

Отчет по лабораторной работе №5

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

Александр Олегович Воробьев

Содержание

| | | |
|----------|---|-----------|
| 1 | Цель работы | 3 |
| 2 | Последовательность выполнения работы | 4 |
| 2.1 | Подготовка лабораторного стенда | 4 |
| 2.2 | Создание программы | 4 |
| 2.3 | Исследование Sticky-бита | 9 |
| 3 | Выводы | 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 Последовательность выполнения работы

2.1 Подготовка лабораторного стенда

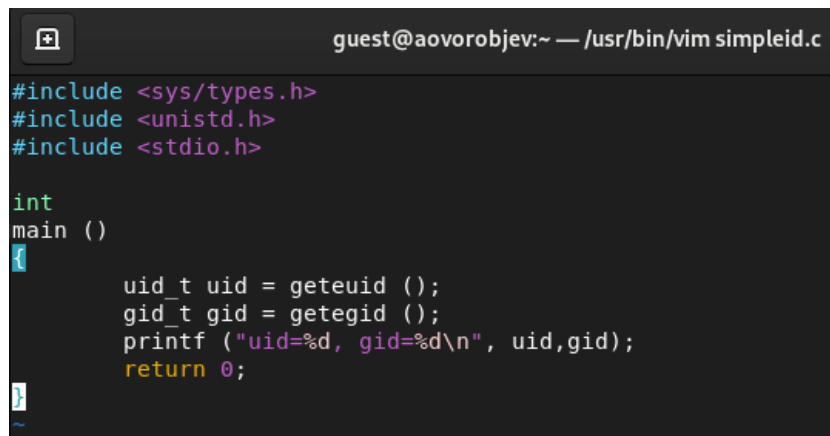
1. Установка gcc

```
[alexander@aovorobjev ~]$ gcc -v
Используются внутренние спецификации.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/usr/libexec/gcc/aarch64-redhat-linux/11/lto-wrapper
Целевая архитектура: aarch64-redhat-linux
Параметры конфигурации: ../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/ --enable-shared --enable-threads=posix --enable-checking=release --enable-multi-lib --with-system-zlib --enable-_cxa_atexit --disable-libunwind-exceptions --enable-gnu-unique-object --enable-linker-build-id --with-gcc-major-version-only --with-linker-hash-style=gnu --enable-plugin --enable-initfini-array --without-isl --enable-gnu-indirect-function --build=aarch64-redhat-linux --with-build-config=bootstrap-lto --enable-link-serialization=1
Модель многопоточности: posix
Supported LTO compression algorithms: zlib zstd
gcc версия 11.2.1 20220127 (Red Hat 11.2.1-9) (GCC)
```

Figure 2.1: Установка gcc

2.2 Создание программы

1. Войдите в систему от имени пользователя guest.
2. Создайте программу simpleid.c:



```
guest@aovorobjev:~ — /usr/bin/vim simpleid.c
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>

int
main ()
{
    uid_t uid = geteuid ();
    gid_t gid = getegid ();
    printf ("uid=%d, gid=%d\n", uid,gid);
    return 0;
}
```

Figure 2.2: Создание программы

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`

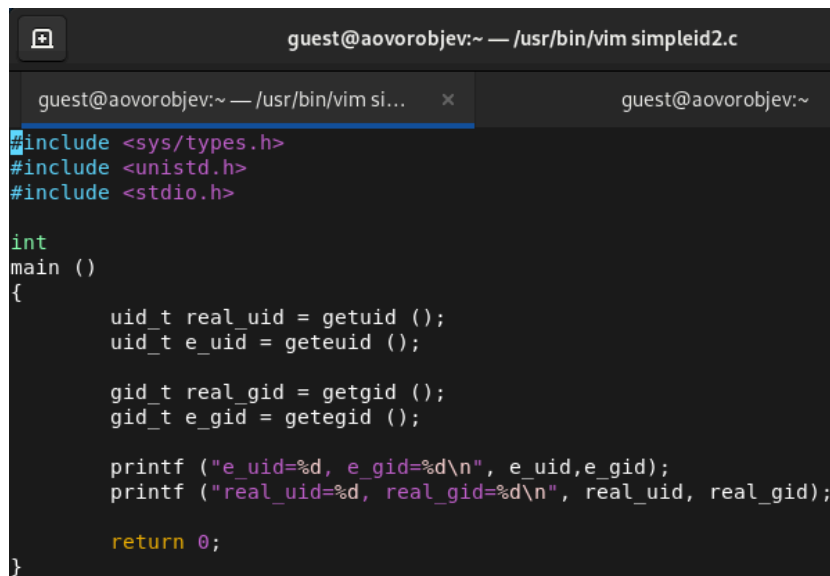
и сравните полученный вами результат с данными предыдущего пункта задания.



```
[guest@aovorobjev ~]$ vi simpleid.c
[guest@aovorobjev ~]$ vi simpleid.c
[guest@aovorobjev ~]$ gcc simpleid.c -o simpleid
[guest@aovorobjev ~]$ ./simpleid
uid=1002, gid=1002
[guest@aovorobjev ~]$ id
uid=1002(guest) gid=1002(guest) rгруппы=1002(guest) контекст=unconfined_u:unconfi
ned_r:unconfined_t:s0-s0:c0.c1023
[guest@aovorobjev ~]$
```

Figure 2.3: Компиляция программы

6. Усложните программу, добавив вывод действительных идентификаторов:



```
guest@aovorobjev:~ — /usr/bin/vim simpleid2.c
guest@aovorobjev:~ — /usr/bin/vim si... x guest@aovorobjev:~
#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 2.4: Усложненная программа

7. Скомпилируйте и запустите simpleid2.c:
gcc simpleid2.c -o simpleid2
./simpleid2
8. От имени суперпользователя выполните команды:
chown root:guest /home/guest/simpleid2
chmod u+s /home/guest/simpleid2
9. Используйте sudo или повысьте временно свои права с помощью su.
Поясните, что делают эти команды.
10. Выполните проверку правильности установки атрибутов расширенных прав владельца файла simpleid2:
ls -l simpleid2
11. Запустите simpleid2 и id:
./simpleid2

id

Сравните результаты.

12. Прodelайте тоже самое относительно SetGID-бита.

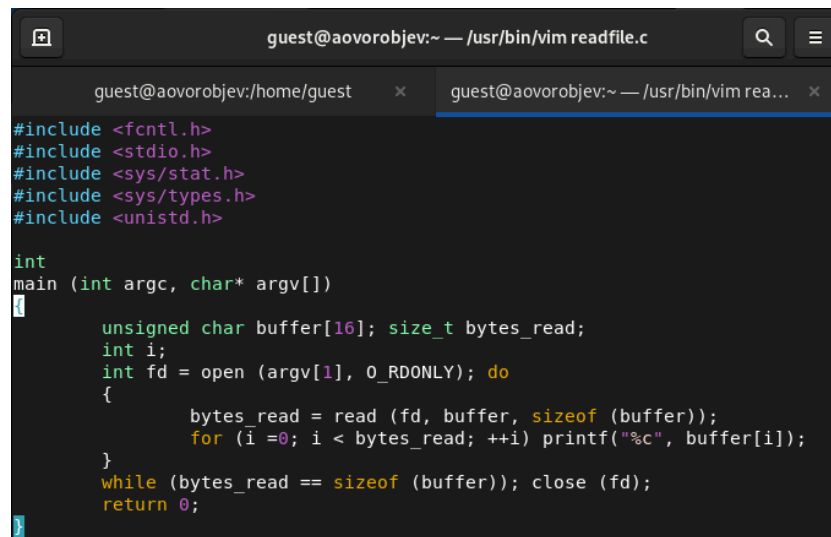
```
[guest@aovorobjev ~]$ vi simpleid2.c
[guest@aovorobjev ~]$ gcc simpleid2.c -o simpleid2
[guest@aovorobjev ~]$ ./simpleid2
e_uid=1002, e_gid=1002
real_uid=1002, real_gid=1002
[guest@aovorobjev ~]$ su
Пароль:
[root@aovorobjev guest]# chown root:guest /home/guest/simpleid2
[root@aovorobjev guest]# chmod u+s /home/guest/simpleid2
[root@aovorobjev guest]# ls -l simpleid2
-rwsrwxr-x. 1 root guest 80584 окт  5 09:55 simpleid2
[root@aovorobjev guest]# ./simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@aovorobjev guest]# id
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Figure 2.5: Компиляция усложненной программы, сравнение результатов

13. Создайте программу readfile.c:

14. Откомпилируйте её.

gcc readfile.c -o readfile



```
guest@aovorobjev:~ — /usr/bin/vim readfile.c
guest@aovorobjev:/home/guest x guest@aovorobjev:~ — /usr/bin/vim rea... x
#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 2.6: Создание новой программы

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.

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

Figure 2.7: Смена владельца у файла

16. Проверьте, что пользователь `guest` не может прочитать файл `readfile.c`.

```
[guest@aovorobjev ~]$ cat readfile.c
cat: readfile.c: Отказано в доступе
```

Figure 2.8: Проверка условия

17. Смените у программы readfile владельца и установите SetU'D-бит.

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

Figure 2.9: смена владельца у файла

18. Проверьте, может ли программа `readfile` прочитать файл `readfile.c`?

```
[root@aovorobjev guest]# ./readfile readfile.c
0A000 00000000000000000000r00000000tqpXr0000Xf00000At00000r00000A0A00000k0j00 0k
+0j000tGr0000H0p@@@Bpb000m00000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000
M000000000!P0003p000/d@@8 00000 @@

Hr0000000000Xr0000000/00.r0jaarch64./readfileareadfile.cSHELL=/bin/bashSESSION
_MANAGER=local/unix:@tmp/.ICE-unix/32886,unix/unix:/tmp/.ICE-unix/32886COLORR
ERM=truecolorHISTCONTROL=ignorespaceXDG_MENU_PREFIX=gnome-HOSTNAME=aovorobjevHI
STSIZE=1000SSH_AUTH_SOCK=/run/user/1002/keyring/sshXMODIFIERS=@im=ibusDESKTOP
_SESSION=gnomePWD=/home/guestXDG_SESSION_DESKTOP=gnomeLOGNAME=guestXDG_SESSIO
N_TYPE=waylandSYSTEMD_EXEC_PID=32910XAUTHORITY=/root/.xauthj9c4n3GDMLANG=ru RU.
UTF-8HOME=/rootUSERNAME=guestLANG=ru RU.UTF-8LS_COLORS=rs=0:di=01;34:ln=01;36:
mh=00;pi=40;33:s0=01;35;d0=1;35;b0=40;33;0f=40;33;0l=01;31;0l:m=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:*.lz4=01;31:*.lzh=
01;31:*.lzm=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:*.tzt=01;31:*.bz2=01;31:*.bz=01;31:*.tbz=01;31:*.tbz2=01;31:*.tz
=01;31:*.deb=01;31:*.rpm=01;31:*.jar=01;31:*.war=01;31:*.ear=01;31:*.sar=01;
```

Figure 2.10: Проверка

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

```
[root@aovorobjev guest]# ./readfile /etc/shadow
00000 0000000000000000000000000000000000H0000tGp000000H00000At@0000`000000A00000000N00D00
00D00Dt@00000Gd0wP@@cBP@cAs0000xS0000X0000S0000;0000-0000-0000_0000_0000%0<%000e%000y%0000%0000%0000%0000%0000%0000%0000S0000*S00005$0000D&0000
US0000J,0000d,0000u,0000o,0000O,00000,00000,00000,0000J-0000U-0000r-0000)-0000
-00000-00000-0000r.00000..00000..00002/0000M/0000!P0003p000/d@e8
000 @Q

0000000/0000(00000Gd0Wdq0+00Daarch64./readfile/etc/shadowSHELL=/bin/bashSESS
ON_MANAGER=local/unix:@tmp/.ICE-unix/32886,unix:/usr/libexec/tmp/:@tmp/.ICE-unix/32886COL
RTERM=truecolorhistCONTROL_IGNOREDUPS=XDG_MENU_PREFIX=gnome-HOSTNAME=aovorobjEV
HISTSIZE=1000SSH_AUTH_SOCK=/run/user/1002/keyring/sshXMODIFIERS=@im=ibusDESKTOP
SESSION=gnomePWD=/home/guestXDG_SESSION_DESKTOP=gnomeLOGNAME=guestXDG_SESSI
_N_TYPE=waylandSYSTEMD_EXEC_PID=32910XAUTHORITY=/root/.xauthja9c4n3GDMLANG=rU
UT.UFT-8HOME=/rootUSERNAME=guestLANG=en_RU.UTF-LS_COLORS=rs=0:di=1;l=1;n=1;
6:mh=0:p=i=40;3:s=o=1;35;d=0=l=35;b=d=40;33;o=c=d=40;33;i:r=o=40;31;o:i=m=1;
7;4l=su=37;4lsg=30;43ca=30;4ltw=30;42ow=34;4dst=37;44ex=01;32*:tar=01;
1*:tzg=01;31*:arc=01;31*:arj=01;31*:taz=01;31*:lha=01;31*:lz4=01;31*:l
h=01;31*.lzm=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*.ztst=01;31*.bz2=01;31*.bz=01;31*.tbz=01;31*.tbz2=01;31*
tar.gz=01;31*/chk=01;31*/rpm=01;31/*.*.tgz=01;31/*.*.rar=01;31/*.*.zip=0
```

Figure 2.11: Проверка

2.3 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
```

```
[guest@aovorobjev ~]$ ls -l / | grep tmp
drwxrwxrwt. 19 root root 4096 окт  5 10:11 tmp
[guest@aovorobjev ~]$ echo "test" > /tmp/file01.txt
[guest@aovorobjev ~]$ ls -l /tmp/file01.txt
-rw-rw-r--. 1 guest guest 5 окт  5 10:12 /tmp/file01.txt
[guest@aovorobjev ~]$ chmod o+rw /tmp/file01.txt
[guest@aovorobjev ~]$ ls -l /tmp/file01.txt
-rw-rw-rw-. 1 guest guest 5 окт  5 10:12 /tmp/file01.txt
[guest@aovorobjev ~]$
```

Figure 2.12: Проверка атрибута

4. От пользователя guest2 (не являющегося владельцем) попробуйте прочитать 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?

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?

8. Проверьте содержимое файла командой

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

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

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

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

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`

11. Покиньте режим суперпользователя командой

exit

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

`-l / | grep tmp`

```
[guest2@aovorobjev ~]$ ls -l / | grep tmp
drwxrwxrwx. 19 root root 4096 окт  5 10:15 tmp
```

Figure 2.13: Повышение прав, проверка отсутствия атрибута

13. Повторите предыдущие шаги. Какие наблюдаются изменения?

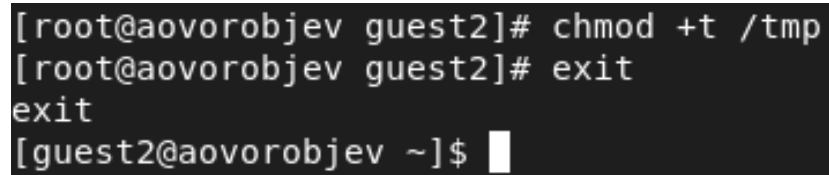
14. Удалось ли вам удалить файл от имени пользователя, не являющегося его владельцем? Ваши наблюдения занесите в отчёт.

```
[guest2@aovorobjev ~]$ echo "test4" > /tmp/file01.txt
[guest2@aovorobjev ~]$ rm /tmp/file01.txt
[guest2@aovorobjev ~]$
```

Figure 2.14: Повторение предыдущих шагов

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

```
su -  
chmod +t /tmp  
exit
```



```
[root@aovorobjev guest2]# chmod +t /tmp  
[root@aovorobjev guest2]# exit  
exit  
[guest2@aovorobjev ~]$
```

Figure 2.15: Возвращение атрибута

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.