

Отчет по лабораторной работе №5

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

Гурбангельдиев Мухаммет НФИбд-03-18

Содержание

1	Цель работы	3
2	Последовательность выполнения работы	4
2.1	Создание программы	4
2.2	Исследование Sticky-бита	11
3	Выводы	15

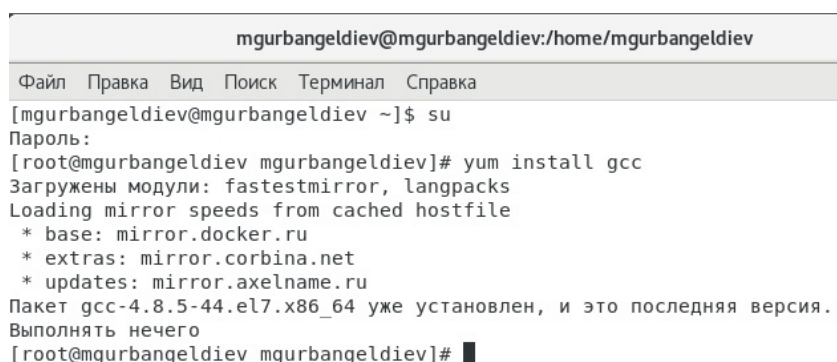
1 Цель работы

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 Последовательность выполнения работы

2.1 Создание программы

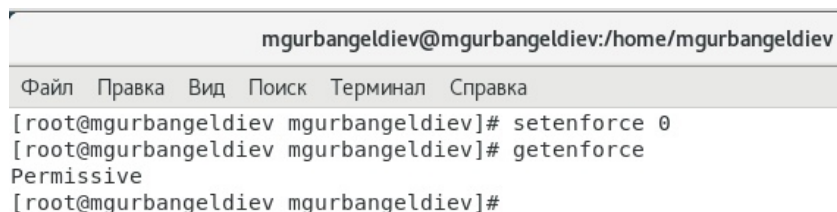
Для начала нам следовало установить компилятор gcc. (рис. 2.1)



```
mgurbangeldiev@mgurbangeldiev:/home/mgurbangeldiev
Файл  Правка  Вид  Поиск  Терминал  Справка
[mgurbangeldiev@mgurbangeldiev ~]$ su
Пароль:
[root@mgurbangeldiev mgurbangeldiev]# yum install gcc
Загружены модули: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: mirror.docker.ru
 * extras: mirror.corbina.net
 * updates: mirror.axelname.ru
Пакет gcc-4.8.5-44.el7.x86_64 уже установлен, и это последняя версия.
Выполнять нечего
[root@mgurbangeldiev mgurbangeldiev]#
```

Figure 2.1: Компилятор gcc

Чтобы защита SELinux не мешала выполнению заданий работы, мы отключили ее. (рис. 2.2)



```
mgurbangeldiev@mgurbangeldiev:/home/mgurbangeldiev
Файл  Правка  Вид  Поиск  Терминал  Справка
[root@mgurbangeldiev mgurbangeldiev]# setenforce 0
[root@mgurbangeldiev mgurbangeldiev]# getenforce
Permissive
[root@mgurbangeldiev mgurbangeldiev]#
```

Figure 2.2: Отключение защиты

1. Войдите в систему от имени пользователя guest.
2. Создайте программу simpleid.c: (рис. 2.3) (рис. 2.4)

