

Лабораторная работа 5

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

Бешкуров Михаил Борисович

Содержание

1	Цель работы	3
2	Задание	4
3	Выполнение лабораторной работы	5
4	Выводы	12
5	Список литературы	13

1 Цель работы

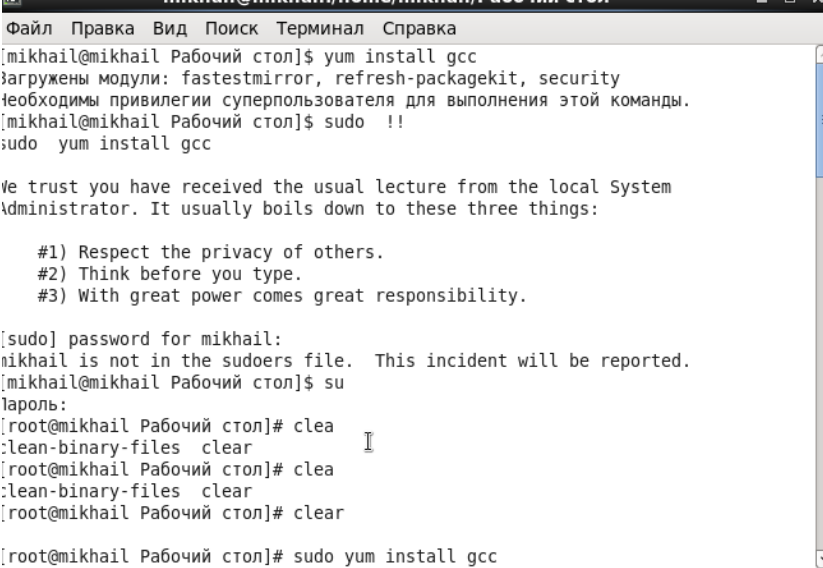
Изучение механизмов изменения идентификаторов, применения SetUID- и Sticky-битов. Получение практических навыков работы в консоли с дополнительными атрибутами. Рассмотрение работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов. [1]

2 Задание

1. Подготовить лабораторный стенд
2. Рассмотреть компиляцию программ
3. Создать программы
4. Исследовать Sticky-бит

3 Выполнение лабораторной работы

1. Предварительно установил компилятор gcc с помощью команды `yum install gcc` (рис - @fig:001).



```
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.

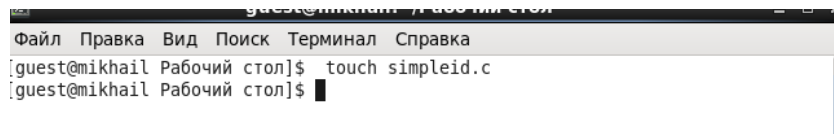
[sudo] password for mikhail:
mikhail is not in the sudoers file. This incident will be reported.
[mikhail@mikhail Рабочий стол]$ su
Пароль:
[root@mikhail Рабочий стол]# clea
:lean-binary-files clear
[root@mikhail Рабочий стол]# clea
:lean-binary-files clear
[root@mikhail Рабочий стол]# clear
[root@mikhail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

Рис. 3.1: Установка компилятора gcc

Отключил систему защиты SELinux с помощью команды `setenforce 0`. После этого команда `getenforce` вывела `Permissive`.

2. Изучил компиляцию программ. Компилятор языка C называется gcc. Компилятор языка C++ называется g++ и запускается с параметрами почти так же, как gcc. Проверил это с помощью команд `whereis gcc` и `whereis g++`.

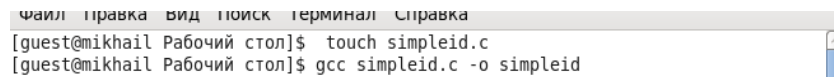
3. Вошел в систему от имени пользователя `guest` и создал программу `simpleid.c`. (рис - @fig:002)



