

# Лабораторная работа №5

Дискреционное разграничение прав Linux. Исследования влияния расширенных атрибутов

---

Гисматуллин А.В.

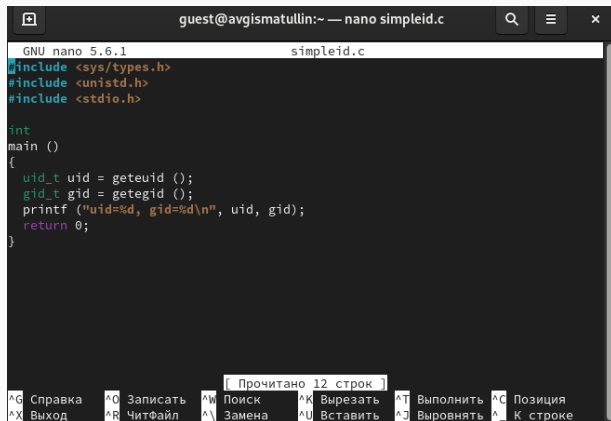
Российский университет дружбы народов, Москва, Россия

- 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

# **Процесс выполнения лабораторной работы**

---

## Файл simpleid.c



```
GNU nano 5.6.1                                simpleid.c
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>

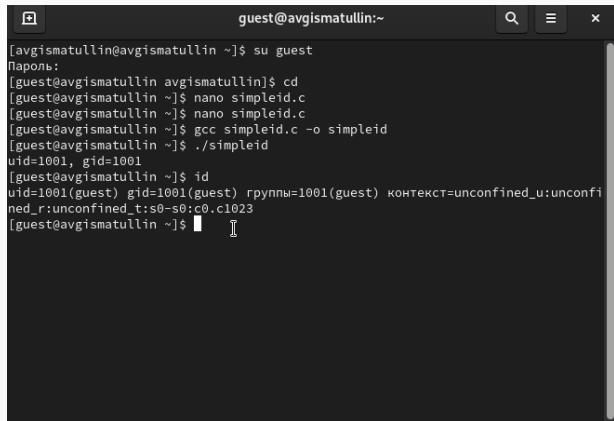
int
main ()
{
    uid_t uid = geteuid ();
    gid_t gid = getegid ();
    printf ("uid=%d, gid=%d\n", uid, gid);
    return 0;
}
```

[ Прочитано 12 строк ]

|                   |                    |                   |                    |                     |                    |
|-------------------|--------------------|-------------------|--------------------|---------------------|--------------------|
| <b>^G</b> Справка | <b>^O</b> Записать | <b>^W</b> Поиск   | <b>^K</b> Вырезать | <b>^T</b> Выполнить | <b>^C</b> Позиция  |
| <b>^X</b> Выход   | <b>^R</b> Читфайл  | <b>^_\</b> Замена | <b>^U</b> Вставить | <b>^J</b> Выровнять | <b>^_</b> К строке |

Рис. 1: Редактор. Файл simpleid.c

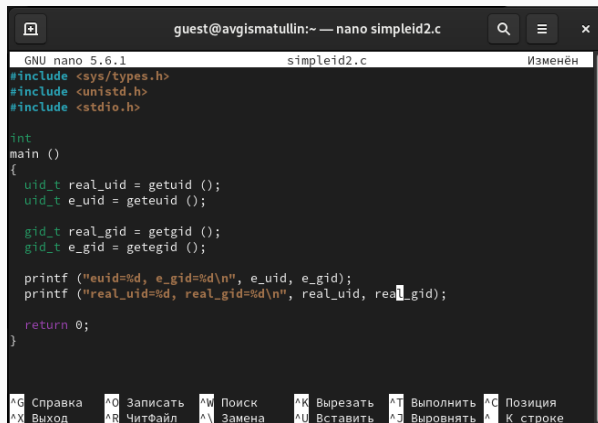
# Сравнение с системной командой



```
guest@avgismatullin:~  
[avgismatullin@avgismatullin ~]$ su guest  
Пароль:  
[guest@avgismatullin avgismatullin]$ cd  
[guest@avgismatullin ~]$ nano simpleid.c  
[guest@avgismatullin ~]$ gcc simpleid.c -o simpleid  
[guest@avgismatullin ~]$ ./simpleid  
uid=1001, gid=1001  
[guest@avgismatullin ~]$ id  
uid=1001(guest) gid=1001(guest) rpyнпы=1001(guest) контекст=unconfined_u:unconfi  
ned_r:unconfined_t:s0-s0:c0.c1023  
[guest@avgismatullin ~]$
```

**Рис. 2:** Командная строка. Сравнение с выводом `id`

# Simpleid2.c



```
GNU nano 5.6.1                                simpleid2.c                                Изменён
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>

int
main ()
{
    uid_t real_uid = getuid ();
    uid_t e_uid = geteuid ();

    gid_t real_gid = getgid ();
    gid_t e_gid = getegid ();

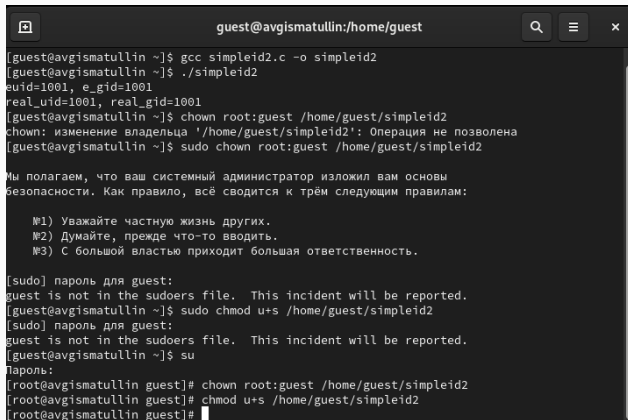
    printf ("euid=%d, e_gid=%d\n", e_uid, e_gid);
    printf ("real_uid=%d, real_gid=%d\n", real_uid, real_gid);

    return 0;
}

^G Справка      ^O Записать    ^W Поиск      ^K Вырезать   ^T Выполнить  ^C Позиция
^X Выход        ^R ЧитФайл    ^\ Замена     ^U Вставить   ^J Выровнять  ^_ К строке
```

Рис. 3: Редактор. Новая программа

# Изменение владельца и атрибутов



```
guest@avgismatullin:/home/guest

[guest@avgismatullin ~]$ gcc simpleid2.c -o simpleid2
[guest@avgismatullin ~]$ ./simpleid2
euid=1001, e_gid=1001
real_uid=1001, real_gid=1001
[guest@avgismatullin ~]$ chown root:guest /home/guest/simpleid2
chown: изменение владельца '/home/guest/simpleid2': Операция не позволена
[guest@avgismatullin ~]$ sudo chown root:guest /home/guest/simpleid2

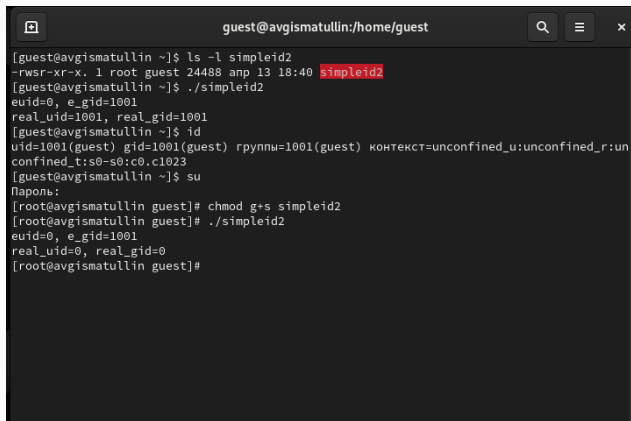
Мы полагаем, что ваш системный администратор изложил вам основы
безопасности. Как правило, всё сводится к трём следующим правилам:

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

[sudo] пароль для guest:
guest is not in the sudoers file. This incident will be reported.
[guest@avgismatullin ~]$ sudo chmod u+s /home/guest/simpleid2
[sudo] пароль для guest:
guest is not in the sudoers file. This incident will be reported.
[guest@avgismatullin ~]$ su
Пароль:
[root@avgismatullin guest]# chown root:guest /home/guest/simpleid2
[root@avgismatullin guest]# chmod u+s /home/guest/simpleid2
[root@avgismatullin guest]#
```

Рис. 4: Командная строка. Изменение владельца и атрибутов файла

# Сравнение вывода



```
guest@avgismatullin:/home/guest

[guest@avgismatullin ~]$ ls -l simpleid2
-rwsr-xr-x. 1 root guest 24488 anp 13 18:40 simpleid2
[guest@avgismatullin ~]$ ./simpleid2
euid=0, e_gid=1001
real_uid=1001, real_gid=1001
[guest@avgismatullin ~]$ id
uid=1001(guest) gid=1001(guest) rгруппы=1001(guest) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@avgismatullin ~]$ su
Пароль:
[root@avgismatullin guest]# chmod g+s simpleid2
[root@avgismatullin guest]# ./simpleid2
euid=0, e_gid=1001
real_uid=0, real_gid=0
[root@avgismatullin guest]#
```

Рис. 5: Командная строка. Сравнение результатов вывода





```
guest@avgismatullin:~ — nano readfile.c
GNU nano 5.6.1                                readfile.c
#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 (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. 6:** Редактор. Файл readfile

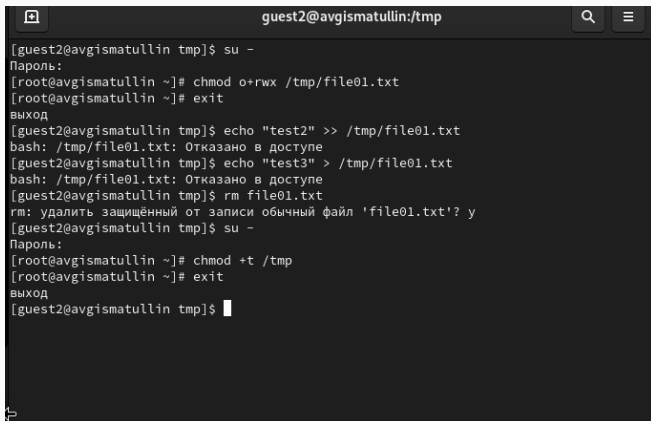
A terminal window titled 'guest@avgismatullin:~' with a search icon in the top right. The terminal shows a sequence of commands and their output. First, a root shell is spawned, and the user 'guest' is prompted. The user runs 'chown root:root readfile', 'chmod o-r readfile.c', 'chmod g-rw readfile.c', and 'chmod u+s readfile'. Then the user types 'exit' and is prompted again. Finally, the user runs 'cat readfile.c', displaying the contents of the file. The code includes standard headers and defines a main function with a buffer and file handle.

```
guest@avgismatullin:~  
[root@avgismatullin guest]# chown root:root readfile  
[root@avgismatullin guest]# chmod o-r readfile.c  
[root@avgismatullin guest]# chmod g-rw readfile.c  
[root@avgismatullin guest]# chmod u+s readfile  
[root@avgismatullin guest]# exit  
exit  
[guest@avgismatullin ~]$ cat 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);
```

**Рис. 7:** Командная строка. Проверка изменений

```
[guest@avgismatullin ~]$ ls -l / | grep tmp
drwxrwxrwt. 15 root root 4096 anp 13 19:02 tmp
[guest@avgismatullin ~]$ echo "test" > /tmp/file01.txt
[guest@avgismatullin ~]$ ls -l /tmp/file01.txt
-rw-r--r--. 1 guest guest 5 anp 13 19:06 /tmp/file01.txt
[guest@avgismatullin ~]$ chmod o_rw /tmp/file01.txt
chmod: неверный режим: «o_rw»
По команде «chmod --help» можно получить дополнительную информацию.
[guest@avgismatullin ~]$ chmod o+rw /tmp/file01.txt
[guest@avgismatullin ~]$ ls -l /tmp/file01.txt
-rw-r--rw-. 1 guest guest 5 anp 13 19:06 /tmp/file01.txt
[guest@avgismatullin ~]$ su guest2
Пароль:
[guest2@avgismatullin guest]$ echo "test2" > /tmp/file01.txt
bash: /tmp/file01.txt: Отказано в доступе
[guest2@avgismatullin guest]$ cat /tmp/file01.txt
test
[guest2@avgismatullin guest]$ echo "test3" > /tmp/file01.txt
bash: /tmp/file01.txt: Отказано в доступе
[guest2@avgismatullin guest]$ echo "test2" >> /tmp/file01.txt
bash: /tmp/file01.txt: Отказано в доступе
[guest2@avgismatullin guest]$ cat /tmp/file01.txt
test
[guest2@avgismatullin guest]$
```

Рис. 8: Командная строка. Исследование Stiky-бита



```
guest2@avgismatullin:/tmp
[guest2@avgismatullin tmp]$ su -
Пароль:
[root@avgismatullin ~]# chmod o+rw /tmp/file01.txt
[root@avgismatullin ~]# exit
выход
[guest2@avgismatullin tmp]$ echo "test2" >> /tmp/file01.txt
bash: /tmp/file01.txt: Отказано в доступе
[guest2@avgismatullin tmp]$ echo "test3" > /tmp/file01.txt
bash: /tmp/file01.txt: Отказано в доступе
[guest2@avgismatullin tmp]$ rm file01.txt
rm: удалить защищённый от записи обычный файл 'file01.txt'? y
[guest2@avgismatullin tmp]$ su -
Пароль:
[root@avgismatullin ~]# chmod +t /tmp
[root@avgismatullin ~]# exit
выход
[guest2@avgismatullin tmp]$
```

**Рис. 9:** Командная строка. Исследование Stiky-бита

## **Выводы по проделанной работе**

---

## Выводы по проделанной работе

В ходе выполнения данной лабораторной работы были изучены механизмы изменения идентификаторов, применения SetUID- и Sticky-битов. Получены практические навыки работы в консоли с дополнительными атрибутами. Рассмотрены работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов