

Отчет по лабораторной работе №5

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

Бармина Ольга Константиновна

2022 Sep 21th

Содержание

1. Цель работы	5
2. Выполнение лабораторной работы	6
3. Выводы	12
4. Список литературы	13

Список таблиц

Список иллюстраций

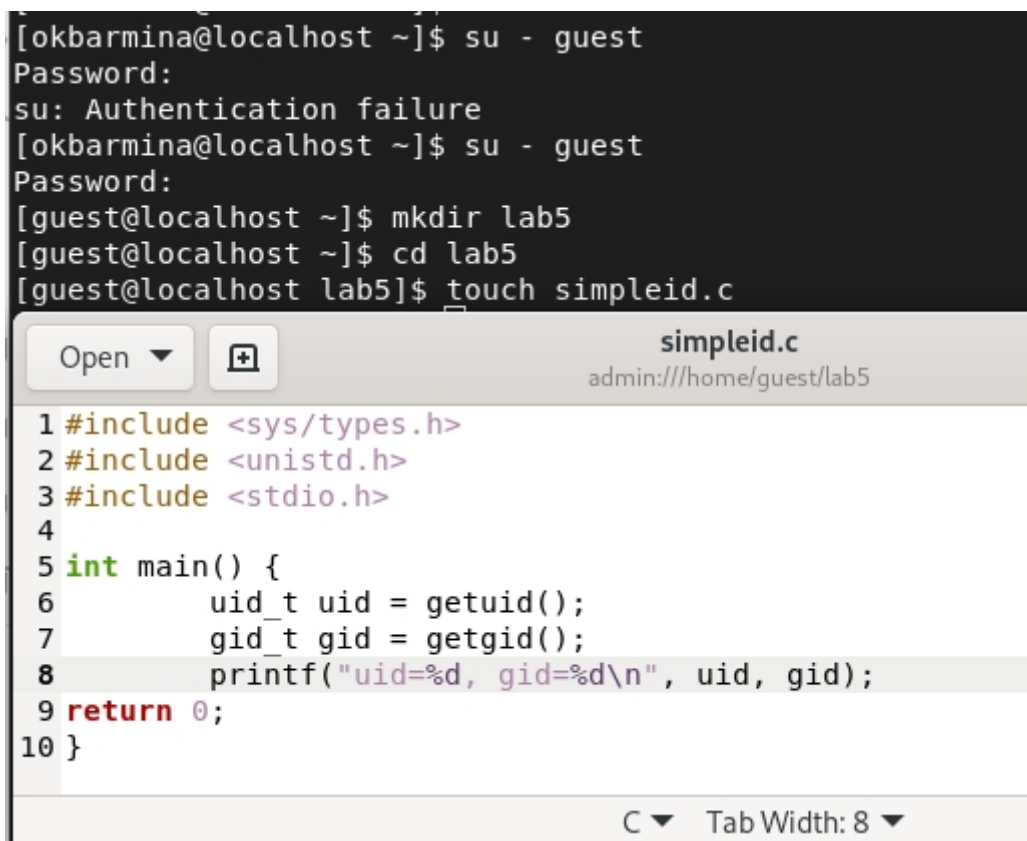
2.1. рис 1. simpleid.c	6
2.2. рис 2. Запуск simpleid.c	7
2.3. рис 3. Дополнение программы	7
2.4. рис 4. Запуск simpleid2.c	7
2.5. рис 5. Команды суперпользователя	8
2.6. рис 6. Запуск readfile.c	9
2.7. рис 7. Запуск readfile.c	9
2.8. рис 8. Проверка readfile.c	10
2.9. рис 9. Тестовый файл	10
2.10. рис 10. Изменение файла другим пользователем	11
2.11. рис 11. Снятие атрибута	11

1. Цель работы

Целью данной работы является 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. [1]

