

Лабораторная работа №5

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

Роман Владимирович Иванов

Содержание

1	Цель работы	6
2	Задание	7
3	Выполнение лабораторной работы	8
4	Выводы	20
5	Список литературы	21

Список иллюстраций

3.1	Установка компилятора gcc	8
3.2	Отключение системы запретов	8
3.3	Проверка названий компиляторов	9
3.4	Создание программы simpleid.c (часть 1)	9
3.5	Создание программы simpleid.c (часть 2)	9
3.6	Компиляция программы	10
3.7	Выполнение созданной программы	10
3.8	Выполнение системной программы id	10
3.9	Усложнение программы	11
3.10	Переименование программы	11
3.11	Компиляция и запуск файла	11
3.12	Смена владельца и атрибутов от имени суперпользователя	12
3.13	Использование оператора su	12
3.14	Проверка правильности установления атрибутов	12
3.15	Проверка id пользователя и группы	12
3.16	Повторение операций относительно SetGID-бита	13
3.17	Создание программы readfile.c (часть 1)	13
3.18	Создание программы readfile.c (часть 2)	13
3.19	Компиляция программы	13
3.20	Смена владельца и изменение прав файла	14
3.21	Попытка прочесть файл	14
3.22	Смена владельца и установка SetUID-бита	14
3.23	Проверка чтения файла (часть 1)	14
3.24	Проверка чтения файла (часть 2)	15
3.25	Проверка чтения файла /etc/shadow	15
3.26	Проверка нахождения атрибута Sticky на директории /tmp	16
3.27	Создание файла и внесение записи в него	16
3.28	Просмотр атрибутов файла и установление прав на чтение и запись для категории “все остальные”	16
3.29	Чтение файла от имени пользователя guest2	16
3.30	Дозапись слова в файл от имени пользователя guest2	17
3.31	Проверка содержимого в файле от имени пользователя guest2	17
3.32	Перезапись информации в файл от имени пользователя guest2	17
3.33	Проверка содержимого в файле от имени пользователя guest2	17
3.34	Попытка удаления файла от имени пользователя guest2	18
3.35	Повышение прав до суперпользователя. Снятие атрибута t	18
3.36	Выход из режима суперпользователя	18

3.37 Проверка отсутствия атрибута t	18
3.38 Повтор предыдущих шагов	19
3.39 Переход в режим суперпользователя и возврат атрибута t	19

Список таблиц

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).

```
[rvivanov@rvivanov Рабочий стол]$ gcc -v
bash: gcc: команда не найдена
[rvivanov@rvivanov Рабочий стол]$ su
Пароль:
[root@rvivanov Рабочий стол]# yum install gcc
Загружены модули: fastestmirror, refresh-packagekit, security
Подготовка к установке
Determining fastest mirrors
base | 3.7 kB | 00:00
extras | 3.3 kB | 00:00
updates | 3.4 kB | 00:00
Разрешение зависимостей
--> Проверка сценария
--> Package gcc.i686 0:4.4.7-23.el6 will be для установки
--> Обработка зависимостей: cpp = 4.4.7-23.el6 для пакета: gcc-4.4.7-23.el6.i686
--> Обработка зависимостей: cloog-ppl >= 0.15 для пакета: gcc-4.4.7-23.el6.i686
--> Проверка сценария
--> Package cloog-ppl.i686 0:0.15.7-1.2.el6 will be для установки
--> Обработка зависимостей: libppl_c.so.2 для пакета: cloog-ppl-0.15.7-1.2.el6.i686
--> Обработка зависимостей: libppl.so.7 для пакета: cloog-ppl-0.15.7-1.2.el6.i686
6
```

Рис. 3.1: Установка компилятора gcc

Отключил систему защиты SELinux с помощью команды `setenforce 0`. После этого команда `getenforce` вывела `Permissive` (рис - @fig:002).

```
[rvivanov@rvivanov Рабочий стол]$ su
Пароль:
[root@rvivanov Рабочий стол]# setenforce 0
[root@rvivanov Рабочий стол]# getenforce
Permissive
[root@rvivanov Рабочий стол]#
```

Рис. 3.2: Отключение системы запретов

2. Изучил компиляцию программ. Компилятор языка С 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++` (рис -@fig:003).

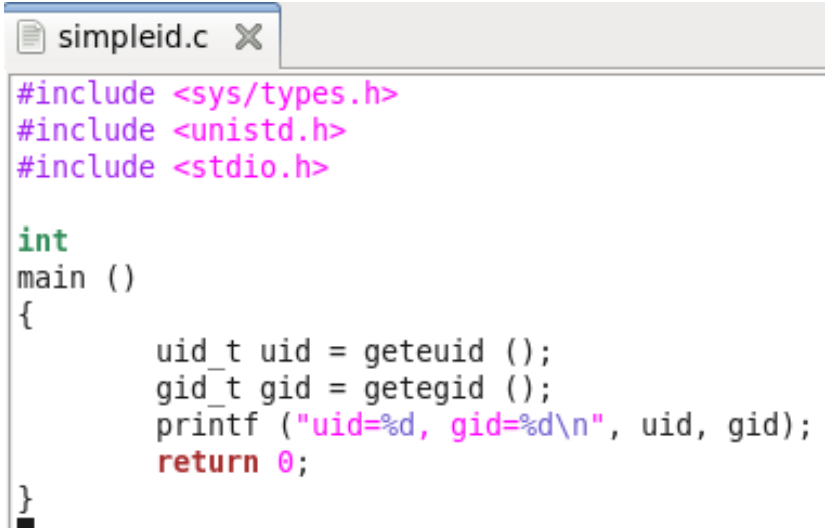
```
[root@rvivanov Рабочий стол]# whereis gcc
gcc: /usr/bin/gcc /usr/lib/gcc /usr/libexec/gcc /usr/share/man/man1/gcc.1.gz
[root@rvivanov Рабочий стол]# whereis g++
g++:
[root@rvivanov Рабочий стол]#
```

Рис. 3.3: Проверка названий компиляторов

3. Вошел в систему от имени пользователя `guest` и создал программу `simpleid.c` (рис -@fig:004 и рис -@fig:005).

```
[guest@rvivanov ~]$ touch simpleid.c
[guest@rvivanov ~]$ ./simpleid.c
```

Рис. 3.4: Создание программы `simpleid.c` (часть 1)



```
simpleid.c X
#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;
}
```

Рис. 3.5: Создание программы `simpleid.c` (часть 2)

Скомпилировал программу и убедился, что файл программы создан с помощью команды `gcc simpleid.c -o simpleid` (рис -@fig:006)

```
[guest@rvivanov ~]$ gcc simpleid.c -o simpleid
[guest@rvivanov ~]$ █
```

Рис. 3.6: Компиляция программы

Выполнил программу `simpleid` (рис -@fig:007)

```
[guest@rvivanov ~]$ ./simpleid
uid=501, gid=501
```

Рис. 3.7: Выполнение созданной программы

Выполнил системную программу `id` (рис -@fig:008)

```
[guest@rvivanov ~]$ id
uid=501(guest) gid=501(guest) группы=501(guest) контекст=unconfined_u:unconfined
_r:unconfined_t:s0-s0:c0.c1023
[guest@rvivanov ~]$ █
```

Рис. 3.8: Выполнение системной программы `id`

Вывод обоих способов совпадает.

Усложнил программу, добавив вывод действительных идентификаторов (рис -@fig:009)

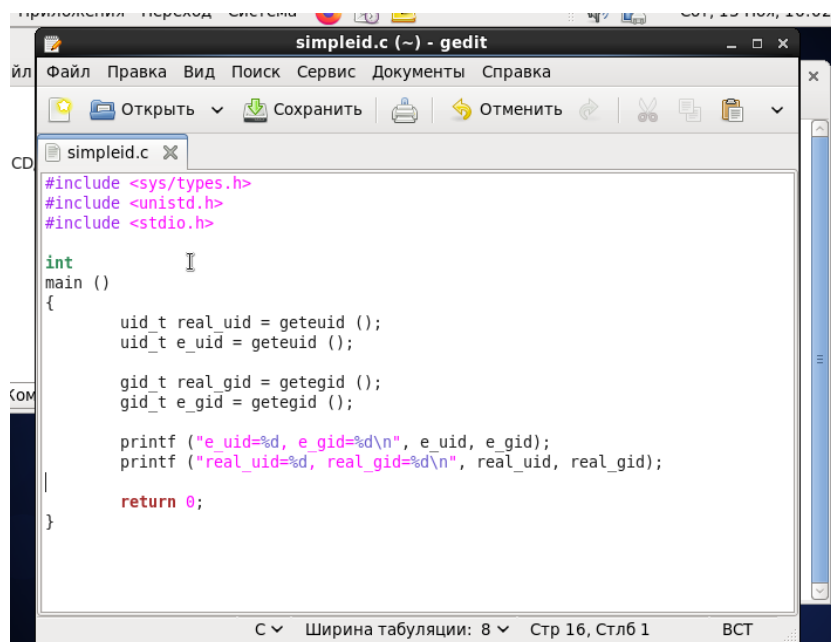


Рис. 3.9: Усложнение программы

Получившуюся программу назвал simpleid2.c (рис -@fig:010)

```

[guest@rvivanov ~]$ mv simpleid.c simpleid2.c
[guest@rvivanov ~]$

```

Рис. 3.10: Переименование программы

Скомпилировал и запустил simpleid2.c (рис -@fig:011)

```

[guest@rvivanov ~]$ gcc simpleid2.c -o simpleid2
[guest@rvivanov ~]$ ./simpleid2
e_uid=501, e_gid=501
real_uid=501, real_gid=501
[guest@rvivanov ~]$

```

Рис. 3.11: Компиляция и запуск файла

От имени суперпользователя выполнил следующие команды (рис -@fig:012)

```
[guest@rvivanov ~]$ su
Пароль:
[root@rvivanov guest]# chown root:guest /home/guest/simpleid2
[root@rvivanov guest]# chmod u+s /home/guest/simpleid2
[root@rvivanov guest]# █
```

Рис. 3.12: Смена владельца и атрибутов от имени суперпользователя

Использовал su для временного повышения своих прав (рис -@fig:013)

```
[guest@rvivanov ~]$ su
Пароль:
```

Рис. 3.13: Использование оператора su

Команда su используется для получения прав суперпользователя.

Выполнил проверку правильности установки новых атрибутов и смены владельца файла simpleid2 (рис -@fig:014)

```
[root@rvivanov guest]# ls -l simpleid2
-rwsrwxr-x. 1 root guest 4971 Ноя 13 16:03 simpleid2
[root@rvivanov guest]# █
```

Рис. 3.14: Проверка правильности установления атрибутов

Запустил simpleid2 и id (рис -@fig:015)

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

Рис. 3.15: Проверка id пользователя и группы

Проделал тоже самое относительно SetGID-бита (рис -@fig:016)

```
[root@rvivanov guest]# chmod g+s /home/guest/simpleid2
[root@rvivanov guest]# ls -l
итого 60
drwxrwxr-x. 2 guest guest 4096 Окт 29 18:04 dir1
-rwxrwxr-x. 1 guest guest 4890 Ноя 13 16:00 simpleid
-rwsrwsr-x. 1 root  guest 4971 Ноя 13 16:03 simpleid2
```

Рис. 3.16: Повторение операций относительно SetGID-бита

Создал программу `readfile.c` (рис -@fig:017 и рис -@fig:018)

```
[guest@rvivanov ~]$ touch readfile.c
[guest@rvivanov ~]$ █
```

Рис. 3.17: Создание программы `readfile.c` (часть 1)

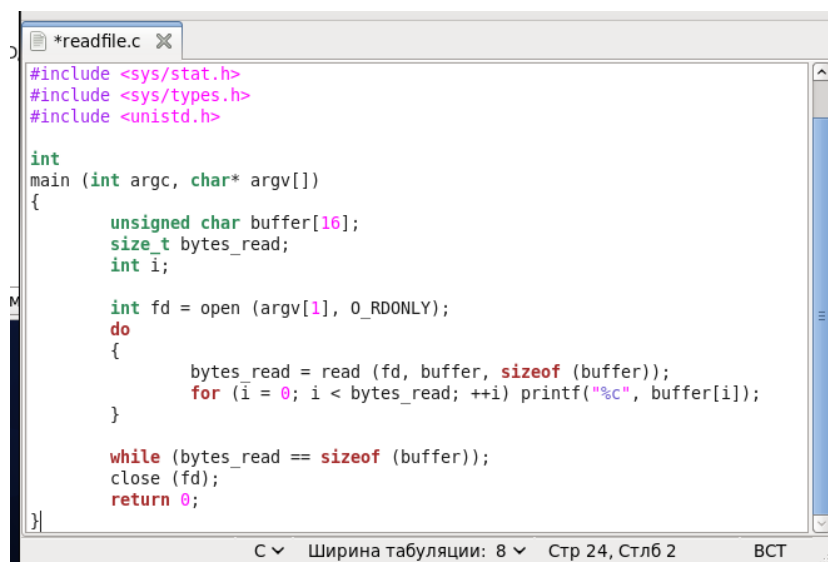


Рис. 3.18: Создание программы `readfile.c` (часть 2)

Откомпилировал созданную программу (рис -@fig:019)

```
[guest@rvivanov ~]$ gcc readfile.c -o readfile
[guest@rvivanov ~]$ █
```

Рис. 3.19: Компиляция программы

Сменил владельца у файла `readfile.c` и изменил права так, чтобы только суперпользователь мог прочитать его, а `guest` не мог (рис -@fig:020)

```
[root@rvivanov guest]# chown root:root /home/guest/readfile.c
[root@rvivanov guest]# chmod 700 readfile.c
[root@rvivanov guest]# █
```

Рис. 3.20: Смена владельца и изменение прав файла

Проверил, что пользователь `guest` не может прочитать файл `readfile.c` (рис -@fig:021)

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

Рис. 3.21: Попытка прочесть файл

Сменил у программы `readfile` владельца и установил SetUID-бит (рис -@fig:022)

```
[root@rvivanov guest]# chown root:root /home/guest/readfile
[root@rvivanov guest]# chmod u+s /home/guest/readfile
[root@rvivanov guest]# █
```

Рис. 3.22: Смена владельца и установка SetUID-бита

Проверил, может ли программа `readfile` прочитать файл `readfile.c`. Да, может. (рис -@fig:023 и рис -@fig:024)

```
Файл  правка  вид  поиск  терминал  справк
[root@rvivanov guest]# ./readfile readfile.c
```

Рис. 3.23: Проверка чтения файла (часть 1)

```

#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;
}

```

Рис. 3.24: Проверка чтения файла (часть 2)

Проверил, может ли программа `readfile` прочитать файл `/etc/shadow`. Да, может. (рис -@fig:025)

```

[root@rivanov guest]# ./readfile /etc/shadow
root:$6$FDIWHuBNmZejwkLe$/rRdDsviWwu.JbBDr0cViGbjgTzwWexTucRB9fgdirbluGjilILbo0E
svE.f0r0CH/qr9I./a8PSBWCfvht8F.: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::::::
usbmuxd:!!:18888::::::
vcasa:!!:18888::::::
rpc:!!:18888:0:99999:7:::
rtkit:!!:18888::::::

```

Рис. 3.25: 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` (рис -@fig:026)

```
[guest@rvivanov ~]$ ls -l / | grep tmp
drwxrwxrwt. 30 root root 4096 Ноя 13 16:10 tmp
[guest@rvivanov ~]$ █
```

Рис. 3.26: Проверка нахождения атрибута Sticky на директории /tmp

От имени пользователя guest создал файл file01.txt в директории /tmp со словом test (рис -@fig:027):

```
[guest@rvivanov ~]$ echo "test" > /tmp/file01.txt
[guest@rvivanov ~]$ █
```

