

Отчёт по лабораторной работе №5

дисциплина: Информационная безопасность

Купатенко Владислав Георгиевич

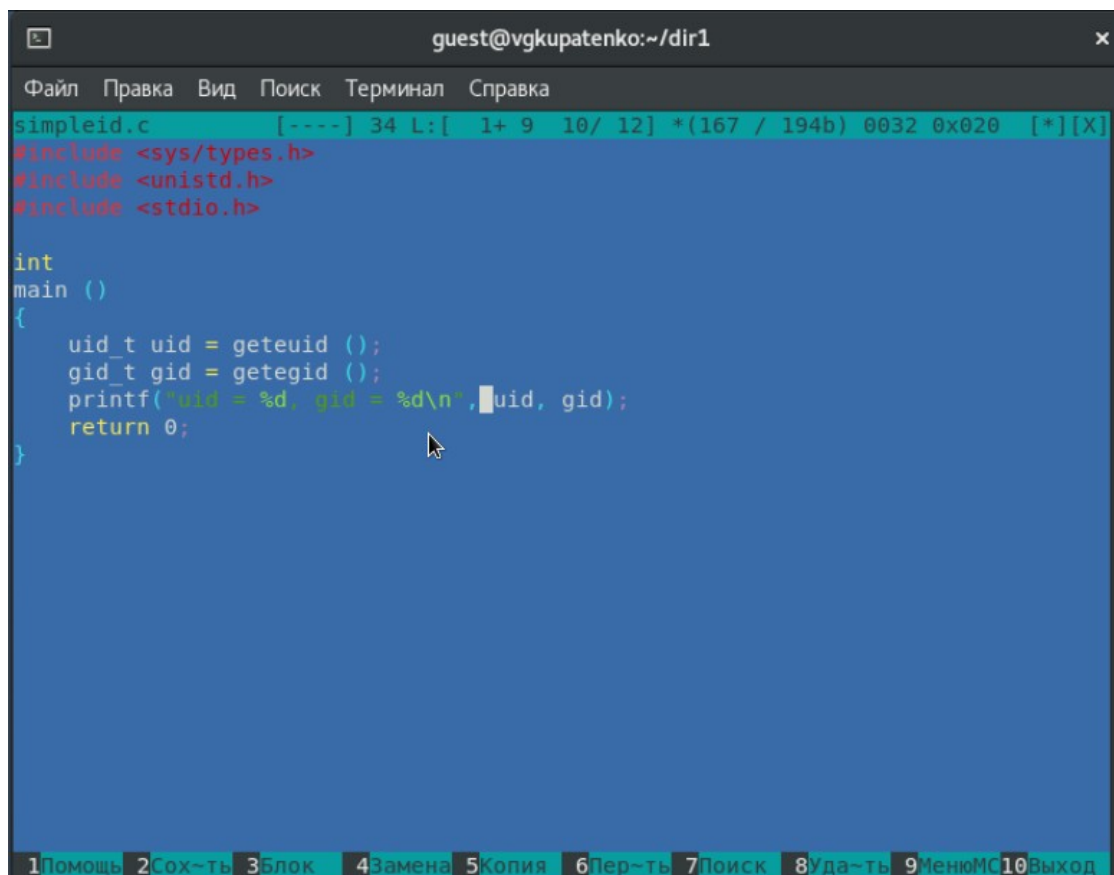
Table of Contents

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

Создан код simpleid.c (рис.1).