```
guest@mikhail Рабочий стол
Файл Правка Вид Поиск Терминал Справка
[guest@mikhail Рабочий стол]$ touch simpleid.c
[guest@mikhail Рабочий стол]$
```

Рис. 3.2: Создание программы `simpleid.c`

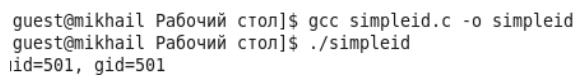
Скомпилировал программу и убедился, что файл программы создан с помощью команды `gcc simpleid.c -o simpleid` (рис @fig:003)



```
Файл Правка Вид Поиск Терминал Справка
[guest@mikhail Рабочий стол]$ touch simpleid.c
[guest@mikhail Рабочий стол]$ gcc simpleid.c -o simpleid
```

Рис. 3.3: Компиляция программы

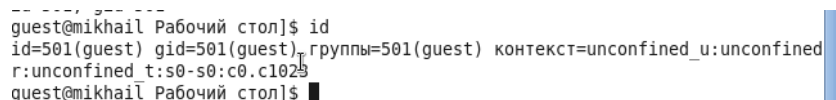
Выполнил программу `simpleid` (рис @fig:004)



```
guest@mikhail Рабочий стол$ gcc simpleid.c -o simpleid
guest@mikhail Рабочий стол$ ./simpleid
id=501, gid=501
```

Рис. 3.4: Выполнение созданной программы

Выполнил системную программу `id` (рис @fig:005)



```
-----
guest@mikhail Рабочий стол$ id
id=501(guest) gid=501(guest) группы=501(guest) контекст=unconfined_u:unconfined
r:unconfined_t:s0-s0:c0.c1023
guest@mikhail Рабочий стол$
```

Рис. 3.5: Выполнение системной программы `id`

Вывод обоих способов совпадает.

Усложнил программу, добавив вывод действительных идентификаторов

Получившуюся программу назвал `simpleid2.c`

Скомпилировал и запустил `simpleid2.c` (рис @fig:006)

```

[guest@mikhail Рабочий стол]$ gcc simpleid2.c -o simpleid2
[guest@mikhail Рабочий стол]$ ./simpleid2
uid=501, e_gid=501
real_uid=501, real_gid=501
[guest@mikhail Рабочий стол]$

```

Рис. 3.6: Компиляция и запуск файла

От имени суперпользователя выполнил следующие команды (рис @fig:007)

```

[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 и каталога
[root@mikhail Рабочий стол]# chown root:guest /home/guest/Рабочий\ стол/simpleid2
[root@mikhail Рабочий стол]# chmod u+s /home/guest/Рабочий\ стол/simpleid2

```

Рис. 3.7: Смена владельца и атрибутов от имени суперпользователя

Команда `su` используется для получения прав суперпользователя.

Выполнил проверку правильности установки новых атрибутов и смены владельца файла `simpleid2` (рис @fig:008)

Запустил `simpleid2` и `id` (рис @fig:008)

```

[root@mikhail Рабочий стол]# ls -l simpleid2
-rwsrwxr-x. 1 root guest 5143 Ноя 13 22:09 simpleid2
[root@mikhail Рабочий стол]# ./simpleid2
uid=0, e_gid=0
real_uid=0, real_gid=0
[root@mikhail Рабочий стол]# id
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0-c1023
[root@mikhail Рабочий стол]#

```

Рис. 3.8: Вывод

Проделал тоже самое относительно SetGID-бита

Создал программу `readfile.c`

Откомпилировал созданную программу (рис @fig:009)

```

[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. 3.9: Компиляция программы

Сменил владельца у файла `readfile.c` и изменил права так, чтобы только суперпользователь мог прочитать его, а `guest` не мог

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

```
[guest@mikhail ~]$ su
Пароль:
[root@mikhail guest]# chown root:root /home/guest/readfile.c
[root@mikhail guest]# chmod 700 /home/guest/readfile.c
[root@mikhail guest]# su - guest
[guest@mikhail ~]$ cat readfile.c
cat: readfile.c: Отказано в доступе
[guest@mikhail ~]$
```

Рис. 3.10: Смена прав и попытка прочесть файл

Сменил у программы `readfile` владельца и установил SetUID-бит

Проверил, может ли программа `readfile` прочитать файл `readfile.c`. Да, может.

```
[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. 3.11: Проверка чтения файла

Проверил, может ли программа `readfile` прочитать файл `/etc/shadow`. Да, может.


