

# Лабораторная работа #5

Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов.

Критский Сергей Дмитриевич

# Содержание

|                                    |    |
|------------------------------------|----|
| Цель работы                        | 3  |
| Выполнение лабораторной работы     | 4  |
| Создание программ . . . . .        | 4  |
| Исследование Sticky-бита . . . . . | 9  |
| Выводы                             | 11 |


## 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.

# Выполнение лабораторной работы

## Создание программ

Написал программу simpleid.c .



```
1 #include <sys/types.h>
2 #include <unistd.h>
3 #include <stdio.h>
4
5 int 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 }
```

Рис. 1: Код программы

Компиляция и выполнение программы.

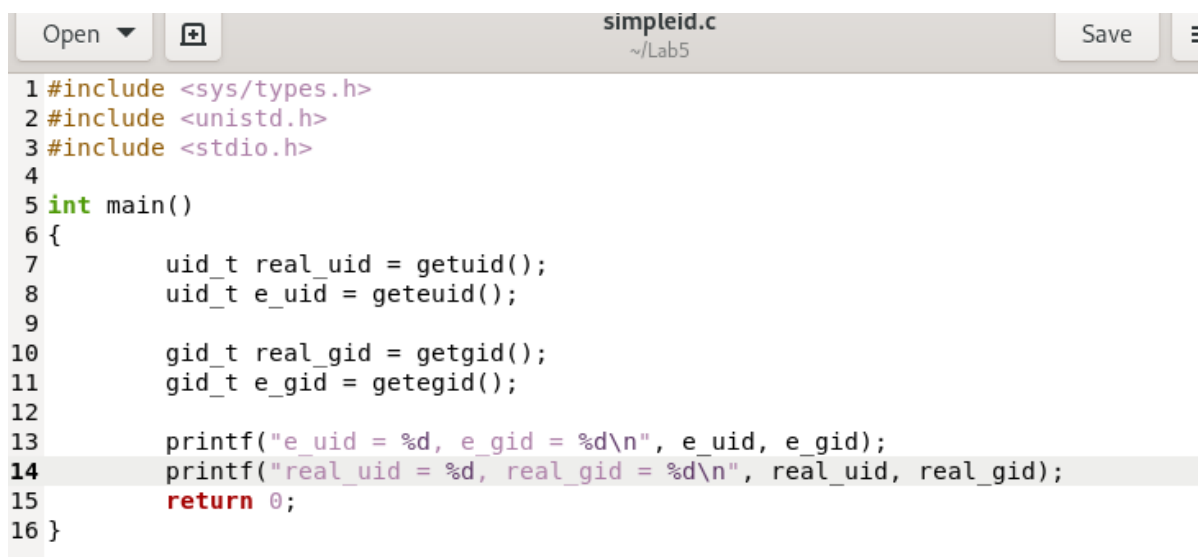
```

function it appears in
guest@SDKritskiy Lab5]$ gcc simpleid.c -o simpleid
guest@SDKritskiy Lab5]$ ./simpleid
id = 1001, gid = 1001
guest@SDKritskiy Lab5]$ id
id=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
guest@SDKritskiy Lab5]$

```

Рис. 2: Результат выполнения

Усложнил первую программу с выводом действительный идентификаторов.



```

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

```

Рис. 3: Код программы

Компиляция и выполнение программы от guest и superuser.

```
[guest@SDKritskiy Lab5]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unco
ed_r:unconfined_t:s0-s0:c0.c1023
[guest@SDKritskiy Lab5]$ gcc simpleid.c -o simpleid2
[guest@SDKritskiy Lab5]$ ./simpleid2
e_uid = 1001, e_gid = 1001
real_uid = 1001, real_gid = 1001
[guest@SDKritskiy Lab5]$
```

Рис. 4: Пользователь guest

```
[root@SDKritskiy Lab5]# ./simpleid2
e_uid = 0, e_gid = 0
real_uid = 0, real_gid = 0
[root@SDKritskiy Lab5]# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unc
ned_t:s0-s0:c0.c1023
[root@SDKritskiy Lab5]#
```

Рис. 5: Суперпользователь

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 main(int argc, char* argv[])
8 {
9     unsigned char buffer[16];
10    size_t bytes_read;
11    int i;
12
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
20    while(bytes_read == sizeof(buffer));
21    close(fd);
22    return 0;
23 }
```

Рис. 6: Код программы

Компиляция и выполнение программы.

```
[root@SDKritskiy Lab5]# ./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;
}
```

Рис. 7: Результат выполнения на примере readfile.c



```
guest@SDKritskiy:/home/guest/Lab5
[root@SDKritskiy Lab5]# ./readfile etc/shadow
@n0000000000000000P.r00@@V@0x60000F0500/x60000V@0000>@0F0y0$c00F00vo0x60000o00G000
.r000600>@x600006000 00p@p60000@h6000AT000LT000WT000gT0000T0000T0000T000U00
0%U000NU000bU000xU0000U0000U0000U0000U0000U0000V000V000"V0001V000BV0007\000Q\000b
\000|\0000\0000\000\000]0007]000B]000_ ]000j]000r]0000]0000]0000]000_ ^0000^0000^0
P00000:_p@0M_0000_000!000030000d@q8

I900000_000Y90000o00G000A0u#x86_64./readfileetc/shadowSHELL=/bin/bashSESSION_MAN
AGER=local/unix:@/tmp/.ICE-unix/3107,unix/unix:/tmp/.ICE-unix/3107COLORTERM=true
colorHISTCONTROL=ignoredupsXDG_MENU_PREFIX=gnome-HOSTNAME=SDKritskiy.localdomain
HISTSIZE=100SSH_AUTH_SOCK=/run/user/1001/keyring/sshXMODIFIERS=@im=ibusDESKTOP_
SESSION=gnomePWD=/home/guest/Lab5XDG_SESSION_DESKTOP=gnomeLOGNAME=guestXDG_SESSI
ON_TYPE=waylandSYSTEMD_EXEC_PID=3129XAUTHORITY=/root/.xauthD48vFKGDM_LANG=en_US.
UTF-8HOME=/rootUSERNAME=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:s
u=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:*.lz4=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:*.xz=01;31:*.zst=01;31
:*.tzst=01;31:*.bz2=01;31:*.bz=01;31:*.tbz=01;31:*.tbz2=01;31:*.tz=01;31:*.deb=0
1;31:*.rpm=01;31:*.jar=01;31:*.war=01;31:*.ear=01;31:*.sar=01;31:*.rar=01;31:*.a
lz=01;31:*.ace=01;31:*.zoo=01;31:*.cpio=01;31:*.7z=01;31:*.rz=01;31:*.cab=01;31:
*.wim=01;31:*.swm=01;31:*.dwm=01;31:*.esd=01;31:*.jpg=01;35:*.jpeg=01;35:*.mjpg=
01;35:*.mjpeg=01;35:*.gif=01;35:*.bmp=01;35:*.pbm=01;35:*.pgm=01;35:*.ppm=01;35:
```

Рис. 8: Неудачный результат выполнения на примере etc/shadow

## Исследование Sticky-бита

Создал файл от имени guest в директории со sticky-битом. Заменял содержимое и прочитал файл от имени пользователя guest2. Попытался удалить файл.

```

[root@SDKritskiy Lab5]# su guest2
[guest2@SDKritskiy Lab5]$ cat /tmp/file01.txt
test
[guest2@SDKritskiy Lab5]$ echo "test2" > /tmp/file01.txt
[guest2@SDKritskiy Lab5]$ cat /tmp/file01.txt
test2
[guest2@SDKritskiy Lab5]$ echo "test3" > /tmp/file01.txt
[guest2@SDKritskiy Lab5]$ cat /tmp/file01.txt
test3
[guest2@SDKritskiy Lab5]$ rm /tmp/file01.txt
rm: cannot remove '/tmp/file01.txt': Operation not permitted

```

Рис. 9: Взаимодействие с файлом в директории со Sticky-битом

Убрал sticky-бит с директории, повторил действия.

```

[guest2@SDKritskiy Lab5]$ su
Password:
[root@SDKritskiy Lab5]# chmod -t /tmp
[root@SDKritskiy Lab5]# exit
exit
[guest2@SDKritskiy Lab5]$ ls -l / | grep tmp
drwxrwxrwx. 16 root root 4096 Oct  8 11:43 tmp
[guest2@SDKritskiy Lab5]$ echo "test3" > /tmp/file01.txt
[guest2@SDKritskiy Lab5]$ echo "test4" > /tmp/file01.txt
[guest2@SDKritskiy Lab5]$ cat /tmp/file01.txt
test4
[guest2@SDKritskiy Lab5]$ rm /tmp/file01.txt
[guest2@SDKritskiy Lab5]$ su
Password:
[root@SDKritskiy Lab5]# chmod +t /tmp
[root@SDKritskiy Lab5]#

```

Рис. 10: Взаимодействие с файлом в директории без Sticky-бита

## Выводы

Я изучил механизмы изменения идентификаторов, применения SetUID- и Sticky-битов. Получил практические навыки работы в консоли с дополнительными атрибутами. Рассмотрел работу механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.