Рис. 3.27: Создание файла и внесение записи в него

Просмотрел атрибуты у только что созданного файла и разрешил чтение и запись для категории пользователей “все остальные” (рис -@fig:028):

```
[guest@rvivanov ~]$ ls -l /tmp/file01.txt
-rw-rw-r--. 1 guest guest 5 Ноя 13 16:24 /tmp/file01.txt
[guest@rvivanov ~]$ chmod o+rw /tmp/file01.txt
[guest@rvivanov ~]$ ls -l /tmp/file01.txt
-rw-rw-rw-. 1 guest guest 5 Ноя 13 16:24 /tmp/file01.txt
[guest@rvivanov ~]$ █
```

Рис. 3.28: 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 (рис -@fig:029):

```
[guest@rvivanov ~]$ su - guest2
Пароль:
[guest2@rvivanov ~]$ cat /tmp/file01.txt
test
[guest2@rvivanov ~]$ █
```

Рис. 3.29: Чтение файла от имени пользователя guest2

От имени пользователя guest2 дозаписал в файл /tmp/file01.txt слово test2 (рис -@fig:030):

```
[guest2@rvivanov ~]$ echo "test2" >> /tmp/file01.txt
[guest2@rvivanov ~]$ █
```

Рис. 3.30: Дозапись слова в файл от имени пользователя guest2

Проверил содержимое файла (рис -@fig:031):

```
[guest2@rvivanov ~]$ cat /tmp/file01.txt
test
test2
[guest2@rvivanov ~]$ █
```

Рис. 3.31: Проверка содежимого в файле от имени пользователя guest2

От имени пользователя guest2 записал в файл /tmp/file01.txt слово test3, стерев при этом всю имеющуюся в файле информацию (рис -@fig:032):

```
-----
[guest2@rvivanov ~]$ echo "test3" > /tmp/file01.txt
[guest2@rvivanov ~]$ █
```

Рис. 3.32: Перезапись информации в файл от имени пользователя guest2

Проверил содержимое файла (рис -@fig:033):

```
[guest2@rvivanov ~]$ echo "test3" > /tmp/file01.txt
[guest2@rvivanov ~]$ cat /tmp/file01.txt
test3
[guest2@rvivanov ~]$ █
```

Рис. 3.33: Проверка содежимого в файле от имени пользователя guest2

От имени пользователя guest2 попробовал удалить файл /tmp/file01.txt (рис -@fig:034):

```
[guest2@rvivanov ~]$ rm /tmp/file01.txt
rm: невозможно удалить «/tmp/file01.txt»: Операция не допускается
[guest2@rvivanov ~]$ █
```

Рис. 3.34: Попытка удаления файла от имени пользователя guest2

Мне не удалось удалить файл.

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

```
[guest2@rvivanov ~]$ su -
Пароль:
[root@rvivanov ~]# chmod -t /tmp
[root@rvivanov ~]# █
```

Рис. 3.35: Повышение прав до суперпользователя. Снятие атрибута `t`

Покинул режим суперпользователя командой `exit` (рис -@fig:036):

```
[root@rvivanov ~]# exit
logout
```

Рис. 3.36: Выход из режима суперпользователя

От имени пользователя `guest2` проверил, что атрибута `t` у директории `/tmp` нет (рис -@fig:037):

```
[guest2@rvivanov ~]$ ls -l / | grep tmp
drwxrwxrwx. 30 root root 4096 Ноя 13 16:24 tmp
[guest2@rvivanov ~]$ █
```

Рис. 3.37: Проверка отсутствия атрибута `t`

Повторил предыдущие шаги (рис -@fig:038):

```

[guest2@rvivanov ~]$ echo "test" > /tmp/file01.txt
[guest2@rvivanov ~]$ cat /tmp/file01.txt
test
[guest2@rvivanov ~]$ echo "test2" >> /tmp/file01.txt
[guest2@rvivanov ~]$ cat /tmp/file01.txt
test
test2
[guest2@rvivanov ~]$ echo "test3" > /tmp/file01.txt
[guest2@rvivanov ~]$ cat /tmp/file01.txt
test3
[guest2@rvivanov ~]$ rm /tmp/file01.txt
[guest2@rvivanov ~]$ █

```

Рис. 3.38: 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` (рис -@fig:039):

```

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

```

Рис. 3.39: 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