2. Выполнение лабораторной работы

1. Вошла в систему от имени пользователя guest. Создала программу simpleid.c



The image shows a terminal window and a code editor. The terminal window displays the following commands and output:

```
[okbarmina@localhost ~]$ su - guest
Password:
su: Authentication failure
[okbarmina@localhost ~]$ su - guest
Password:
[guest@localhost ~]$ mkdir lab5
[guest@localhost ~]$ cd lab5
[guest@localhost lab5]$ touch simpleid.c
```

The code editor shows the content of the file simpleid.c:

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

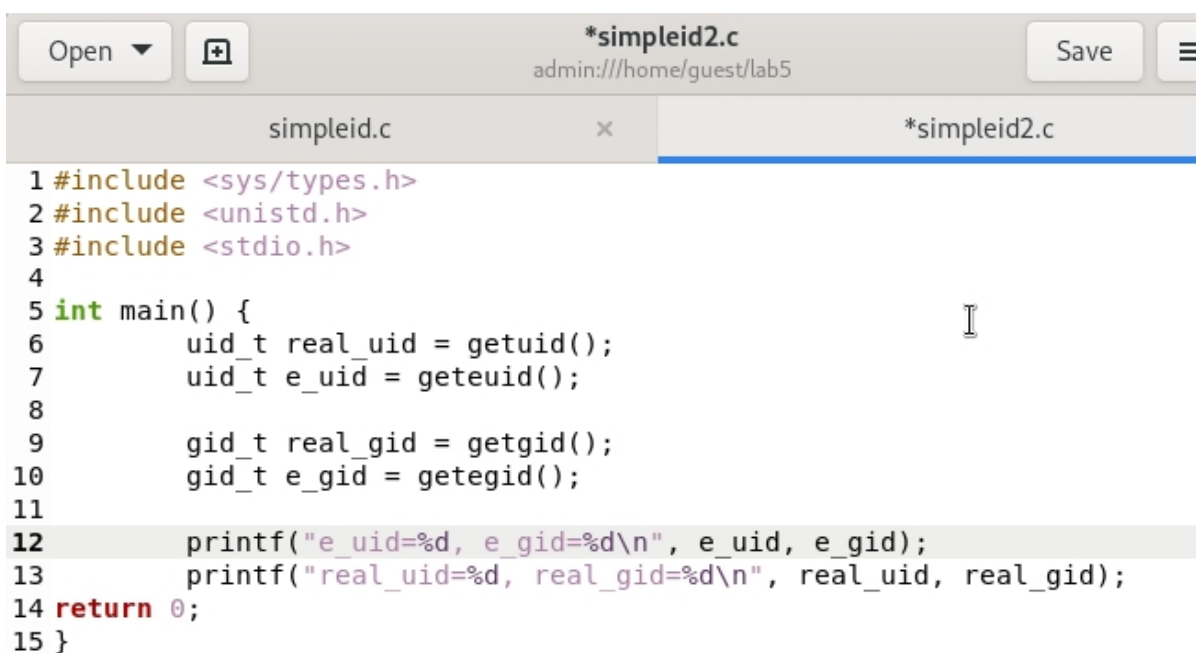
Рис. 2.1.: рис 1. simpleid.c

2. Скомпилировала программу и убедилась, что файл программы создан. Выполнила программу. Для сравнения выполнила системную программу id.

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

Рис. 2.2.: рис 2. Запуск simpleid.c

3. Усложнила программу, добавив вывод действительных идентификаторов.



```
*simpleid2.c
admin:///home/guest/lab5
Save

simpleid.c x *simpleid2.c

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
9     gid_t real_gid = getgid();
10    gid_t e_gid = getegid();
11
12    printf("e_uid=%d, e_gid=%d\n", e_uid, e_gid);
13    printf("real_uid=%d, real_gid=%d\n", real_uid, real_gid);
14    return 0;
15 }
```

Рис. 2.3.: рис 3. Дополнение программы

4. Скомпилировала и запустила simpleid2.c

```
[guest@localhost lab5]$ gcc simpleid2.c -o simpleid2
[guest@localhost lab5]$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
[guest@localhost lab5]$
```