```
[guest@mikhail ~]$ ./readfile /etc/shadow
root:$6$LZ65/1sC15IBCjA$QjFjpvb47BbfXeZcrH8qDCzLEpxy2jr2dMyXl4X3v4d2MXgydwceMHa0tTYKkFS76LPvk
fpBCK6FzU9XVvxW/:18888:0:99999:7:::
bin:!:15980:0:99999:7:::
daemon:!:15980:0:99999:7:::
adm:!:15980:0:99999:7:::
lp:!:15980:0:99999:7:::
sync:!:15980:0:99999:7:::
shutdown:!:15980:0:99999:7:::
halt:!:15980:0:99999:7:::
mail:!:15980:0:99999:7:::
uucp:!:15980:0:99999:7:::
operator:!:15980:0:99999:7:::
games:!:15980:0:99999:7:::
gopher:!:15980:0:99999:7:::
ftp:!:15980:0:99999:7:::
nobody:!:15980:0:99999:7:::
dbus:!!:18888:0:99999:7:::
usbmuxd:!!:18888:0:99999:7:::
vcsa:!!:18888:0:99999:7:::
rpc:!!:18888:0:99999:7:::
rtkit:!!:18888:0:99999:7:::
avahi-autoipd:!!:18888:0:99999:7:::
pulse:!!:18888:0:99999:7:::
```

Рис. 3.12: Проверка чтения файла /etc/shadow

4. Исследовал Sticky-бит Выяснил, что атрибут Sticky установлен на директорию /tmp, для чего выполнил команду `ls -l / | grep tmp`

От имени пользователя guest создал файл file01.txt в директории /tmp со словом test:

Просмотрел атрибуты у только что созданного файла и разрешил чтение и запись для категории пользователей “все остальные”:

```
[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 ~]$
```

Рис. 3.13: Просмотр атрибутов файла и установление прав на чтение и запись для категории “все остальные”

От имени пользователя guest2 (не являющегося владельцем) прочитал файл /tmp/file01.txt:

От имени пользователя guest2 дозаписал в файл /tmp/file01.txt слово test2:

Проверил содержимое файла:

От имени пользователя `guest2` записал в файл `/tmp/file01.txt` слово `test3`, стерев при этом всю имеющуюся в файле информацию:

Проверил содержимое файла:

От имени пользователя `guest2` попробовал удалить файл `/tmp/file01.txt`:

Мне не удалось удалить файл.

```
[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. 3.14: Выполнение операций

Повысил свои права до суперпользователя и выполнил после этого команду, снимающую атрибут `t` (Sticky-бит) с директории `/tmp` (рис @fig:035):

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

Рис. 3.15: Повышение прав до суперпользователя. Снятие атрибута `t`

Покинул режим суперпользователя командой `exit`:

Повторил предыдущие шаги:

```
[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. 3.16: Повтор предыдущих шагов

Как видно из рисунка, удалось выполнить все команды, которые были рассмотрены выше, включая удаление.

Повысил свои права до суперпользователя и вернул атрибут `t` на директорию `/tmp`:

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

Рис. 3.17: Переход в режим суперпользователя и возврат атрибута `t`

4 Выводы

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.

5 Список литературы

1. Кулябов Д. С., Королькова А. В., Геворкян М. Н. Информационная безопасность компьютерных сетей. Лабораторная работа № 5. Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов