

Информационная безопасность. Лабораторная работа № 5 на тему “Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов”

Горбунова Ярослава Михайловна

RUDN University, Moscow, Russian Federation

Содержание

- Цели и задачи
- Выполнение
- Результаты
- Список литературы

Цели и задачи

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

```
[ymgorbunova@ymgorbunova ~]$ gcc -v
Используются внутренние спецификации.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/usr/libexec/gcc/x86_64-redhat-linux/11/lto-wrapper
OFFLOAD_TARGET_NAMES=nvptx-none
OFFLOAD_TARGET_DEFAULT=1
Целевая архитектура: x86_64-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-bug-url=https://bugs.rockylinux.org/ --enable-shared --enable-threads=posix --enable-checking=release --enable-multilib --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-offload-targets=nvptx-none --without-cuda-driver --enable-gnu-indirect-function --enable-cet --with-tune=generic --with-arch_64=x86-64-v2 --with-arch_32=x86-64 --build=x86_64-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 1: В системе установлен компилятор gcc

```
ymgorbunova@ymgorbunova ~]$ sudo setenforce 0
```

Мы полагаем, что ваш системный администратор изложил вам основы безопасности. Как правило, всё сводится к трём следующим правилам:

- №1) Уважайте частную жизнь других.
- №2) Думайте, прежде что-то ввести.
- №3) С большой властью приходит большая ответственность.

```
[sudo] пароль для ymgorbunova:
```

```
ymgorbunova@ymgorbunova ~]$ getenforce
```

```
Permissive
```

```
ymgorbunova@ymgorbunova ~]$
```

Figure 2: Отключение системы запретов до очередной перезагрузки системы

```
[guest@ymgorbunova dir1]$ touch simpleid.c
[guest@ymgorbunova dir1]$ gcc simpleid.c -o simpleid
[guest@ymgorbunova dir1]$ ls -l
итого 32
-rwxrwxr-x. 1 guest guest 25904 сен 28 14:49 simpleid
-rw-rw-r--. 1 guest guest 175 сен 28 14:49 simpleid.c
[guest@ymgorbunova dir1]$ ./simpleid
uid=1001, gid=1001
[guest@ymgorbunova dir1]$ id
uid=1001(guest), gid=1001(guest), группы=1001(guest), контексты=unconfined,unconfined,unconfined, t=60, c0
```

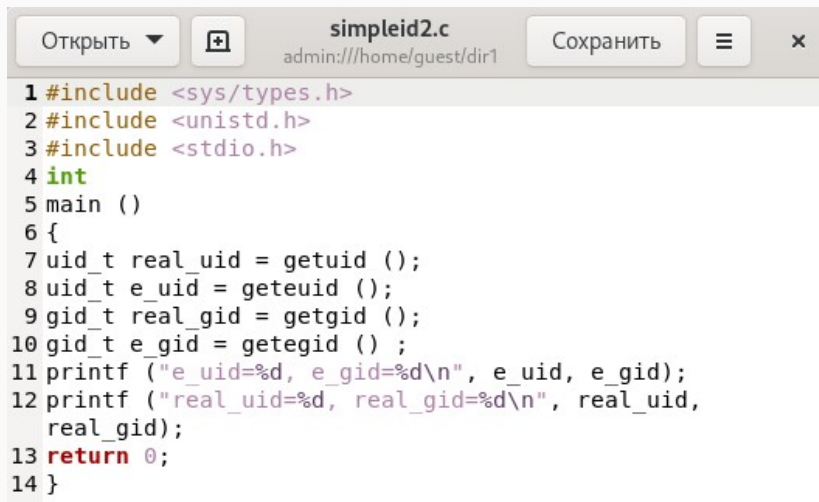



```
1 #include <sys/types.h>
2 #include <unistd.h>
3 #include <stdio.h>
4 int
5 main ()
6 {
7     uid_t uid = geteuid ();
8     gid_t gid = getegid ();
9     printf ("uid=%d, gid=%d\n", uid, gid);
10    return 0;
11 }
```

Figure 4: Создание программы. Программа simpleid.c

```
[guest@ymgorbunova dir1]$ touch simpleid2.c
[guest@ymgorbunova dir1]$ gcc simpleid2.c -o simpleid2
simpleid2.c: В функции «main»:
simpleid2.c:13:1: ошибка: expected expression before «,» token
   13 | ^ →
      | ^
simpleid2.c:13:3: ошибка: в программе обнаружен некорректный символ «\342»
   13 | ^ →
      | ^
[guest@ymgorbunova dir1]$ gcc simpleid2.c -o simpleid2
[guest@ymgorbunova dir1]$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
```

Figure 5: Создание программы. Пункты 6-7



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

Figure 6: Создание программы. Программа simpleid2.c

