

# Лабораторная работа №5

---

Бешкуров Михаил

13.11.2021

# Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

---

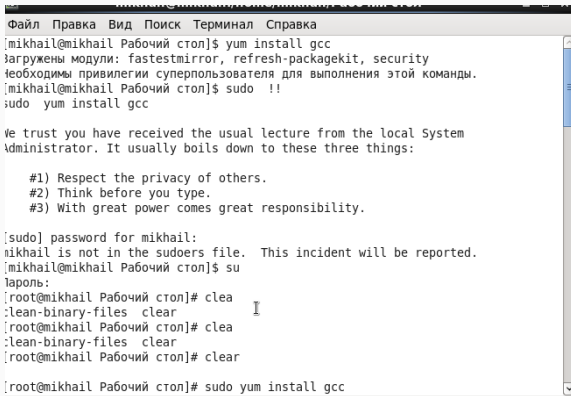
- Помимо прав администратора в некоторых случаях требуются средства разработки приложений. Для разграничения их прав нужно использовать дополнительные атрибуты.

- 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.

- Подготовить лабораторный стенд
- Рассмотреть компиляцию программ
- Создать программы
- Исследовать Sticky-бит

# Результаты выполнения лабораторной работы

- Подготовил лабораторный стенд (рис - @fig:001).

A screenshot of a terminal window with a menu bar at the top containing 'Файл', 'Правка', 'Вид', 'Поиск', 'Терминал', and 'Справка'. The terminal shows a user named 'mikhail' at a prompt '[mikhail@mikhail Рабочий стол]' running the command 'yum install gcc'. The output indicates that modules 'fastestmirror', 'refresh-packagekit', and 'security' are loaded, and that root privileges are required. The user then runs 'sudo yum install gcc'. The terminal displays a message from the local System Administrator about receiving a lecture from the local System Administrator, listing three things: #1) Respect the privacy of others, #2) Think before you type, and #3) With great power comes great responsibility. The user then enters the password for 'mikhail', but is informed that 'mikhail' is not in the sudoers file. The user then switches to root using 'su', and the prompt changes to '[root@mikhail Рабочий стол]'. The user runs 'clean-binary-files clear' three times. Finally, the user runs 'sudo yum install gcc' as root.

```
Файл Правка Вид Поиск Терминал Справка
[mikhail@mikhail Рабочий стол]$ yum install gcc
Загружены модули: fastestmirror, refresh-packagekit, security
Необходимы привилегии суперпользователя для выполнения этой команды.
[mikhail@mikhail Рабочий стол]$ sudo !!
sudo yum install gcc

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

[sudo] password for mikhail:
mikhail is not in the sudoers file. This incident will be reported.
[mikhail@mikhail Рабочий стол]$ su
Пароль:
[root@mikhail Рабочий стол]# clean
clean-binary-files clear
[root@mikhail Рабочий стол]# clea
clean-binary-files clear
[root@mikhail Рабочий стол]# clear
[root@mikhail Рабочий стол]# sudo yum install gcc
```

Рис. 1: Установка компилятора gcc

- Провел работу с программами (рис @fig:003, рис @fig:004, рис @fig:005).

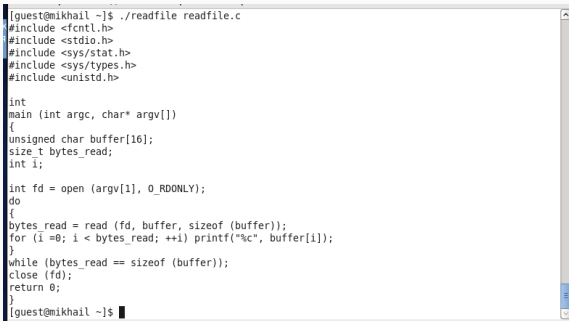
```
[guest@mikhail Рабочий стол]$ su
Пароль:
[root@mikhail Рабочий стол]# chown root:guest /home/guest/simpleid2
chown: невозможно получить доступ к «/home/guest/simpleid2»: Нет такого файла ил
и каталога
[root@mikhail Рабочий стол]# chown root:guest /home/guest/Рабочий\ стол/simplei
d2
[root@mikhail Рабочий стол]# chmod u+s /home/guest/Рабочий\ стол/simpleid2
```

Рис. 2: Смена владельца и атрибутов от имени суперпользователя

```
[guest@mikhail ~]$ gcc readfile.c -o readfile
[guest@mikhail ~]$ ls -l readfile.c
-rw-rw-r--. 1 guest guest 404 Ноя 13 22:16 readfile.c
[guest@mikhail ~]$ █
```

Рис. 3: Компиляция программы `readfile.c`





```
[guest@mikhail ~]$ ./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;
}
[guest@mikhail ~]$
```

Рис. 4: Проверка чтения файла

- Исследовал Sticky-бит (рис @fig:006, рис @fig:007, рис @fig:008).

```
[guest@mikhail ~]$ ls -l | grep tmp
[guest@mikhail ~]$ echo "test" > /tmp/file01.txt
[guest@mikhail ~]$ ls -l /tmp/file01.txt
-rw-rw-r--. 1 guest guest 5 Ноя 13 22:25 /tmp/file01.txt
[guest@mikhail ~]$ chmod o+rw /tmp/file01.txt
[guest@mikhail ~]$ ls -l /tmp/file01.txt
-rw-rw-rw-. 1 guest guest 5 Ноя 13 22:25 /tmp/file01.txt
[guest@mikhail ~]$ su - guest2
Пароль:
[guest2@mikhail ~]$ cat /tmp/file01.txt
test
[guest2@mikhail ~]$ echo "test" > /tmp/file01.txt
[guest2@mikhail ~]$ cat /tmp/file01.txt
test
[guest2@mikhail ~]$ echo "test" >> /tmp/file01.txt
[guest2@mikhail ~]$ cat /tmp/file01.txt
test
test
[guest2@mikhail ~]$ echo "test3" > /tmp/file01.txt
[guest2@mikhail ~]$ cat /tmp/file01.txt
test3
[guest2@mikhail ~]$ rm /tmp/file01.txt
rm: невозможно удалить «/tmp/file01.txt»: Операция не допускается
[guest2@mikhail ~]$ █
```

Рис. 5: Попытка удаления файла от имени пользователя guest2

```
[guest2@mikhail ~]$ su -  
Пароль:  
[root@mikhail ~]# chmod -t /tmp  
[root@mikhail ~]# exit  
logout
```

Рис. 6: Смена бита

```
[guest2@mikhail ~]$ cat /tmp/file01.txt
test3
[guest2@mikhail ~]$ echo "test" >> /tmp/file01.txt
[guest2@mikhail ~]$ cat /tmp/file01.txt
test3
test
[guest2@mikhail ~]$ echo "test5" > /tmp/file01.txt
[guest2@mikhail ~]$ cat /tmp/file01.txt
test5
[guest2@mikhail ~]$ rm /tmp/file01.txt
[guest2@mikhail ~]$
```

Рис. 7: Проведение различных операций после удаления атрибута t

Таким образом, 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.