

Лабораторная работа №5

Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

Кувшинова Ксения Олеговна¹

7.10.2022, Moscow

¹RUDN University, Moscow, Russian Federation

Целью данной лабораторной работы является изучить механизмы 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.

Подготовка лабораторного стенда

Установили компилятор gcc. Отключили систему запретов до очередной перезагрузки системы. После этого команда `getenforce` вывела *Permissive*. (fig. 1)

```
Стем - под именем пользователя root).
[guest@kokuvshinova ~]$ su root
Пароль:
[root@kokuvshinova guest]# yum install gcc
Rocky Linux 9 - BaseOS                    5.2 kB/s | 3.6 kB      00:00
Rocky Linux 9 - BaseOS                    107 kB/s | 1.7 MB      00:16
Rocky Linux 9 - AppStream                 6.3 kB/s | 3.6 kB      00:00
Rocky Linux 9 - AppStream                 3.5 MB/s | 6.0 MB      00:01
Rocky Linux 9 - Extras                    5.4 kB/s | 2.9 kB      00:00
Пакет gcc-11.2.1-9.4.el9.x86_64 уже установлен.
Зависимости разрешены.
Отсутствуют действия для выполнения.
Выполнено!
[root@kokuvshinova guest]# setenforce 0
[root@kokuvshinova guest]# getenforce
Permissive
```

Figure 1: Установка компилятора gcc

Создание программы

Вошли в систему от имени пользователя guest и создали программу simpleid.c. (fig. 2)

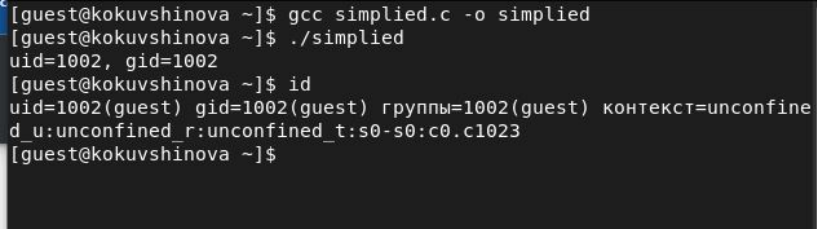


The image shows a code editor window with a title bar containing the filename "simplified.c" and a tilde symbol "~". The window has buttons for "Открыть" (Open), "Сохранить" (Save), and a menu icon. The code is written in C and is as follows:

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

Figure 2: Программа simpleid.c

Скомпилировали программу и убедились, что файл программы создан, выполнили программу `simpleid`, а затем выполнили системную программу `id`. Обе программы выводят одинаковые значения для `uid` и `gid`. (fig. 3)

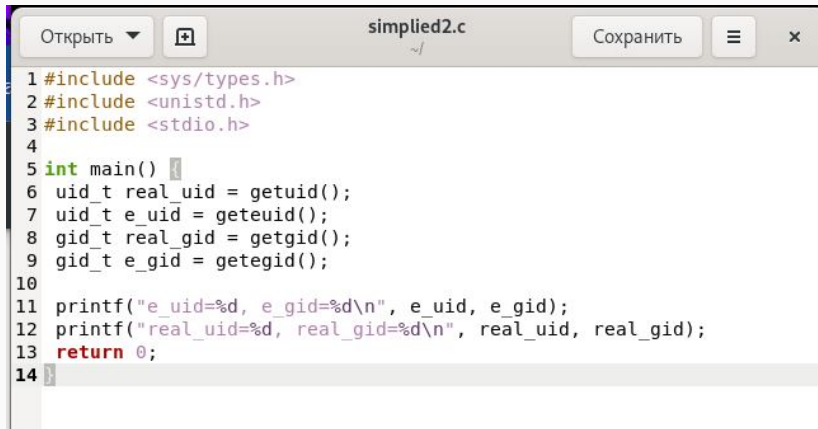


```
[guest@kokuvshinova ~]$ gcc simplified.c -o simplified
[guest@kokuvshinova ~]$ ./simplified
uid=1002, gid=1002
[guest@kokuvshinova ~]$ id
uid=1002(guest) gid=1002(guest) группы=1002(guest) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@kokuvshinova ~]$
```

Figure 3: Выполнение программ `simpleid` и `id`

Создание программы

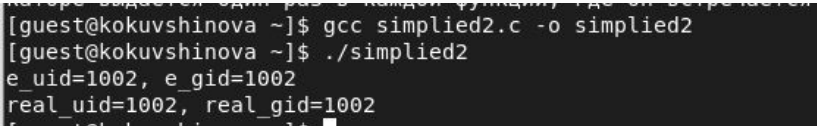
Усложнили программу, добавив вывод действительных идентификаторов и назвали ее `simplified2.c`. (fig. 4)



```
1 #include <sys/types.h>
2 #include <unistd.h>
3 #include <stdio.h>
4
5 int main() {
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
11     printf("e_uid=%d, e_gid=%d\n", e_uid, e_gid);
12     printf("real_uid=%d, real_gid=%d\n", real_uid, real_gid);
13     return 0;
14 }
```

Figure 4: Программа `simpleid2.c`

Скомпилировали и запустили `simplified2.c`. (fig. 5)



```
[guest@kokuvshinova ~]$ gcc simplified2.c -o simplified2
[guest@kokuvshinova ~]$ ./simplified2
e_uid=1002, e_gid=1002
real_uid=1002, real_gid=1002
```

Figure 5: Выполнение программы `simplified2.c`

Повысили временно свои права от имени суперпользователя. Выполнили проверку правильности установки новых атрибутов и смены владельца файла `simplid2`. Запустили `simplid2` и `id`. Значения вывода обеих программ совпадают. (fig. 6)

```
[root@kokuvshinova guest]# chown root:guest /home/guest/simplid2
[root@kokuvshinova guest]# chmod u+s /home/guest/simplid2
[root@kokuvshinova guest]# ls -l simplid2
-rwsrwxr-x. 1 root guest 26008 окт  6 21:56 simplid2
[root@kokuvshinova guest]# ./simplid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@kokuvshinova guest]# id
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Figure 6: Выполнение программ `simplid2` и `id`

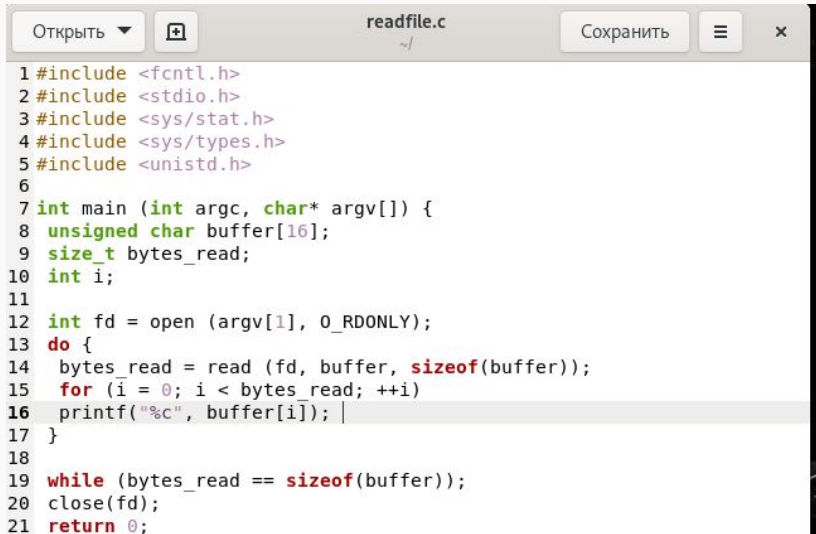
Проделили тоже самое относительно SetGID-бита. Значения вывода обеих программ совпадают, только в отличие от предыдущего пункта значение `e_gid = 1002`. (fig. 7)

```
[root@kokuvshinova guest]# chmod g+s /home/guest/simplified2
[root@kokuvshinova guest]# ls -l simplified2
-rwsrwsr-x. 1 root guest 26008 окт  6 21:56 simplified2
[root@kokuvshinova guest]# ./simplified2
e_uid=0, e_gid=1002
real_uid=0, real_gid=0
[root@kokuvshinova guest]# id
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@kokuvshinova guest]# su guest
```

Figure 7: Выполнение программ `simplified2` и `id` относительно SetGID-бита

Создание программы

Создали программу readfile.c. (fig. 8)



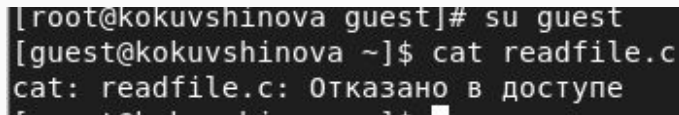
```
1 #include <fcntl.h>
2 #include <stdio.h>
3 #include <sys/stat.h>
4 #include <sys/types.h>
5 #include <unistd.h>
6
7 int main (int argc, char* argv[]) {
8     unsigned char buffer[16];
9     size_t bytes_read;
10    int i;
11
12    int fd = open (argv[1], O_RDONLY);
13    do {
14        bytes_read = read (fd, buffer, sizeof(buffer));
15        for (i = 0; i < bytes_read; ++i)
16            printf("%C", buffer[i]);
17    }
18
19    while (bytes_read == sizeof(buffer));
20    close(fd);
21    return 0;
```

Откомпилировали программу `readfile.c`. Сменили владельца у файла и изменили права так, чтобы только суперпользователь мог прочитать его. (fig. 9)