```
[root@ymgorbunova ~]# chown root:guest /home/guest/dir1/simpleid2
[root@ymgorbunova ~]# chmod u+s /home/guest/dir1/simpleid2
[root@ymgorbunova ~]# ls -l simpleid2
ls: невозможно получить доступ к 'simpleid2': Нет такого файла или каталога
[root@ymgorbunova ~]# ls -l /home/guest/dir1/simpleid2
-rwsrwxr-x. 1 root guest 26008 сен 28 14:53 /home/guest/dir1/simpleid2
```

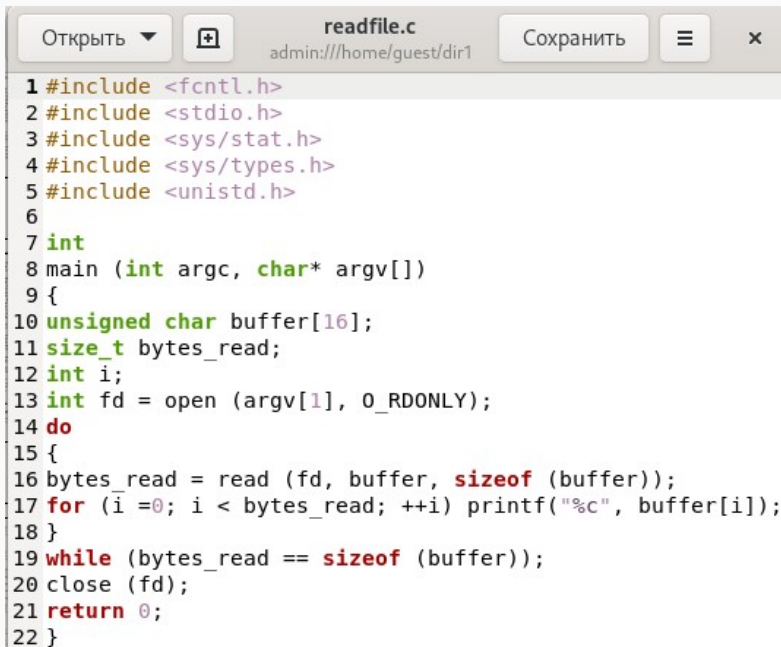
Figure 7: Создание программы. Пункты 8-10

```
[root@ymgorbunova dir1]# ./simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@ymgorbunova dir1]# id
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Figure 8: Создание программы. Пункт 11

```
[root@ymgorbunova dir1]# chmod g+s /home/guest/simpleid2
chmod: невозможно получить доступ к '/home/guest/simpleid2': Нет такого файла или каталога
[root@ymgorbunova dir1]# chmod g+s simpleid2
[root@ymgorbunova dir1]# ls -l simpleid2
-rwsrwsr-x. 1 root guest 26008 сен 28 14:53 simpleid2
[root@ymgorbunova dir1]# ls -l
итого 64
-rwxrwxr-x. 1 guest guest 25904 сен 28 14:49 simpleid
-rwsrwsr-x. 1 root guest 26008 сен 28 14:53 simpleid2
-rw-rw-r--. 1 guest guest 303 сен 28 14:52 simpleid2.c
-rw-rw-r--. 1 guest guest 175 сен 28 14:49 simpleid.c
[root@ymgorbunova dir1]# ./simpleid2
e_uid=0, e_gid=1001
real_uid=0, real_gid=0
[root@ymgorbunova dir1]# id
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Figure 9: Создание программы. Пункт 12



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

```
[guest@ymgorbunova dir1]$ gcc readfile.c -o readfile
[guest@ymgorbunova dir1]$ ls -l
итого 96
-rwxrwxr-x. 1 guest guest 25952 сен 28 22:54 readfile
-rw-rw-r--. 1 guest guest  403 сен 28 22:54 readfile.c
-rwxrwxr-x. 1 guest guest 25904 сен 28 14:49 simpleid
-rwsrwsr-x. 1 root  guest 26008 сен 28 14:53 simpleid2
-rw-rw-r--. 1 guest guest  303 сен 28 14:52 simpleid2.c
-rw-rw-r--. 1 guest guest  175 сен 28 14:49 simpleid.c
```

Figure 11: Создание программы. Пункт 14

```
[root@ymgorbunova dir1]# chown root:guest readfile.c
[root@ymgorbunova dir1]# ls -l
итого 96
-rwxrwxr-x. 1 guest guest 25952 сен 28 22:54 readfile
-rw-rw-r--. 1 root  guest  403 сен 28 22:54 readfile.c
-rwxrwxr-x. 1 guest guest 25904 сен 28 14:49 simpleid
-rwsrwsr-x. 1 root  guest 26008 сен 28 14:53 simpleid2
-rw-rw-r--. 1 guest guest  303 сен 28 14:52 simpleid2.c
-rw-rw-r--. 1 guest guest  175 сен 28 14:49 simpleid.c
[root@ymgorbunova dir1]# chmod 600 readfile.c
[root@ymgorbunova dir1]# ls -l
итого 96
-rwxrwxr-x. 1 guest guest 25952 сен 28 22:54 readfile
-rw-----. 1 root  guest  403 сен 28 22:54 readfile.c
-rwxrwxr-x. 1 guest guest 25904 сен 28 14:49 simpleid
-rwsrwsr-x. 1 root  guest 26008 сен 28 14:53 simpleid2
-rw-rw-r--. 1 guest guest  303 сен 28 14:52 simpleid2.c
-rw-rw-r--. 1 guest guest  175 сен 28 14:49 simpleid.c
```

Figure 12: Создание программы. Пункт 15


