защиты SELinux не мешала выполнению заданий работы, отключили систему запретов до очередной перезагрузки системы командой `setenforce 0`:
3. Команда `getenforce` вывела `Permissive`:



```
root@daBorovikov:~  
[root@daBorovikov ~]# 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-now --enable-languages=c,c++,fortran,lto --prefix=/usr --mandir=/usr/share  
/man --infodir=/usr/share/info --with-bugurl=https://bugs.rockylinux.org/ --enab  
le-shared --enable-threads=posix --enable-checking=release --with-system-zlib --  
enable-__cxa_atexit --disable-libunwind-exceptions --enable-gnu-unique-object --  
enable-linker-build-id --with-gcc-major-version-only --enable-plugin --enable-in  
itfini-array --without-isl --enable-multilib --with-linker-hash-style=gnu --enab  
le-offload-targets=nvptx-none --without-cuda-driver --enable-gnu-indirect-functi  
on --enable-cet --with-tune=generic --with-arch_64=x86-64-v2 --with-arch_32=x86-  
64 --build=x86_64-redhat-linux --with-build-config=bootstrap-lto --enable-link-s  
erialization=1  
Thread model: posix  
Supported LTO compression algorithms: zlib zstd  
gcc version 11.4.1 20230605 (Red Hat 11.4.1-2) (GCC)  
[root@daBorovikov ~]# getenforce  
Enforcing  
[root@daBorovikov ~]# setenforce 0  
[root@daBorovikov ~]# getenforce  
Permissive  
[root@daBorovikov ~]#
```

Figure 2.1: подготовка к работе

2.2 Изучение механики SetUID

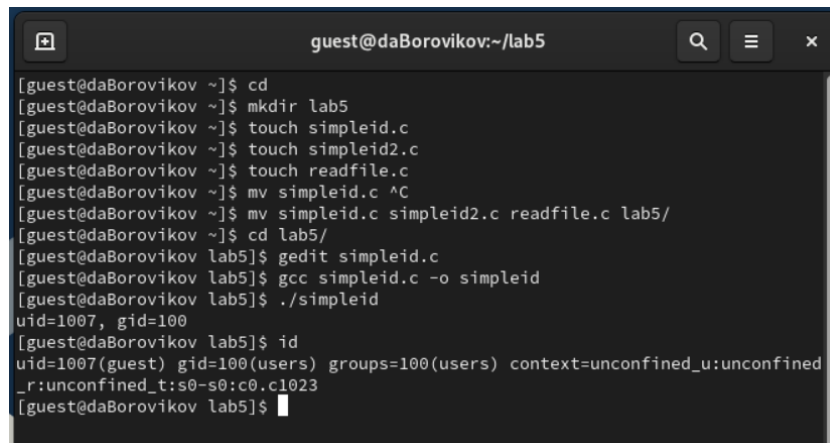
1. Вошли в систему от имени пользователя guest.
2. Написали программу simpleid.c.



```
*simpleid.c  
~/lab5  
1 #include <sys/types.h>  
2 #include <unistd.h>  
3 #include <stdio.h>  
4 int  
5 main ()  
6 {  
7     uid_t uid = geteuid ();  
8     gid_t gid = getegid ();  
9     printf ("uid=%d, gid=%d\n", uid, gid);  
10    return 0;  
11 }  
12 |
```

Figure 2.2: программа simpleid

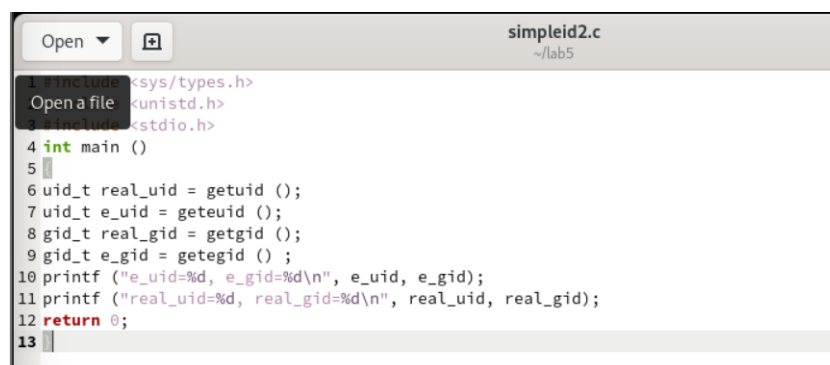
3. Скомпилировали программу и убедились, что файл программы создан: `gcc simpleid.c -o simpleid`
4. Выполнили программу `simpleid` командой `./simpleid`
5. Выполнили системную программу `id` с помощью команды `id`. `uid` и `gid` совпадает в обеих программах



```
guest@daBorovikov:~/lab5
[guest@daBorovikov ~]$ cd
[guest@daBorovikov ~]$ mkdir lab5
[guest@daBorovikov ~]$ touch simpleid.c
[guest@daBorovikov ~]$ touch simpleid2.c
[guest@daBorovikov ~]$ touch readfile.c
[guest@daBorovikov ~]$ mv simpleid.c ^C
[guest@daBorovikov ~]$ mv simpleid2.c readfile.c lab5/
[guest@daBorovikov ~]$ cd lab5/
[guest@daBorovikov lab5]$ gedit simpleid.c
[guest@daBorovikov lab5]$ gcc simpleid.c -o simpleid
[guest@daBorovikov lab5]$ ./simpleid
uid=1007, gid=100
[guest@daBorovikov lab5]$ id
uid=1007(guest) gid=100(users) groups=100(users) context=unconfined_u:unconfined
_r:unconfined_t:s0-s0:c0.c1023
[guest@daBorovikov lab5]$
```

Figure 2.3: результат программы `simpleid`

6. Усложнили программу, добавив вывод действительных идентификаторов.



```
simpleid2.c
~/lab5
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>
4 int main ()
5 {
6     uid_t real_uid = getuid ();
7     uid_t e_uid = geteuid ();
8     gid_t real_gid = getgid ();
9     gid_t e_gid = getegid ();
10    printf ("e_uid=%d, e_gid=%d\n", e_uid, e_gid);
11    printf ("real_uid=%d, real_gid=%d\n", real_uid, real_gid);
12    return 0;
13 }
```

Figure 2.4: 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
```

9. Использовали su для повышения прав до суперпользователя

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

```
ls -l simpleid2
```

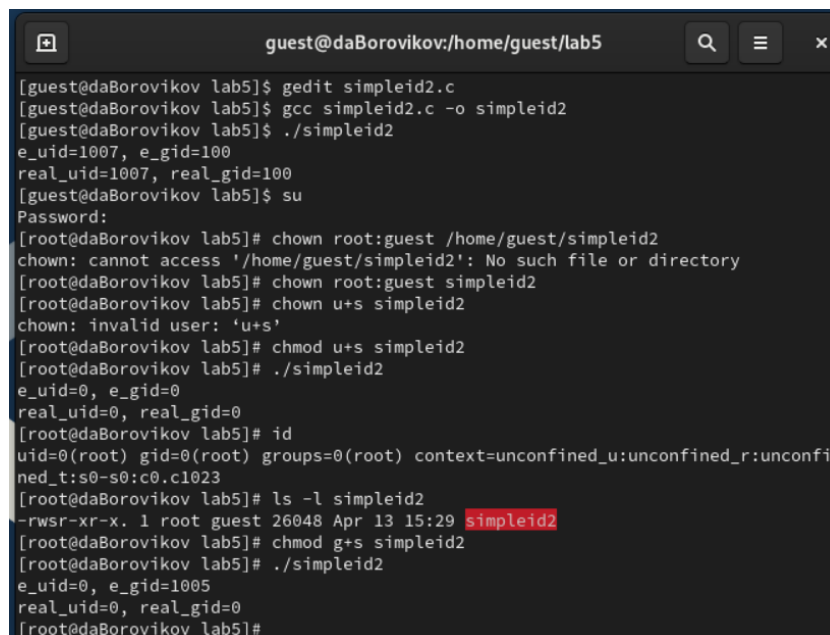
11. Запустили simpleid2 и id:

```
./simpleid2
```

```
id
```

Результат выполнения программ теперь немного отличается

12. Проделали тоже самое относительно SetGID-бита.



```
guest@daBorovikov:/home/guest/lab5
[guest@daBorovikov lab5]$ gedit simpleid2.c
[guest@daBorovikov lab5]$ gcc simpleid2.c -o simpleid2
[guest@daBorovikov lab5]$ ./simpleid2
e_uid=1007, e_gid=100
real_uid=1007, real_gid=100
[guest@daBorovikov lab5]$ su
Password:
[root@daBorovikov lab5]# chown root:guest /home/guest/simpleid2
chown: cannot access '/home/guest/simpleid2': No such file or directory
[root@daBorovikov lab5]# chown root:guest simpleid2
[root@daBorovikov lab5]# chown u+s simpleid2
chown: invalid user: 'u+s'
[root@daBorovikov lab5]# chmod u+s simpleid2
[root@daBorovikov lab5]# ./simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@daBorovikov lab5]# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@daBorovikov lab5]# ls -l simpleid2
-rwsr-xr-x. 1 root guest 26048 Apr 13 15:29 simpleid2
[root@daBorovikov lab5]# chmod g+s simpleid2
[root@daBorovikov lab5]# ./simpleid2
e_uid=0, e_gid=1005
real_uid=0, real_gid=0
[root@daBorovikov lab5]#
```

Figure 2.5: результат программы simpleid2

13. Написали 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 int main (int argc, char* argv[])
7 {
8     unsigned char buffer[16];
9     size_t bytes_read;
10    int i;
11    int fd = open (argv[1], O_RDONLY);
12    do
13    {
14        bytes_read = read (fd, buffer, sizeof (buffer));
15        for (i = 0; i < bytes_read; ++i) printf("%c", buffer[i]);
16    }
17    while (bytes_read == sizeof (buffer));
18    close (fd);
19    return 0;
20 }
21 |
```

Figure 2.6: 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

```
[root@daBorovikov lab5]# gcc readfile.c -o readfile
[root@daBorovikov lab5]# chmod root:root readfile
chmod: invalid mode: 'root:root'
Try 'chmod --help' for more information.
[root@daBorovikov lab5]# chown root:root readfile
[root@daBorovikov lab5]# chmod o-r readfile.c
[root@daBorovikov lab5]# chmod g-r readfile.c
[root@daBorovikov lab5]# chmod u-r readfile.c
[root@daBorovikov lab5]# chmod u+s readfile.c
[root@daBorovikov lab5]# exit
exit
[guest@daBorovikov lab5]$ cat readfile.c
cat: readfile.c: Permission denied
[guest@daBorovikov lab5]$ ./readfile readfile.c
```

Figure 2.7: результат программы readfile

2.3 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
```

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
```

```
guest@daBorovikov:~  
[guest@daBorovikov ~]$ ls -l | grep tmp  
[guest@daBorovikov ~]$ echo "test" > /tmp/file01.txt  
[guest@daBorovikov ~]$ ls -l /tmp/file01.txt  
-rw-r--r--. 1 guest users 5 Apr 13 15:46 /tmp/file01.txt  
[guest@daBorovikov ~]$ chmod o+rw /tmp/file01.txt  
[guest@daBorovikov ~]$ ls -l /tmp/file01.txt  
-rw-r--rw-. 1 guest users 5 Apr 13 15:46 /tmp/file01.txt  
[guest@daBorovikov ~]$  
  
guest2@daBorovikov:/home/guest  
[guest@daBorovikov ~]$ guest2  
bash: guest2: command not found...  
[guest@daBorovikov ~]$ su guest2  
Password:  
[guest2@daBorovikov guest]$ cat /tmp/file01.txt  
test  
[guest2@daBorovikov guest]$ echo "test2" > /tmp/file01.txt  
bash: /tmp/file01.txt: Permission denied  
[guest2@daBorovikov guest]$ cat /tmp/file01.txt  
test  
[guest2@daBorovikov guest]$ echo "test3" > /tmp/file01.txt  
bash: /tmp/file01.txt: Permission denied  
[guest2@daBorovikov guest]$ cat /tmp/file01.txt  
test  
[guest2@daBorovikov guest]$ rm /tmp/file01.txt  
rm: cannot remove '/tmp/file01.txt': No such file or directory  
[guest2@daBorovikov guest]$ su -  
Password:  
[root@daBorovikov ~]# chmod -t /tmp  
[root@daBorovikov ~]# exit  
logout  
[guest2@daBorovikov guest]$ ls -l / | grep tmp  
drwxrwxrwx. 18 root root 4096 Apr 13 15:51 tmp  
[guest2@daBorovikov guest]$ echo "test2" > /tmp/file01.txt  
bash: /tmp/file01.txt: Permission denied  
[guest2@daBorovikov guest]$ cat /tmp/file01.txt  
test  
[guest2@daBorovikov guest]$ echo "test3" > /tmp/file01.txt  
bash: /tmp/file01.txt: Permission denied  
[guest2@daBorovikov guest]$ cat /tmp/file01.txt  
test  
[guest2@daBorovikov guest]$ rm /tmp/file01.txt  
rm: cannot remove '/tmp/file01.txt': No such file or directory  
[guest2@daBorovikov guest]$ su -  
Password:  
su: Authentication failure  
[guest2@daBorovikov guest]$ su -  
Password:  
[root@daBorovikov ~]# chmod +t /tmp  
[root@daBorovikov ~]# exit  
logout  
[guest2@daBorovikov guest]$
```

Figure 2.8: 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