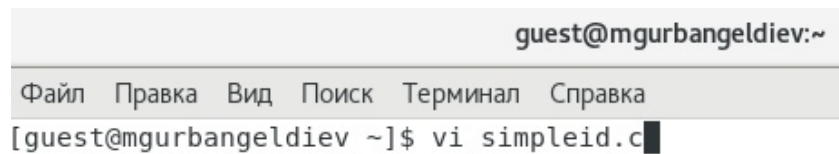


Figure 2.3: Программа simpleid.c

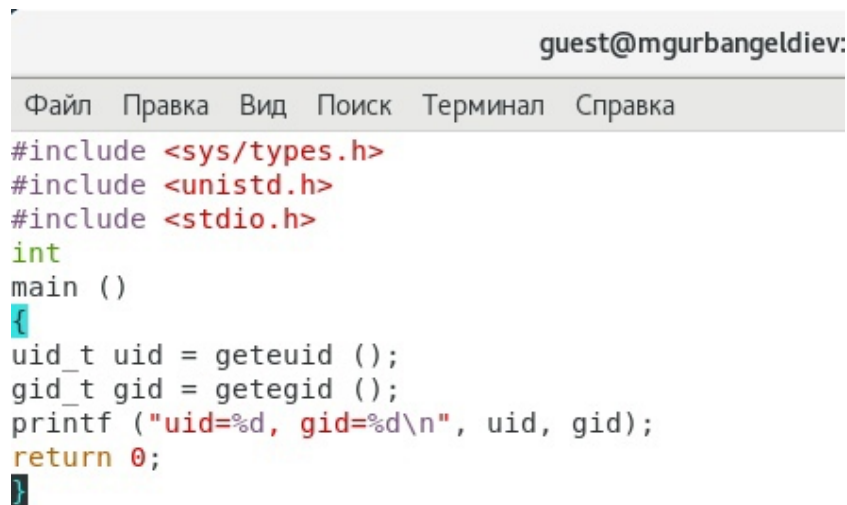


Figure 2.4: Программа simpleid.c

3. Скомпилируйте программу и убедитесь, что файл программы создан:

```
gcc simpleid.c -o simpleid
```

4. Выполните программу simpleid:

```
./simpleid
```

5. Выполните системную программу id:

```
id
```

и сравните полученный вами результат с данными предыдущего пункта задания. (рис. 2.5)

```
guest@mgurbangeldiev:~  
Файл Правка Вид Поиск Терминал Справка  
[guest@mgurbangeldiev ~]$ vi simpleid.c  
[guest@mgurbangeldiev ~]$ gcc simpleid.c -o simpleid  
[guest@mgurbangeldiev ~]$ ./simpleid  
uid=1001, gid=1001  
[guest@mgurbangeldiev ~]$ id  
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfi  
ned_r:unconfined_t:s0-s0:c0.c1023  
[guest@mgurbangeldiev ~]$
```

Figure 2.5: Компиляция и выполнения программы

6. Усложните программу, добавив вывод действительных идентификаторов:
(рис. 2.6) (рис. 2.7)

```
[guest@mgurbangeldiev ~]$ vi simpleid2.c
```

Figure 2.6: Программа simpleid2.c

```
guest@mgurbangeldiev:~  
Файл Правка Вид Поиск Терминал Справка  
#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 2.7: Программа simpleid2.c

7. Скомпилируйте и запустите simpleid2.c:

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

```
./simpleid2 (рис. 2.8)
```

```
[guest@mgurbangeldiev ~]$ gcc simpleid2.c -o simpleid2
[guest@mgurbangeldiev ~]$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
[guest@mgurbangeldiev ~]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfi
ned_r:unconfined_t:s0-s0:c0.c1023
[guest@mgurbangeldiev ~]$
```

Figure 2.8: Компиляция и выполнения программы

8. От имени суперпользователя выполните команды:

`chown root:guest /home/guest/simpleid2`

`chmod u+s /home/guest/simpleid2` (рис. 2.9)

```
[root@mgurbangeldiev mgurbangeldiev]# chown root:guest /home/guest/simpleid2
[root@mgurbangeldiev mgurbangeldiev]# chmod u+s /home/guest/simpleid2
```

Figure 2.9: Смена пользователя и установка SetU'D-бита

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

Команда `sudo` позволяет пользователям выполнять указанные программы с административными привилегиями без ввода пароля суперпользователя `root`.

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

`ls -l simpleid2`

11. Запустите `simpleid2` и `id`:

`./simpleid2`

`id`

Сравните результаты. (рис. 2.10)

```
[root@mgurbangeldiev guest]# ls -l simpleid2
-rwsrwxr-x. 1 root guest 8616 ноя 13 17:02 simpleid2
[root@mgurbangeldiev guest]# ./simpleid2
e_uid=0, e_gid=1001
real_uid=0, real_gid=0
[root@mgurbangeldiev guest]# id
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:unconf
ined_t:s0-s0:c0.c1023
```

Figure 2.10: Проверка правильности установки новых атрибутов

12. Прodelайте тоже самое относительно SetGID-бита. (рис. 2.11) (рис. 2.12)

```
[root@mgurbangeldiev mgurbangeldiev]# chown root:guest /home/guest/simpleid2
[root@mgurbangeldiev mgurbangeldiev]# chmod u+s /home/guest/simpleid2
[root@mgurbangeldiev mgurbangeldiev]# chmod g+s /home/guest/simpleid2
[root@mgurbangeldiev mgurbangeldiev]# █
```

Figure 2.11: Проверка правильности установки новых атрибутов

```
[root@mgurbangeldiev guest]# ls -l simpleid2
-rwsrwxr-x. 1 root guest 8616 ноя 13 17:02 simpleid2
[root@mgurbangeldiev guest]# ./simpleid2
e_uid=0, e_gid=1001
real_uid=0, real_gid=0
[root@mgurbangeldiev guest]# id
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
_
```

Figure 2.12: Проверка правильности установки новых атрибутов

13. . Создайте программу readfile.c: (рис. 2.13)

```
#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 2.13: Программа readfile.c

14. Откомпилируйте её.

gcc readfile.c -o readfile (рис. 2.14)


```
[root@mgurbangeldiev guest]# vi readfile.c
[root@mgurbangeldiev guest]# gcc readfile.c -o readfile
[root@mgurbangeldiev guest]# █
```

Figure 2.14: Программа readfile

15. 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. 2.15)

```
[root@mgurbangeldiev mgurbangeldiev]# chown root /home/guest/readfile.c
[root@mgurbangeldiev mgurbangeldiev]# chmod u+x /home/guest/readfile.c
[root@mgurbangeldiev mgurbangeldiev]# chmod g-rw /home/guest/readfile.c
[root@mgurbangeldiev mgurbangeldiev]# chmod o-r /home/guest/readfile.c
```

Figure 2.15: Смена владельца и изменения прав

16. Проверьте, что пользователь guest не может прочитать файл readfile.c. (рис. 2.16)

```
[root@mgurbangeldiev guest]# gcc readfile.c -o readfile
[root@mgurbangeldiev guest]# ls -l readfile.c
-rwx-----. 1 root root 402 ноя 13 16:07 readfile.c
[root@mgurbangeldiev guest]# █
```

Figure 2.16: Проверка на правильность

17. Смените у программы readfile владельца и установите SetU'D-бит. (рис. 2.17)

```
[root@mgurbangeldiev mgurbangeldiev]# chmod u+x /home/guest/readfile.c
[root@mgurbangeldiev mgurbangeldiev]# chmod g-rw /home/guest/readfile.c
[root@mgurbangeldiev mgurbangeldiev]# chmod o-r /home/guest/readfile.c
[root@mgurbangeldiev mgurbangeldiev]# chown root /home/guest/readfile.c
[root@mgurbangeldiev mgurbangeldiev]# chmod u+s /home/guest/readfile.c
[root@mgurbangeldiev mgurbangeldiev]# chmod u+s /home/guest/readfile.c
```

Figure 2.17: Смена пользователя и установка SetU'D-бита

18. Проверьте, может ли программа readfile прочитать файл readfile.c? (рис. 2.18) (рис. 2.19)

```
[root@mgurbangeldiev guest]# ls -l readfile
-rwxr-xr-x. 1 root root 8552 ноя 13 16:07 readfile
[root@mgurbangeldiev guest]# █
```

Figure 2.18: Проверка на правильность

```

#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 2.19: Чтения файла

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

Отразите полученный результат и ваши объяснения в отчёте. (рис. 2.20)

```

guest@mgurbangeldiev:/home/guest
Файл Правка Вид Поиск Терминал Справка
unbound!!!:18887:...:
qemu!!!:18887:...:
tss!!!:18887:...:
usbmuxd!!!:18887:...:
geoclue!!!:18887:...:
gluster!!!:18887:...:
gdm!!!:18887:...:
rpcuser!!!:18887:...:
nfsnobody!!!:18887:...:
gnome-initial-setup!!!:18887:...:
sshd!!!:18887:...:
avahi!!!:18887:...:
postfix!!!:18887:...:
ntp!!!:18887:...:
lcpdump!!!:18887:...:
mgurbangeldiev:S6Ss9UdhoKhhrPwRLLpSBvCz5H9dJJZWD0AASZTp.sGZaMkPESoXmp.MSySvpqQc
alYkhUZjiyvjJp5zQHJ2guH0KmakxDSnAUKuccpC/::0:99999:7:::
vboxadd!!!:18887:...:
guest:S6Sw/y6A9e2SKPY9LyvSk20ZLgtWwIt0IzoezDmB4xnHz0gTBqJvPSA1aex5HzbqYNKWj2Qu
JL7q02YN/bz8uxYMCmsNylD.:18902:0:99999:7:::
jupyter.:18912:0:99999:7:::
guest2:S6SgcHwF3.TS2no.PzGhM24EC/TM8dfJI2g0fbyHTFM5wX24WgmLPkN5enPGMEuxHIY0
sVSMcQ0H2/J0VvEDtjRWlqb0/:18914:0:99999:7:::
[root@mgurbangeldiev guest]# exi

```

Figure 2.20: Чтения файла

2.2 Исследование Sticky-бита

1. Выясните, установлен ли атрибут Sticky на директории /tmp, для чего выполните команду

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

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

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

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