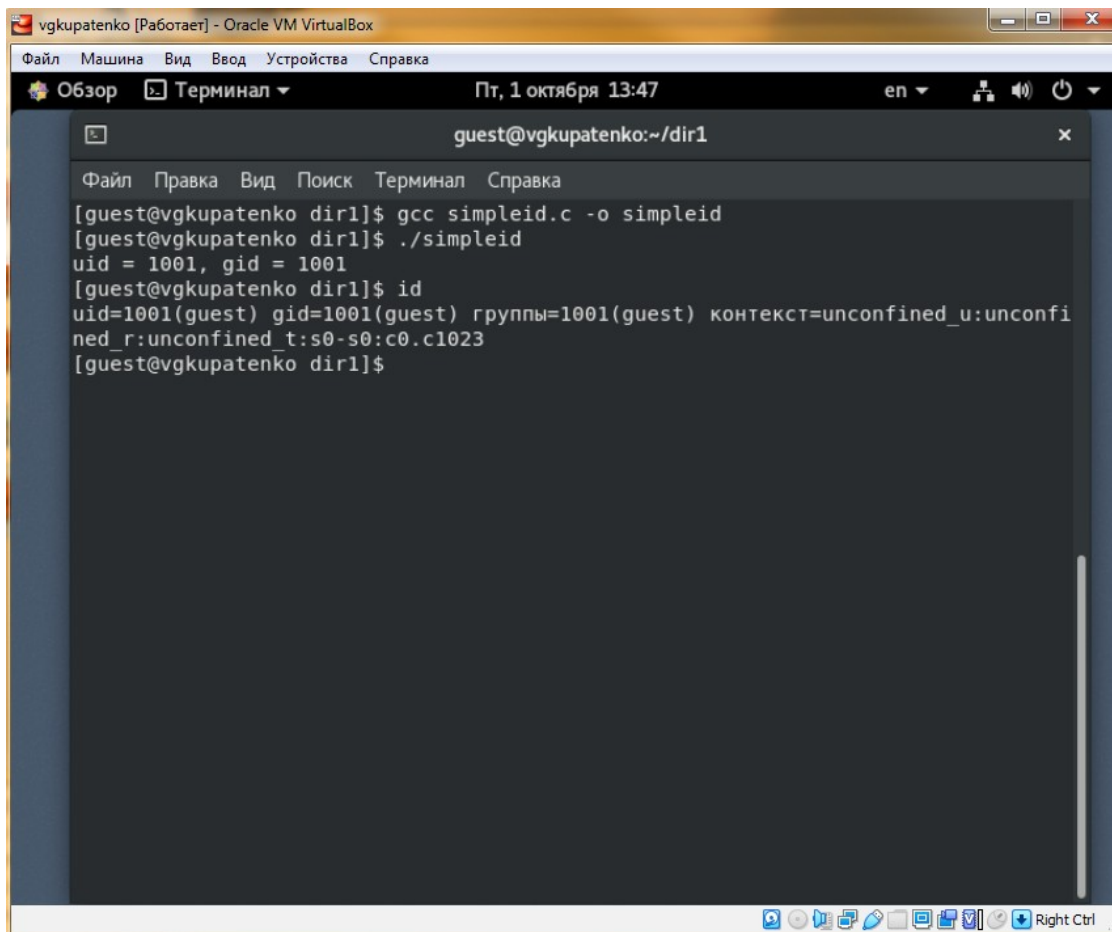
```
guest@vgkupatenko:~/dir1
Файл  Правка  Вид  Поиск  Терминал  Справка
simpleid.c  [----] 34 L: [ 1+ 9 10/ 12] *(167 / 194b) 0032 0x020 [*][X]
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>

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

1Помощь 2Сох-ть 3Блок 4Замена 5Копия 6Пер-ть 7Поиск 8Уда-ть 9МенюМС10Выход

Рисунок №1: simpleid.c

Проведена компиляция кода и сравнение с id (рис.2).



The image shows a screenshot of a VirtualBox window titled "vgkupatenko [Работает] - Oracle VM VirtualBox". The main menu bar includes "Файл", "Машина", "Вид", "Ввод", "Устройства", and "Справка". Below this is a toolbar with icons for "Обзор", "Терминал", and a dropdown menu. The date and time "Пт, 1 октября 13:47" are displayed, along with language settings "en" and system icons for network, volume, and power. The terminal window itself has a title bar "guest@vgkupatenko:~/dir1" and a menu bar "Файл", "Правка", "Вид", "Поиск", "Терминал", "Справка". The terminal output shows the following commands and results:

```
guest@vgkupatenko:~/dir1$ gcc simpleid.c -o simpleid
guest@vgkupatenko:~/dir1$ ./simpleid
uid = 1001, gid = 1001
guest@vgkupatenko:~/dir1$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfi
ned_r:unconfined_t:s0-s0:c0.c1023
guest@vgkupatenko:~/dir1$
```

Рисунок №2: Компиляция *simpleid.c*
Модифицирован код *simpleid* (рис.3).

```
guest@vgkupatenko:~/dir1
Файл  Правка  Вид  Поиск  Терминал  Справка
simpleid2.c  [----]  0 L:[ 1+15 16/ 18] *(331 / 347b) 0010 0x00A [*][X]
#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 ();

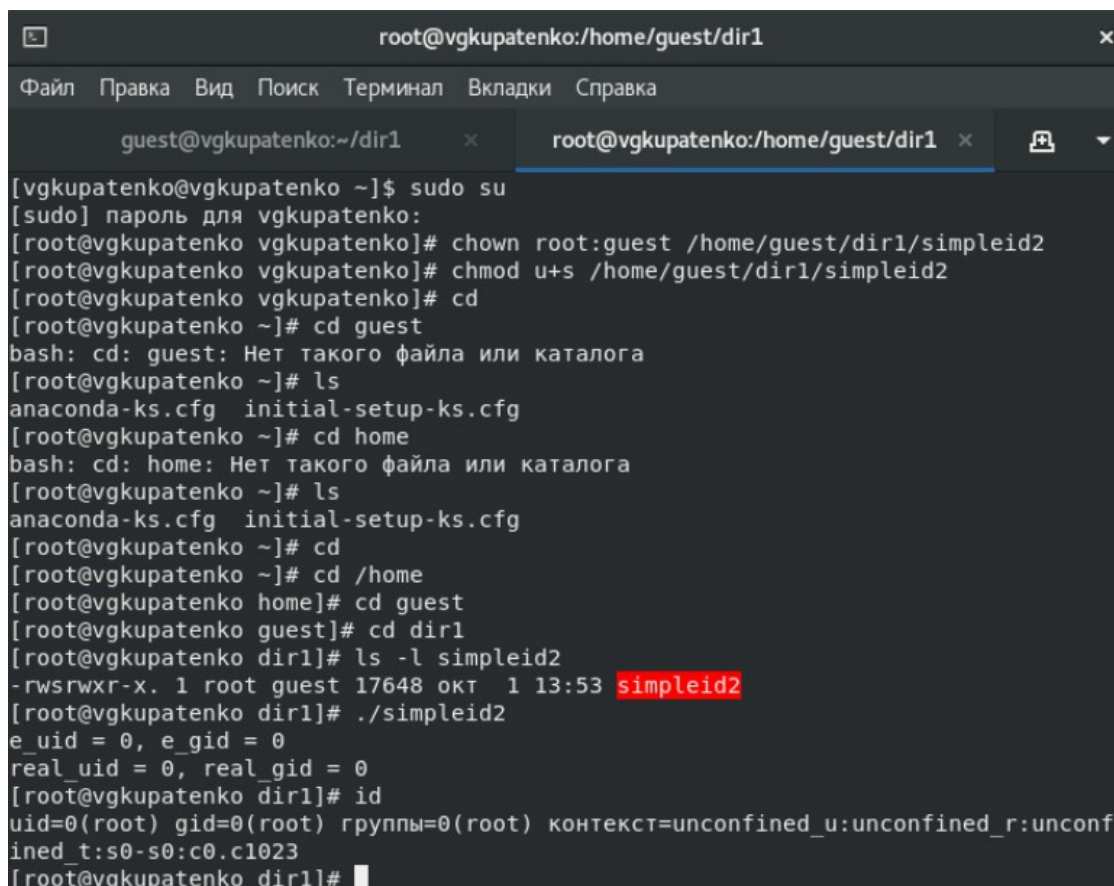
    ....
    printf("e_uid = %d, e_gid = %d\n", e_uid, e_gid);
    printf("real_uid = %d, real_gid = %d\n", real_uid, real_gid);

    return 0;
}

1Помощь 2Сохранить 3Блок 4Замена 5Копия 6Перейти 7Поиск 8Удалить 9МенюМС10Выход
```

Рисунок №3: *simpleid2.c*

Выполнены команды с правами от имени суперпользователя. (рис. 4).

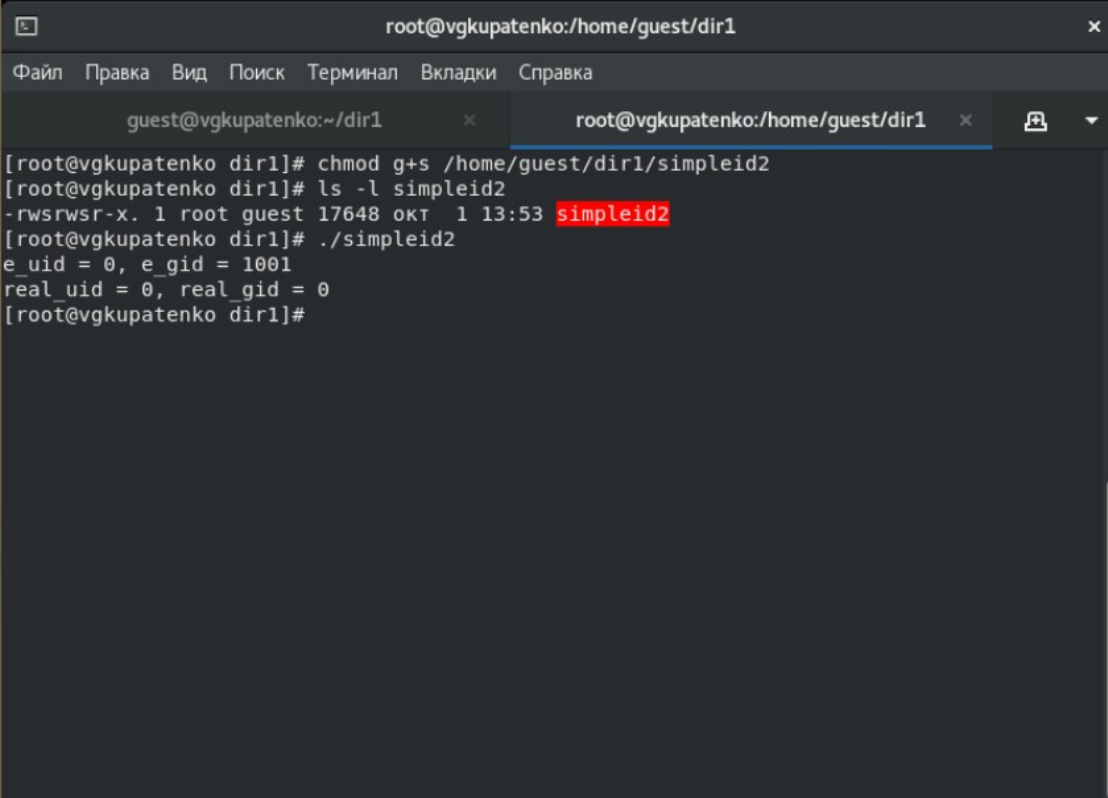


The image shows a terminal window with a dark background and light text. The title bar at the top reads 'root@vgkupatenko:/home/guest/dir1'. Below the title bar is a menu bar with options: 'Файл', 'Правка', 'Вид', 'Поиск', 'Терминал', 'Вкладки', and 'Справка'. There are two tabs open: 'guest@vgkupatenko:~/dir1' and 'root@vgkupatenko:/home/guest/dir1'. The active tab is 'root@vgkupatenko:/home/guest/dir1'. The terminal content shows a series of commands and their outputs, demonstrating a privilege escalation process. The user starts as 'vgkupatenko', uses 'sudo su' to become root, then changes permissions on '/home/guest/dir1/simpleid2' to 'u+s'. After several 'cd' attempts that fail due to non-existent paths, the user successfully navigates to '/home/guest/dir1' and then to 'simpleid2'. The final output shows the user is now root (uid=0, gid=0) with a context of 'unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023'.

```
[vgkupatenko@vgkupatenko ~]$ sudo su
[sudo] пароль для vgkupatenko:
[root@vgkupatenko vgkupatenko]# chown root:guest /home/guest/dir1/simpleid2
[root@vgkupatenko vgkupatenko]# chmod u+s /home/guest/dir1/simpleid2
[root@vgkupatenko vgkupatenko]# cd
[root@vgkupatenko ~]# cd guest
bash: cd: guest: Нет такого файла или каталога
[root@vgkupatenko ~]# ls
anaconda-ks.cfg  initial-setup-ks.cfg
[root@vgkupatenko ~]# cd home
bash: cd: home: Нет такого файла или каталога
[root@vgkupatenko ~]# ls
anaconda-ks.cfg  initial-setup-ks.cfg
[root@vgkupatenko ~]# cd
[root@vgkupatenko ~]# cd /home
[root@vgkupatenko home]# cd guest
[root@vgkupatenko guest]# cd dir1
[root@vgkupatenko dir1]# ls -l simpleid2
-rwsrwxr-x. 1 root guest 17648 окт 1 13:53 simpleid2
[root@vgkupatenko dir1]# ./simpleid2
e_uid = 0, e_gid = 0
real_uid = 0, real_gid = 0
[root@vgkupatenko dir1]# id
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@vgkupatenko dir1]#
```

Рисунок №4: Опыты с правами

Повторение операции относительно GID бита (рис. 5).



A terminal window titled 'root@vgkupatenko:/home/guest/dir1'. The window has a menu bar with 'Файл', 'Правка', 'Вид', 'Поиск', 'Терминал', 'Вкладки', and 'Справка'. Below the menu bar, there are two tabs: 'guest@vgkupatenko:~/dir1' and 'root@vgkupatenko:/home/guest/dir1'. The active tab is 'root@vgkupatenko:/home/guest/dir1'. The terminal content shows the following commands and output:

```
[root@vgkupatenko dir1]# chmod g+s /home/guest/dir1/simpleid2
[root@vgkupatenko dir1]# ls -l simpleid2
-rwsrwsr-x. 1 root guest 17648 окт  1 13:53 simpleid2
[root@vgkupatenko dir1]# ./simpleid2
e_uid = 0, e_gid = 1001
real_uid = 0, real_gid = 0
[root@vgkupatenko dir1]#
```

Рисунок №5: Повторение для GID-bit

Создан код readfile.c (рис. 6).

```
guest@vgkupatenko:~/dir1
Файл  Правка  Вид  Поиск  Терминал  Вкладки  Справка
guest@vgkupatenko:~/dir1  x  root@vgkupatenko:/home/guest/dir1  x  [?]  ▾
readfile.c  [-M--]  1  L:[  1+23  24/ 24]  *(454 / 454b)  <EOF>  [*][X]
#include <fcntl.h>
#include <stdio.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>

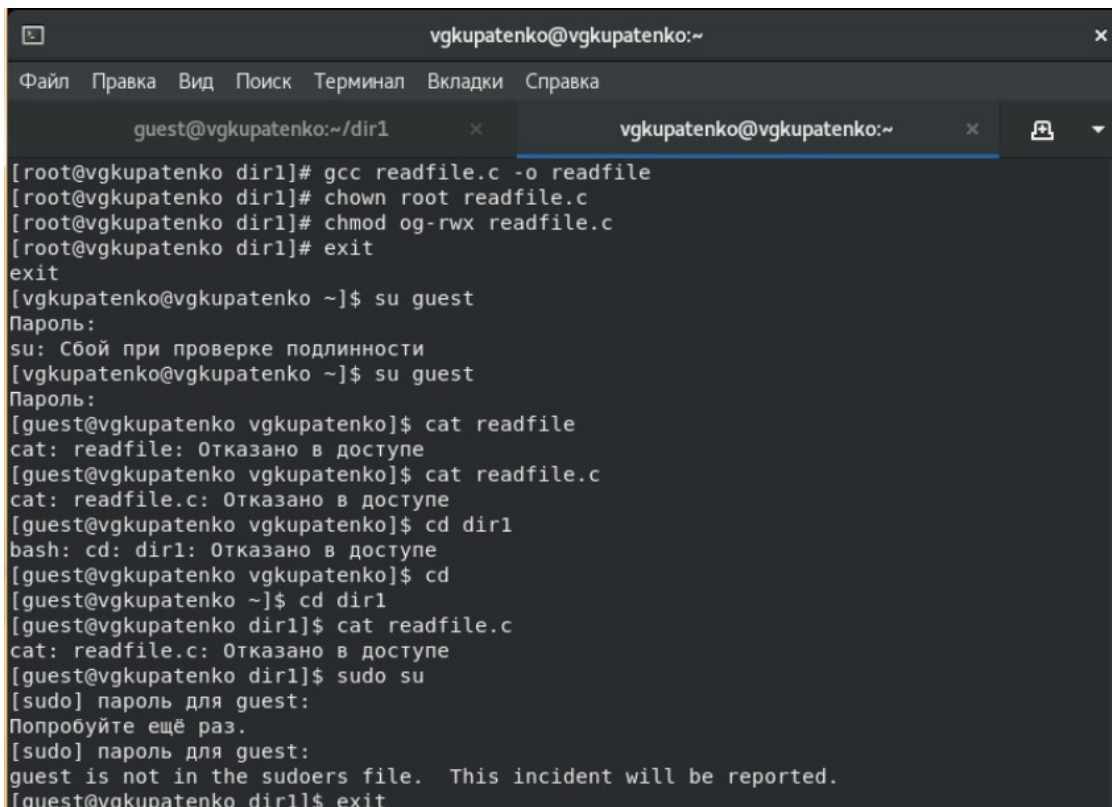
int
main (int argc, char* argv[])
{
    unsigned char buffer[16];
    size_t bytes_read;
    int i;

    ....
    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;
}
```

Рисунок №6: readfile.c

Компиляция кода readfile и работа с правами (рис. 7).

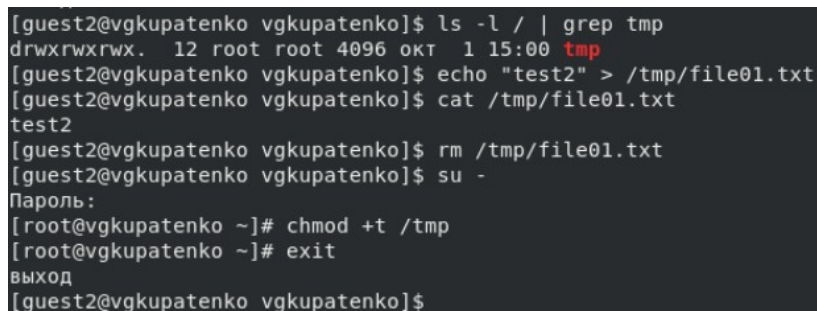


```
vgkupatenko@vgkupatenko:~  
Файл Правка Вид Поиск Терминал Вкладки Справка  
guest@vgkupatenko:~/dir1 x vgkupatenko@vgkupatenko:~ x  
[root@vgkupatenko dir1]# gcc readfile.c -o readfile  
[root@vgkupatenko dir1]# chown root readfile.c  
[root@vgkupatenko dir1]# chmod og-rwx readfile.c  
[root@vgkupatenko dir1]# exit  
exit  
[vgkupatenko@vgkupatenko ~]$ su guest  
Пароль:  
su: Сбой при проверке подлинности  
[vgkupatenko@vgkupatenko ~]$ su guest  
Пароль:  
[guest@vgkupatenko vgkupatenko]$ cat readfile  
cat: readfile: Отказано в доступе  
[guest@vgkupatenko vgkupatenko]$ cat readfile.c  
cat: readfile.c: Отказано в доступе  
[guest@vgkupatenko vgkupatenko]$ cd dir1  
bash: cd: dir1: Отказано в доступе  
[guest@vgkupatenko vgkupatenko]$ cd  
[guest@vgkupatenko ~]$ cd dir1  
[guest@vgkupatenko dir1]$ cat readfile.c  
cat: readfile.c: Отказано в доступе  
[guest@vgkupatenko dir1]$ sudo su  
[sudo] пароль для guest:  
Попробуйте ещё раз.  
[sudo] пароль для guest:  
guest is not in the sudoers file. This incident will be reported.  
[guest@vgkupatenko dir1]$ exit
```

Рисунок №7: Компиляция *readfile.c*

Сменил владельца у файла *readfile.c* и измените права так, чтобы только суперпользователь (root) мог прочитать его, а guest не мог. Проверил, что пользователь guest не может прочитать файл *readfile.c*. Сменил у программы *readfile* владельца и установил SetU'D-бит. Проверил, может ли программа *readfile* прочитать файл *readfile.c* (рис. 8).

От пользователя guest2 попробовал прочитать файл /tmp/file01.txt. От пользователя guest2 попробовал дозаписать в файл /tmp/file01.txt слово test2. Удалось выполнить операцию. Проверил содержимое файла. От пользователя guest2 попробовал записать в файл /tmp/file01.txt слово test3, стерев при этом всю имеющуюся в файле информацию. Удалось выполнить операцию. Проверил содержимое файла. От пользователя guest2 попробовал удалить файл /tmp/file01.tx. Не удалось выполнить операцию. Повысил свои права до суперпользователя и выполнил после этого команду, снимающую атрибут t (Sticky-бит) с директории /tmp. Покинул режим суперпользователя. От пользователя guest2 проверил, что атрибута t у директории /tmp нет. Повторил предыдущие шаги. Удалось успешно выполнить каждый шаг. Повысил свои права до суперпользователя и вернул атрибут t на директорию /tmp (рис. 10).



```
[guest2@vgkupatenko vgkupatenko]$ ls -l / | grep tmp
drwxrwxrwx. 12 root root 4096 окт  1 15:00 tmp
[guest2@vgkupatenko vgkupatenko]$ echo "test2" > /tmp/file01.txt
[guest2@vgkupatenko vgkupatenko]$ cat /tmp/file01.txt
test2
[guest2@vgkupatenko vgkupatenko]$ rm /tmp/file01.txt
[guest2@vgkupatenko vgkupatenko]$ su -
Пароль:
[root@vgkupatenko ~]# chmod +t /tmp
[root@vgkupatenko ~]# exit
выход
[guest2@vgkupatenko vgkupatenko]$
```

Рисунок №10: Возвращение атрибута t

Выводы

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.