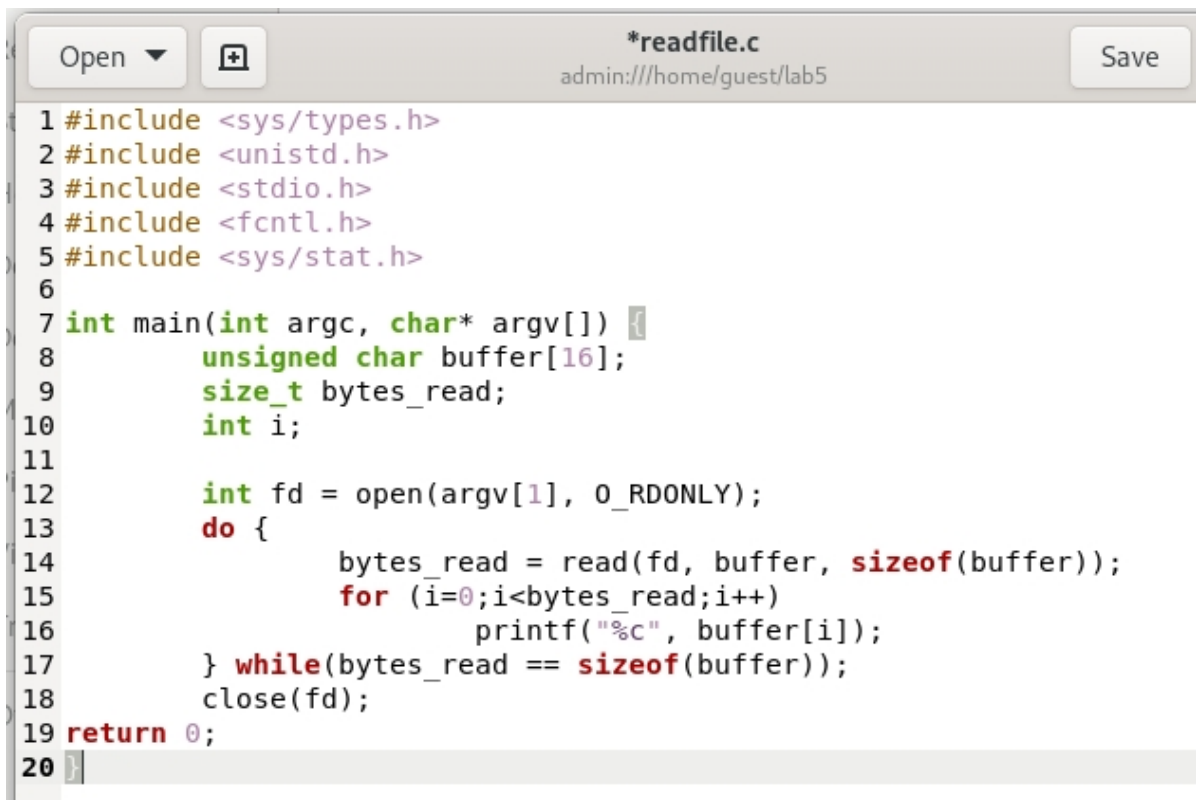
Рис. 2.4.: рис 4. Запуск simpleid2.c

5. От имени суперпользователя выполните команды `chown root:guest /home/guest/simpleid2` и `chmod u+s /home/guest/simpleid2`. Выполните проверку правильности установки новых атрибутов и смены владельца файла `simpleid2`. Запустила `simpleid2` и `id`.

```
[guest@localhost lab5]$ su
Password:
[root@localhost lab5]# chown root:guest /home/guest/lab5/simpleid2
[root@localhost lab5]# chmod u+s /home/guest/lab5/simpleid2
[root@localhost lab5]# exit
exit
[guest@localhost lab5]$ ls -l simpleid2
-rwsr-xr-x. 1 root guest 26008 Sep 21 12:32 simpleid2
[guest@localhost lab5]$ ./simpleid2
e_uid=0, e_gid=1001
real_uid=1001, real_gid=1001
[guest@localhost lab5]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Рис. 2.5.: рис 5. Команды суперпользователя

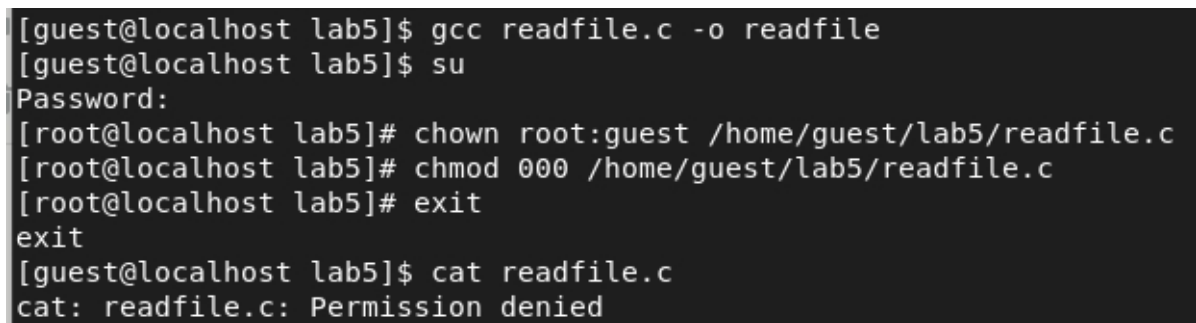
6. Создала программу `readfile.c`



```
1 #include <sys/types.h>
2 #include <unistd.h>
3 #include <stdio.h>
4 #include <fcntl.h>
5 #include <sys/stat.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    } while(bytes_read == sizeof(buffer));
18    close(fd);
19    return 0;
20 }
```

Рис. 2.6.: рис 6. Запуск readfile.c

7. Откомпилировала её. Сменила владельца у файла readfile.c и изменила права так, чтобы только суперпользователь мог прочитать его, а guest не мог.



```
[guest@localhost lab5]$ gcc readfile.c -o readfile
[guest@localhost lab5]$ su
Password:
[root@localhost lab5]# chown root:guest /home/guest/lab5/readfile.c
[root@localhost lab5]# chmod 000 /home/guest/lab5/readfile.c
[root@localhost lab5]# exit