```
[guest@kokuvshinova ~]$ su root
Пароль:
[root@kokuvshinova guest]# chown root /home/guest/readfile.c
[root@kokuvshinova guest]# chmod 700 /home/guest/readfile.c
```

Figure 9: Смена владельца и изменение прав программы `readfile.c`

Проверили, что пользователь guest не может прочитать файл readfile.c.(fig. 10)

A terminal window with a black background and white text. The first line shows a root user at a machine named kokuvshinova switching to the guest user. The second line shows the guest user attempting to use the 'cat' command to read the file 'readfile.c'. The third line shows the error message 'cat: readfile.c: Отказано в доступе' (cat: readfile.c: Access denied).

```
[root@kokuvshinova guest]# su guest  
[guest@kokuvshinova ~]$ cat readfile.c  
cat: readfile.c: Отказано в доступе
```

Figure 10: Проверка возможности чтения файла readfile.c пользователем guest

Сменили у программы readfile владельца и установили SetU'D-бит.
(fig. 11)

```
[root@kokuvshinova guest]# chown root:guest /home/guest/readfile.c  
[root@kokuvshinova guest]# chmod u+s /home/guest/readfile.c  
[root@kokuvshinova guest]# ./readfile
```

Figure 11: Смена у программы readfile владельца и установка SetU'D-бита

Проверим, может ли программа readfile прочитать файл readfile.c.
(fig. 12)

```
[root@kokuvshinova guest]# ./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;

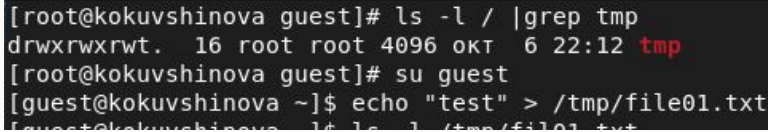
    int fd = open (argv[1], O_RDONLY);
    do {
        bytes_read = read (fd, buffer, sizeof(buffer));
        for (i = 0; i < bytes_read; ++i)
            printf("%c", buffer[i]);
    }

    while (bytes_read == sizeof(buffer));
}
```

Проверим, может ли программа readfile прочитать файл /etc/shadow.
(fig. 13)

```
[root@kokuvshinova guest]# ./readfile /etc/shadow
root:$6$aTmaFYMvaDswKT0J$rIp.yU/HiorM7EBzAmRBTqUFUp59Ticdw7yTkByk8gkj
dbvRp1m0ysvM6ueYernlyIls8BXzxr36GC15GLzEU1::0:99999:7:::
bin:!:19123:0:99999:7:::
daemon:!:19123:0:99999:7:::
adm:!:19123:0:99999:7:::
lp:!:19123:0:99999:7:::
sync:!:19123:0:99999:7:::
shutdown:!:19123:0:99999:7:::
halt:!:19123:0:99999:7:::
mail:!:19123:0:99999:7:::
operator:!:19123:0:99999:7:::
games:!:19123:0:99999:7:::
ftp:!:19123:0:99999:7:::
nobody:!:19123:0:99999:7:::
systemd-coredump:!!:19242:::::
dbus:!!:19242:::::
polkitd:!!:19242:::::
rtkit:!!:19242:::::
sssd:!!:19242:::::
avahi:!!:19242:::::
pipewire:!!:19242:::::
libstoragemgmt:!!:19242:::::
tss:!!:19242:::::
geoclue:!!:19242:::::
cockpit-ws:!!:19242:::::
cockpit-wsinstance:!!:19242:::::
setroubleshoot:!!:19242:::::
```

Выяснили, что установлен атрибут Sticky на директории /tmp. От имени пользователя guest создали файл file01.txt. (fig. 14)



```
[root@kokuvshinova guest]# ls -l / |grep tmp
drwxrwxrwt. 16 root root 4096 окт  6 22:12 tmp
[root@kokuvshinova guest]# su guest
[guest@kokuvshinova ~]$ echo "test" > /tmp/file01.txt
[guest@kokuvshinova ~]$ ls -l /tmp/file01.txt
```

Figure 14: Выполнение команды `ls -l / | grep tmp` и создание файла `file01.txt`

Просмотрели атрибуты у только что созданного файла и разрешили чтение и запись для категории пользователей «все остальные». (fig. 15)

```
[guest@kokuvshinova ~]$ ls -l /tmp/file01.txt
-rw-rw-r--. 1 guest guest 5 окт  6 22:17 /tmp/file01.txt
[guest@kokuvshinova ~]$ chmod o+rw /tmp/file01.txt
[guest@kokuvshinova ~]$ ls -l /tmp/file01.txt
-rw-rw-rw-. 1 guest guest 5 окт  6 22:17 /tmp/file01.txt
[guest@kokuvshinova ~]$
```

Figure 15: Атрибуты файла file01.txt

От пользователя guest2 попробовали прочитать, дозаписать, записать, удалить файл /tmp/file01.txt. Выполнено все, кроме удаления файла.
(fig. 16)

```
[guest@kokuvshinova ~]$ su guest2
Пароль:
[guest2@kokuvshinova guest]$ cat tmp/file01.txt
cat: tmp/file01.txt: Нет такого файла или каталога
[guest2@kokuvshinova guest]$ cat /tmp/file01.txt
test
[guest2@kokuvshinova guest]$ echo "test2" >> /tmp/file01.txt
[guest2@kokuvshinova guest]$ cat /tmp/file01.txt
test
test2
[guest2@kokuvshinova guest]$ echo "test3" > /tmp/file01.txt
[guest2@kokuvshinova guest]$ cat /tmp/file01.txt
test3
[guest2@kokuvshinova guest]$ rm /tmp/file01.txt
rm: невозможно удалить '/tmp/file01.txt': Операция не позволена
[guest2@kokuvshinova guest]$
```

Повысили свои права до суперпользователя и сняли атрибут t (Sticky-бит) с директории /tmp. (fig. 17)

```
[guest2@kokuvshinova guest]$ su -  
Пароль:  
[root@kokuvshinova ~]# chmod -t /tmp  
[root@kokuvshinova ~]# exit  
выход  
[guest2@kokuvshinova guest]$
```

Figure 17: Сняли Sticky-бит с директории /tmp

Повторили предыдущие шаги. В данном случае получилось выполнить удаление файла. (fig. 18)

```
[guest2@kokuvshinova guest]$ cat /tmp/file01.txt
test3
[guest2@kokuvshinova guest]$ echo "test2" >> /tmp/file01.txt
[guest2@kokuvshinova guest]$ cat /tmp/file01.txt
test3
test2
[guest2@kokuvshinova guest]$ echo "test3" > /tmp/file01.txt
[guest2@kokuvshinova guest]$ cat /tmp/file01.txt
test3
[guest2@kokuvshinova guest]$ rm /tmp/file01.txt
[guest2@kokuvshinova guest]$
```

Figure 18: Чтение, дозапись, запись, удаление файл /tmp/file01.txt без атрибута t

Повысили свои права до суперпользователя и вернули атрибут `t` на директорию `/tmp`. (fig. 19)

```
[guest2@kokuvshinova guest]$ su -  
Пароль:  
[root@kokuvshinova ~]# chmod +t /tmp  
[root@kokuvshinova ~]# exit  
выход  
[guest2@kokuvshinova guest]$ ls -l / | grep tmp  
drwxrwxrwt. 18 root root 4096 окт  6 22:23 tmp  
[guest2@kokuvshinova guest]$
```

Figure 19: Чтение, дозапись, запись, удаление файл `/tmp/file01.txt` без атрибута `t`

В ходе выполнения лабораторной работы мы приобрели 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.

1. Кулябов Д. С., Королькова А. В., Геворкян М. Н. Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов [Текст] / Кулябов Д. С., Королькова А. В., Геворкян М. Н. - Москва: - 7 с. [^1]: Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов.
2. Справочник 70 основных команд Linux: полное описание с примерами (<https://eternalhost.net/blog/sozдание-saytov/osnovnyye-komandy-linux>)