```
ls -l /tmp/file01.txt
```

```
chmod o+rw /tmp/file01.txt
```

```
ls -l /tmp/file01.txt (рис. 2.21)
```

```
[root@mgurbangeldiev guest]# ls -l / | grep tmp
drwxrwxrwt. 21 root root 4096 ноя 13 16:18 tmp
[root@mgurbangeldiev guest]# echo "test" > /tmp/file01.txt
[root@mgurbangeldiev guest]# ls -l /tmp/file01.txt
-rw-r--r--. 1 root root 5 ноя 13 16:27 /tmp/file01.txt
[root@mgurbangeldiev guest]# chmod o+rw /tmp/file01.txt
[root@mgurbangeldiev guest]# ls -l /tmp/file01.txt
-rw-r--rw-. 1 root root 5 ноя 13 16:27 /tmp/file01.txt
```

Figure 2.21: Просмотр атрибутов и разрешения прав

4. От пользователя guest2 (не являющегося владельцем) попробуйте прочесть файл /tmp/file01.txt:

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

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

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

Удалось ли вам выполнить операцию? Да, удалось.

6. Проверьте содержимое файла командой

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

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

```
echo "test3" > /tmp/file01.txt
```

Удалось ли вам выполнить операцию? Да, удалось выполнить операцию.

8. Проверьте содержимое файла командой

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

9. От пользователя guest2 попробуйте удалить файл /tmp/file01.txt командой

```
rm /tmp/file01.txt
```

Удалось ли вам удалить файл? Нет, не удалось удалить файл. (рис. 2.22)

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

Figure 2.22: Просмотр атрибутов и разрешения прав

10. Повысьте свои права до суперпользователя следующей командой

```
su -
```

и выполните после этого команду, снимающую атрибут t (Sticky-бит) с директории /tmp:

```
chmod -t /tmp (рис. 2.23)
```

```
последний вход в систему: пол 10 10:11
[root@mgurbangeldiev ~]# chmod -t /tmp
```

Figure 2.23: Просмотр атрибутов и разрешения прав

11. Покиньте режим суперпользователя командой

`exit`

12. От пользователя `guest2` проверьте, что атрибута `t` у директории `/tmp` нет:

`ls -l / | grep tmp` (рис. 2.24)

```
[guest2@mgurbangeldiev ~]$ ls -l / | grep tmp
drwxrwxrwx. 21 root root 4096 ноя 13 16:33 tmp
[guest2@mgurbangeldiev ~]$ cat /tmp/file01.txt
```

Figure 2.24: Просмотр атрибутов и разрешения прав

13. Повторите предыдущие шаги. Какие наблюдаются изменения?

14. Удалось ли вам удалить файл от имени пользователя, не являющегося его владельцем? Ваши наблюдения занесите в отчёт. (рис. 2.25)

```
[guest2@mgurbangeldiev ~]$ echo "test2" > /tmp/file01.txt
[guest2@mgurbangeldiev ~]$ cat /tmp/file01.txt
test2
[guest2@mgurbangeldiev ~]$ echo "test3" > /tmp/file01.txt
[guest2@mgurbangeldiev ~]$ cat /tmp/file01.txt
test3
[guest2@mgurbangeldiev ~]$ rm /tmp/file01.txt
[guest2@mgurbangeldiev ~]$ cat /tmp/file01.txt
cat: /tmp/file01.txt: Нет такого файла или каталога
```

Figure 2.25: Просмотр атрибутов и разрешения прав

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

15. Повысьте свои права до суперпользователя и верните атрибут `t` на директорию `/tmp`:

`su -`

`chmod +t /tmp`

`exit` (рис. 2.26)

```
[root@mgurbangeldiev ~]# chmod +t /tmp  
[root@mgurbangeldiev ~]# exit  
logout
```

Figure 2.26: Просмотр атрибутов и разрешения прав

3 Выводы

Изучил механизмы изменения идентификаторов, применив SetUID- и Sticky-биты. Получил практические навыки работы в консоли с дополнительными атрибутами. Рассмотрел работы механизмов смены идентификаторов процесса пользователей, а также влияние бита Sticky на запись и удаление файлов.