exit
[guest@localhost lab5]$ cat readfile.c
cat: readfile.c: Permission denied
```

Рис. 2.7.: рис 7. Запуск readfile.c

8. Сменила у программы readfile владельца и установила SetU'D-бит. Проверила, может ли программа readfile прочитать файл readfile.c, Проверила,

может ли программа readfile прочитать файл /etc/shadow.

```
[guest@localhost lab5]$ ./readfile /etc/shadow
@@00# 000000000000Pn0# @v@0HC0\00r00)ZHC0\0V@000# >@060ivie00;0HC0\
0000D000n0# `C0\>@HC0\0`C0\0 0# p@C0\00@8C0\0G0\0)G0\05G0\0EG0\0\G0\0jG0\
0}G0\00G0\00G0\00G0\00G0\00G0\00M0\00M0\0N0\0N0\0(N0\00N0\0IN0\00N0\000\0:00\000
00#000\0p@0\030000d@08
00iE0\00000\0yE0\00000D00;0000sVx86_64./readfile/etc/shadowSHELL=/bin/bashHISTCO
NTROL=ignoredupsHISTSIZE=1000HOSTNAME=localhostPWD=/home/guest/lab5LOGNAME=guest
XAUTHORITY=/home/guest/.xauthBmfb10HOME=/home/guestLANG=en_US.UTF-8LS_COLORS=rs=
0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33;01:cd=40;33;01:or=
40;31;01:mi=01;37;41:su=37;41:sg=30;43:ca=30;41:tw=30;42:ow=34;42:st=37;44:ex=01
;32:*.tar=01;31:*.tgz=01;31:*.arc=01;31:*.arj=01;31:*.taz=01;31:*.lha=01;31:*.lz
4=01;31:*.lzh=01;31:*.lzma=01;31:*.tlz=01;31:*.txz=01;31:*.tzo=01;31:*.t7z=01;31
:*.zip=01;31:*.z=01;31:*.dz=01;31:*.gz=01;31:*.lrz=01;31:*.lz=01;31:*.lzo=01;31:
```

Рис. 2.8.: рис 8. Проверка readfile.c

9. Выяснила, установлен ли атрибут Sticky на директории /tmp. От имени пользователя guest создала файл file01.txt в директории /tmp со словом test. Просмотрела атрибуты у только что созданного файла и разрешила чтение и запись для категории пользователей «все остальные».

```
[guest@localhost ~]$ ls -l / | grep tmp
drwxrwxrwt. 18 root root 4096 Sep 21 12:45 tmp
[guest@localhost ~]$ echo "test" > /tmp/file01.txt
[guest@localhost ~]$ ls -l /tmp/file01.txt
-rw-r--r--. 1 guest guest 5 Sep 21 12:48 /tmp/file01.txt
[guest@localhost ~]$ chmod o+rw /tmp/file01.txt
[guest@localhost ~]$ ls -l /tmp/file01.txt
-rw-r--rw-. 1 guest guest 5 Sep 21 12:48 /tmp/file01.txt
```

Рис. 2.9.: рис 9. Тестовый файл

10. От пользователя guest2 попробовала прочитать файл, дозаписать в файл слово test2. Проверила содержимое файла. Попробовала записать в файл слово test3, стерев при этом всю имеющуюся в файле информацию. Проверила содержимое файла. Попробовала удалить файл.

```
[guest@localhost ~]$ su guest2
Password:
[guest2@localhost guest]$ cat /tmp/file01.txt
test
[guest2@localhost guest]$ echo "test2" > /tmp/file01.txt
bash: /tmp/file01.txt: Permission denied
[guest2@localhost guest]$ cat /tmp/file01.txt
test
[guest2@localhost guest]$ echo "test3" > /tmp/file01.txt
bash: /tmp/file01.txt: Permission denied
[guest2@localhost guest]$ rm /tmp/file01.txt
rm: remove write-protected regular file '/tmp/file01.txt'? y
rm: cannot remove '/tmp/file01.txt': Operation not permitted
```

Рис. 2.10.: рис 10. Изменение файла другим пользователем

11. Повысила свои права до суперпользователя и выполнила команду, снимающую атрибут t с директории /tmp. Покинула режим суперпользователя. От пользователя guest2 проверила, что атрибута t у директории /tmp нет. Попробовала прочитать файл, дозаписать в файл, удалить его. Повысила свои права до суперпользователя и вернула атрибут t на директорию /tmp.

```
[guest2@localhost guest]$ su -
Password:
[root@localhost ~]# chmod -t /tmp
[root@localhost ~]# exit
logout
[guest2@localhost guest]$ ls -l / | grep tmp
drwxrwxrwx. 19 root root 4096 Sep 21 12:55 tmp
[guest2@localhost guest]$ echo "test2" > /tmp/file01.txt
bash: /tmp/file01.txt: Permission denied
[guest2@localhost guest]$ cat /tmp/file01.txt
test
[guest2@localhost guest]$ rm /tmp/file01.txt
rm: remove write-protected regular file '/tmp/file01.txt'? y
[guest2@localhost guest]$ su -
Password:
[root@localhost ~]# chmod +t /tmp
[root@localhost ~]# exit
logout
```

Рис. 2.11.: рис 11. Снятие атрибута

3. Выводы

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

4. Список литературы

1. Методические материалы курса