```
[guest@ymgorbunova dir1]$ cat readfile.c  
cat: readfile.c: Отказано в доступе
```

Figure 13: Создание программы. Пункт 16

```
[root@ymgorbunova dir1]# chown root:guest readfile  
[root@ymgorbunova dir1]# chmod u+s readfile  
[root@ymgorbunova dir1]# ls -l  
итого 96  
-rwsrwxr-x. 1 root  guest 25952 сен 28 22:54 readfile  
-rw----- 1 root  guest   403 сен 28 22:54 readfile.c  
-rwxrwxr-x. 1 guest  guest 25904 сен 28 14:49 simpleid  
-rwsrwsr-x. 1 root  guest 26008 сен 28 14:53 simpleid2  
-rw-rw-r-- 1 guest  guest   303 сен 28 14:52 simpleid2.c  
-rw-rw-r-- 1 guest  guest   175 сен 28 14:49 simpleid.c
```

Figure 14: Создание программы. Пункт 17

```
[root@ymgorbunova dir1]# ./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;
}
```

```
[root@ymgorbunova dir1]# ./readfile /etc/shadow
root:$6$67zd8mWm4E6RCrcf$A8rm6YQPZ5XDKNFqtDM7EKLZXLeF.M6t15rt3QkhsJzVNNQA2mFMpy9/zzYKR.hAtnwzc449IRRBv7Z
tx0oQ4.: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:!!!!:
```

Figure 16: Создание программы. Пункт 19

```
[guest@ymgorbunova ~]$ ls -l / | grep tmp
drwxrwxrwt. 17 root root 4096 сен 28 23:36 tmp
[guest@ymgorbunova ~]$ echo "test" > /tmp/file01.txt
[guest@ymgorbunova ~]$ ls -l /tmp/file01.txt
-rw-rw-r--. 1 guest guest 5 сен 28 23:40 /tmp/file01.txt
[guest@ymgorbunova ~]$ chmod o+rw /tmp/file01.txt
[guest@ymgorbunova ~]$ ls -l /tmp/file01.txt
-rw-rw-rw-. 1 guest guest 5 сен 28 23:40 /tmp/file01.txt
[guest@ymgorbunova ~]$
```

Figure 17: Исследование Sticky-бита. Пункты 1-3

```
[guest2@ymgorbunova ~]$ cat /tmp/file01.txt
test
[guest2@ymgorbunova ~]$ echo "test2" >> /tmp/file01.txt
[guest2@ymgorbunova ~]$ cat /tmp/file01.txt
test
test2
[guest2@ymgorbunova ~]$ echo "test3" > /tmp/file01.txt
[guest2@ymgorbunova ~]$ cat /tmp/file01.txt
test3
```

Figure 18: Исследование Sticky-бита. Пункты 4-8

```
[guest2@ymgorbunova ~]$ rm /tmp/file01.txt
rm: невозможно удалить '/tmp/file01.txt': Операция не позволена
[guest2@ymgorbunova ~]$ su
Пароль:
[root@ymgorbunova guest2]# chmod -t /tmp
[root@ymgorbunova guest2]# exit
exit
[guest2@ymgorbunova ~]$ ls -l / | grep tmp
drwxrwxrwx. 17 root root 4096 сен 28 23:45 tmp
[guest2@ymgorbunova ~]$ rm /tmp/file01.txt
[guest2@ymgorbunova ~]$ ls -l file01.txt
ls: невозможно получить доступ к 'file01.txt': Нет такого файла или каталога
[guest2@ymgorbunova ~]$ ls -l /tmp/file01.txt
ls: невозможно получить доступ к '/tmp/file01.txt': Нет такого файла или каталога
```

Figure 19: Исследование Sticky-бита. Пункты 9-14

```
[guest2@ymgorbunova ~]$ su
Пароль:
[root@ymgorbunova guest2]# chmod +t /tmp
[root@ymgorbunova guest2]# ls -l / | grep tmp
drwxrwxrwt. 17 root root 4096 сен 28 23:47 tmp
[root@ymgorbunova guest2]# exit
exit
[guest2@ymgorbunova ~]$
```

Figure 20: Исследование Sticky-бита. Пункт 15

Результаты

Изучены механизмы изменения идентификаторов, применения SetUID- и Sticky-битов. Получены практические навыки работы в консоли с дополнительными атрибутами. Рассмотрена работа механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

Список литературы

1. Методические материалы курса