

Лабораторная Работа №4. Дискреционное разграничение прав в Linux. Расширенные атрибуты

Операционные системы

Дудырев Г. А.

Российский университет дружбы народов им. Патриса Лумумбы, Москва, Россия

- Дудырев Глеб Андреевич
- НПИБд-01-22
- Российский университет дружбы народов
- [1132222013@pfur.ru]

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

Создаю файл 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;
}
~
~
~
~
~
~
~
~
~
~
~

```

-- ВСТАВКА --

11,11-18

Весь

Figure 1: Программа simpleid.c

Компилирую программу командой `gcc simpleid.c -o simpleid` и проверяю, что файл создан. Выполняю программу `simpleid` командой `./simpleid`, а затем системную программу `id` - вывод одинаков.

```
[guest@gadudihrev ~]$ vim simpleid.c
[guest@gadudihrev ~]$ gcc simpleid.c -o simpleid
gcc: ошибка: unrecognized command-line option «-o»
[guest@gadudihrev ~]$ gcc simpleid.c -o simpleid
simpleid.c: В функции «main»:
simpleid.c:10:15: ошибка: expected «;» before «{» token
   10 |         printf {"uid=%d, gid=%d\n", uid, gid};
      |                  ^~
      |                  ;
[guest@gadudihrev ~]$ vim simpleid.c
[guest@gadudihrev ~]$ gcc simpleid.c -o simpleid
simpleid.c: В функции «main»:
simpleid.c:10:46: ошибка: expected «;» before «return»
   10 |         printf ("uid=%d, gid=%d\n", uid, gid)
      |                                                    ^
   11 |         return 0;
      |         ~~~~~
[guest@gadudihrev ~]$ vim simpleid.c
[guest@gadudihrev ~]$ gcc simpleid.c -o simpleid
[guest@gadudihrev ~]$ ./simpleid
uid=1001, gid=1001
[guest@gadudihrev ~]$
```

Figure 2: Выполнение программы

Усложняю программу и записываю ее в файл simpleid2.c



```
guest@gadudihrev:~ — vim simpleid2.c

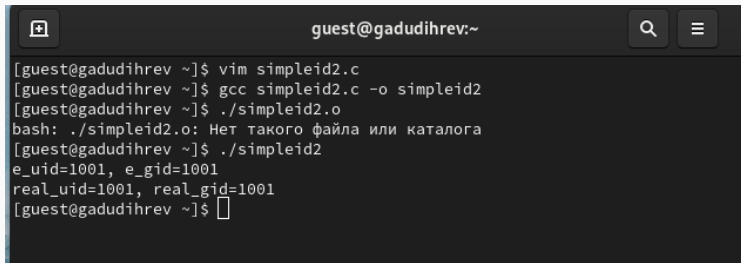
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>

int
main ()
{
    uid resl_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;
}

~
~
~
~
~
~
-- ВСТАВКА -- 14,18  Весь
```

Figure 3: Программа simpleid2.c

Компилирую и запускаю программу командами `gcc simpleid2.c -o simpleid2` и `./simpleid2`



```
guest@gadudihrev:~  
[guest@gadudihrev ~]$ vim simpleid2.c  
[guest@gadudihrev ~]$ gcc simpleid2.c -o simpleid2  
[guest@gadudihrev ~]$ ./simpleid2.o  
bash: ./simpleid2.o: Нет такого файла или каталога  
[guest@gadudihrev ~]$ ./simpleid2  
e_uid=1001, e_gid=1001  
real_uid=1001, real_gid=1001  
[guest@gadudihrev ~]$
```

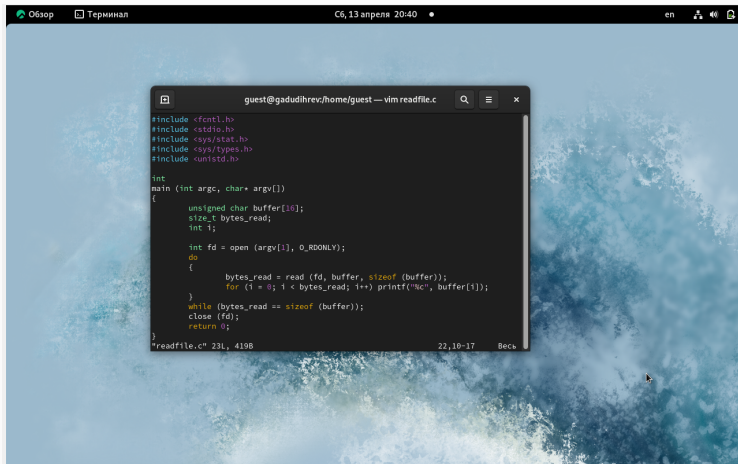
Figure 4: Компиляция и запуск

От суперпользователя выполняю команды `chown root:guest /home/guest/simpleid2` и `chmod u+s /home/guest/simpleid2`. Проверяю правильность новых атрибутов командой `ls -l simpleid2`. Запускаю `simpleid2` и `id: ./simpleid2, id`

```
[guest@gadudihrev ~]$ su
Пароль:
[root@gadudihrev guest]# chown root:guest /home/guest/simpleid2
[root@gadudihrev guest]# chmod u+s /home/guest/simpleid2
[root@gadudihrev guest]# ls -l simpleid2
-rwsr-xr-x. 1 root guest 24488 апр 13 20:23 simpleid2
[root@gadudihrev guest]# ./simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@gadudihrev guest]# id
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@gadudihrev guest]#
```

Figure 5: Изменение атрибутов, запуск

Создаю программу readfile.c



The screenshot shows a terminal window with a dark theme. The title bar of the terminal window reads "C6, 13 апреля 20:40". Inside the terminal, a vim editor window is open, titled "guest@gadudihrev/home/guest — vim readfile.c". The editor displays the following C code:

```
#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[10];
    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;
}
```

At the bottom of the vim window, the status line shows "readfile.c" 23L, 419B on the left and "22,10-17" and "Бек" on the right.

Figure 6: Программа readfile.c

Компилирую ее командой `gcc readfile.c -o readfile` и изменяю права доступа так, чтобы только суперпользователь мог прочитать его, а `guest` не мог

```
[root@gadudihrev guest]# chown root:guest /home/guest/readfile.c
[root@gadudihrev guest]# chmod 700 readfile.c
bash: chmod: command not found...
Similar command is: 'chmod'
[root@gadudihrev guest]# chmod 700 readfile.c
[root@gadudihrev guest]# ls -l /home/guest/readfile.c
-rwx-----. 1 root guest 419 anp 13 20:40 /home/guest/readfile.c
[root@gadudihrev guest]#
```

Figure 7: Компиляция программы, смена прав доступа

Командой `cat readfile.c` проверяю, что пользователь `guest` не может прочитать файл `readfile.c`. Устанавливаю SetU'D-бит и теперь от пользователя `guest` можно прочитать файл

```
[root@gadudihrev guest]# su guest
[guest@gadudihrev ~]$ cat readfile.c
cat: readfile.c: Отказано в доступе
[guest@gadudihrev ~]$ su
Пароль:
[root@gadudihrev guest]# chmod u+s /home/guest/readfile.c
[root@gadudihrev guest]# readfile readfile.c
bash: readfile: command not found...
[root@gadudihrev 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;

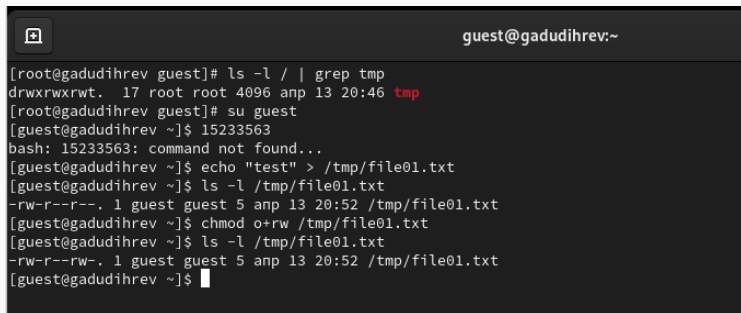
    int fd = open (argv[1], O_RDONLY);
    do
    {
        bytes_read = read (fd, buffer, sizeof (buffer));
        for (i = 0; i < bytes_read; i++) printf("%c", buffer[i]);
    }
    while (bytes_read == sizeof (buffer));
    close (fd);
    return 0;
}
[root@gadudihrev guest]#
```

Проверяю, может ли программа readfile прочитать файл /etc/shadow - да, может

```
[root@gadudihrev guest]# ./readfile /etc/shadow
root:$6$1WRz..Hlah4g8YS8$vZUdYL1JycC8MK.4/FohMXML8bCUa0NSLwBzMoxXk6l.upogud02tHucQrgRI/AtUoxl6N6QRna4R6F9LjKw20::0:99999:7:::
bin:!:19469:0:99999:7:::
daemon:!:19469:0:99999:7:::
adm:!:19469:0:99999:7:::
lp:!:19469:0:99999:7:::
sync:!:19469:0:99999:7:::
shutdown:!:19469:0:99999:7:::
halt:!:19469:0:99999:7:::
mail:!:19469:0:99999:7:::
operator:!:19469:0:99999:7:::
games:!:19469:0:99999:7:::
ftp:!:19469:0:99999:7:::
nobody:!:19469:0:99999:7:::
systemd-coredump:!!:19811:::
dbus:!!:19811:::
polkitd:!!:19811:::
avahi:!!:19811:::
rtkit:!!:19811:::
```

Figure 9: Файл /etc/shadow

Проверяю, установлен ли атрибут Sticky на директории /tmp командой `ls -l / | grep tmp`. От пользователя guest создаю файл со словом test командой `echo "test" > /tmp/file01.txt`. Просматриваю атрибуты у только что созданного файла и разрешаю чтение и запись для категории пользователей «все остальные»



```
guest@gadudihrev:~  
[root@gadudihrev guest]# ls -l / | grep tmp  
drwxrwxrwt. 17 root root 4096 anp 13 20:46 tmp  
[root@gadudihrev guest]# su guest  
[guest@gadudihrev ~]$ 15233563  
bash: 15233563: command not found...  
[guest@gadudihrev ~]$ echo "test" > /tmp/file01.txt  
[guest@gadudihrev ~]$ ls -l /tmp/file01.txt  
-rw-r--r--. 1 guest guest 5 anp 13 20:52 /tmp/file01.txt  
[guest@gadudihrev ~]$ chmod o+rw /tmp/file01.txt  
[guest@gadudihrev ~]$ ls -l /tmp/file01.txt  
-rw-r--rw-. 1 guest guest 5 anp 13 20:52 /tmp/file01.txt  
[guest@gadudihrev ~]$
```

Figure 10: Проверка атрибута, работа с файлом

От пользователя `guest` пробую прочитать файл командой `cat /tmp/file01.txt`, далее записываю в файл слово `test2` и вновь читаю его - текст файла изменен

```
[guest2@gadudihrev guest]$ cat /tmp/file01.txt
test
[guest2@gadudihrev guest]$ echo "test" > /tmp/file01.txt
[guest2@gadudihrev guest]$ cat /tmp/file01.txt
test
[guest2@gadudihrev guest]$ echo "test2" > /tmp/file01.txt
[guest2@gadudihrev guest]$ cat /tmp/file01.txt
test2
[guest2@gadudihrev guest]$
```

Figure 11: Действия с файлом от другого пользователя

От пользователя `guest2` пробую записать в файл `/tmp/file01.txt` слово `test3`, стерев при этом всю имеющуюся в файле информацию командой `echo "test3" > /tmp/file01.txt` - операцию выполнить удалось. Просматриваю содержимое файла и пробую удалить его - удалить не удалось

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

Figure 12: Действия с файлом от другого пользователя

От суперпользователя ввожу команду, снимающую атрибут `t` (Sticky-бит) с директории `/tmp`: `chmod -t /tmp`. Проверяю от пользователя `guest2`, что атрибута `t` у директории `/tmp` нет командой `ls -l | grep tmp`

```
[guest2@gadudihrev guest]$ su
Пароль:
[root@gadudihrev guest]# chmod -t /tmp
[root@gadudihrev guest]# exit
exit
[guest2@gadudihrev guest]$ ls -l | grep tmp
[guest2@gadudihrev guest]$ ls -l / | grep tmp
drwxrwxrwx. 17 root root 4096 апр 13 21:05 tmp
[guest2@gadudihrev guest]$
```

Figure 13: Снятие Sticky-бита с директории

Снова пробуем записать, прочитать и удалить файл - все операции выполнены успешно

```
[guest2@gadudihrev guest]$ cat /tmp/file01.txt  
test3  
[guest2@gadudihrev guest]$ echo "test4" > /tmp/file01.txt  
[guest2@gadudihrev guest]$ cat /tmp/file01.txt  
test4  
[guest2@gadudihrev guest]$ rm /tmp/file01.txt  
[guest2@gadudihrev guest]$
```

Figure 14: Запись, чтение и удаление

Возвращаюсь в суперпользователя и возвращаю атрибут `t` на директорию `/tmp` командой `chmod +t /tmp`

```
[guest2@gadudihrev guest]$ su
Пароль:
[root@gadudihrev guest]# chmod +t /tmp
[root@gadudihrev guest]# exit
exit
[guest2@gadudihrev guest]$ ls -l / | grep tmp
drwxrwxrwt. 17 root root 4096 anp 13 21:07 tmp
[guest2@gadudihrev guest]$
```

Figure 15: Возвращение атрибута `t`

Выводы

Я научился применять SetUID- и Sticky-биты, поработал с дополнительными атрибутами в консоли, рассмотрел работу механизма смены идентификатора процессов пользователей и влияние бита Sticky на запись и удаление файлов.