

# **Отчёт по лабораторной работе № 5**

**Информационная безопасность**

Кинсиклунон Доря Флора

# Содержание

0.1	Цель работы . . . . .	4
0.2	Теоретическое введение . . . . .	4
0.3	Выполнение лабораторной работы . . . . .	5
0.4	5.1 Создание программы . . . . .	5
0.5	3.2 Исследование Sticky-бита . . . . .	11
0.6	Выводы . . . . .	14

## Список иллюстраций

1	Рис. 5.1: Предварительная подготовка . . . . .	5
2	Рис. 5.2: Команда “whereis” . . . . .	6
3	Рис. 5.3: Вход в систему и создание программы . . . . .	6
4	Рис. 5.4: Код программы simpleid.c . . . . .	7
5	Рис. 5.5: Компиляция и выполнение программы simpleid . . . . .	7
6	Рис. 5.6: Усложнение программы . . . . .	8
7	Рис. 5.7: Переименование программы в simpleid2.c . . . . .	8
8	Рис. 5.8: Компиляция и выполнение программы simpleid2 . . . . .	8
9	Рис. 5.9: Установка новых атрибутов (SetUID) и смена владельца файла . . . . .	9
10	Рис. 5.10: Запуск simpleid2 после установки SetUID . . . . .	9
11	Рис. 5.10: Запуск simpleid2 после установки SetUID . . . . .	10
12	Рис. 5.12: Код программы readfile.c . . . . .	10
13	Рис. 5.13: Смена владельца и прав доступа у файла readfile.c . . . . .	11
14	Рис. 5.14: Запуск программы readfile . . . . .	11
15	Рис. 5.15: Создание файла file01.txt . . . . .	12
16	Рис. 5.16: Попытка выполнить действия над файлом file01.txt от имени пользователя guest2 . . . . .	13
17	Рис. 5.17: Удаление атрибута t (Sticky-бита) и повторение действий	13
18	Рис. 5.18: Возвращение атрибута t (Sticky-бита) . . . . .	14

# Список таблиц

## 0.1 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.

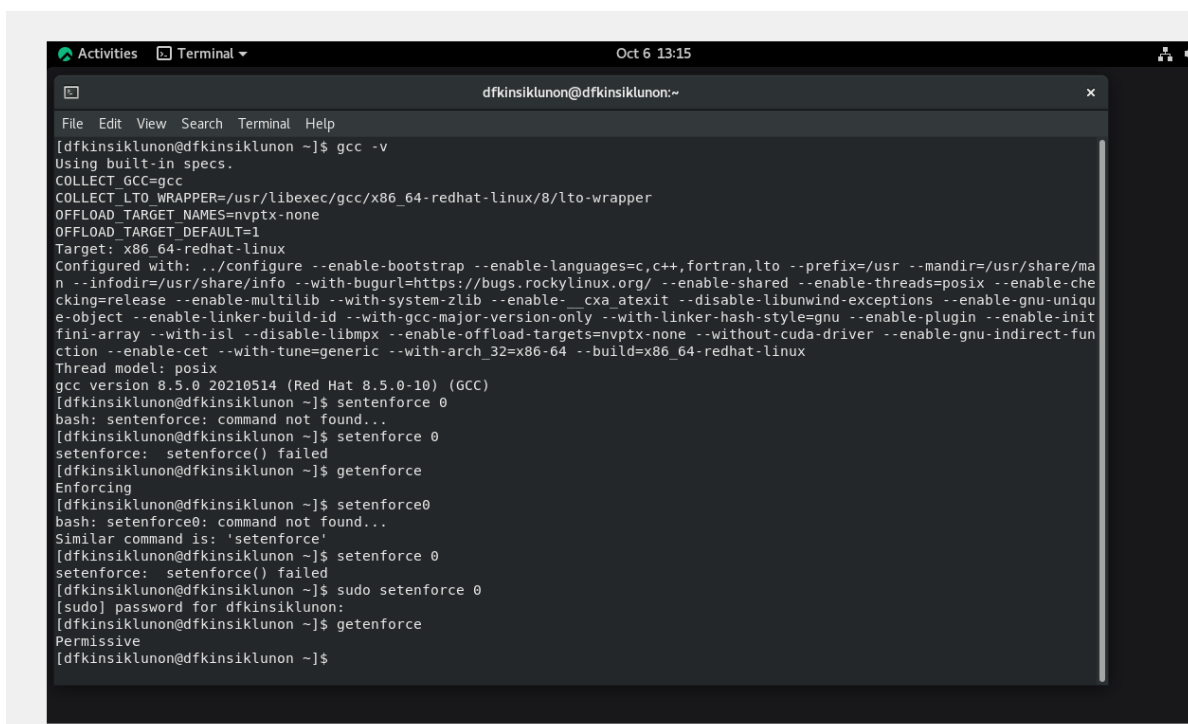
## 0.2 Теоретическое введение

SetUID, SetGID и Sticky - это специальные типы разрешений позволяют задавать расширенные права доступа на файлы или каталоги. • SetUID (set user ID upon execution — «установка ID пользователя во время выполнения) являются флагами прав доступа в Unix, которые разрешают пользователям запускать исполняемые файлы с правами владельца исполняемого файла. • SetGID (set group ID upon execution — «установка ID группы во время выполнения») являются флагами прав доступа в Unix, которые разрешают пользователям запускать исполняемые файлы с правами группы исполняемого файла. • Sticky bit в основном используется в общих каталогах, таких как /var или /tmp, поскольку пользователи могут создавать файлы, читать и выполнять их, принадлежащие другим пользователям, но не могут удалять файлы, принадлежащие другим пользователям.

## 0.3 Выполнение лабораторной работы

### 0.4 5.1 Создание программы


Для начала я убедилась, что компилятор gcc установлен, используя команду “gcc -v”. Затем отключила систему запретов до очередной перезагрузки системы командой “sudo setenforce 0”, после чего команда “getenforce” вывела “Permissive” (рис. 5.1).



```
dfkinsiklunon@dfkinsiklunon:~$ gcc -v
Using built-in specs.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/usr/libexec/gcc/x86_64-redhat-linux/8/lto-wrapper
OFFLOAD_TARGET_NAMES=nvptx-none
OFFLOAD_TARGET_DEFAULT=1
Target: x86_64-redhat-linux
Configured with: ../configure --enable-bootstrap --enable-languages=c,c++,fortran,lto --prefix=/usr --mandir=/usr/share/man --infodir=/usr/share/info --with-bugurl=https://bugs.rockylinux.org/ --enable-shared --enable-threads=posix --enable-checking=release --enable-multilib --with-system-zlib --enable-_cxa_atexit --disable-libunwind-exceptions --enable-gnu-uniquify-object --enable-linker-build-id --with-gcc-major-version-only --with-linker-hash-style=gnu --enable-plugin --enable-initfini-array --with-lsl --disable-libmpx --enable-offload-targets=nvptx-none --without-cuda-driver --enable-gnu-indirect-function --enable-cet --with-tune=generic --with-arch_32=x86-64 --build=x86_64-redhat-linux
Thread model: posix
gcc version 8.5.0 20210514 (Red Hat 8.5.0-10) (GCC)
dfkinsiklunon@dfkinsiklunon:~$ setenforce 0
bash: setenforce: command not found...
dfkinsiklunon@dfkinsiklunon:~$ setenforce 0
setenforce: setenforce() failed
dfkinsiklunon@dfkinsiklunon:~$ getenforce
Enforcing
dfkinsiklunon@dfkinsiklunon:~$ setenforce0
bash: setenforce0: command not found...
Similar command is: 'setenforce'
dfkinsiklunon@dfkinsiklunon:~$ setenforce 0
setenforce: setenforce() failed
dfkinsiklunon@dfkinsiklunon:~$ sudo setenforce 0
[sudo] password for dfkinsiklunon:
dfkinsiklunon@dfkinsiklunon:~$ getenforce
Permissive
dfkinsiklunon@dfkinsiklunon:~$
```

Рис. 1: Рис. 5.1: Предварительная подготовка

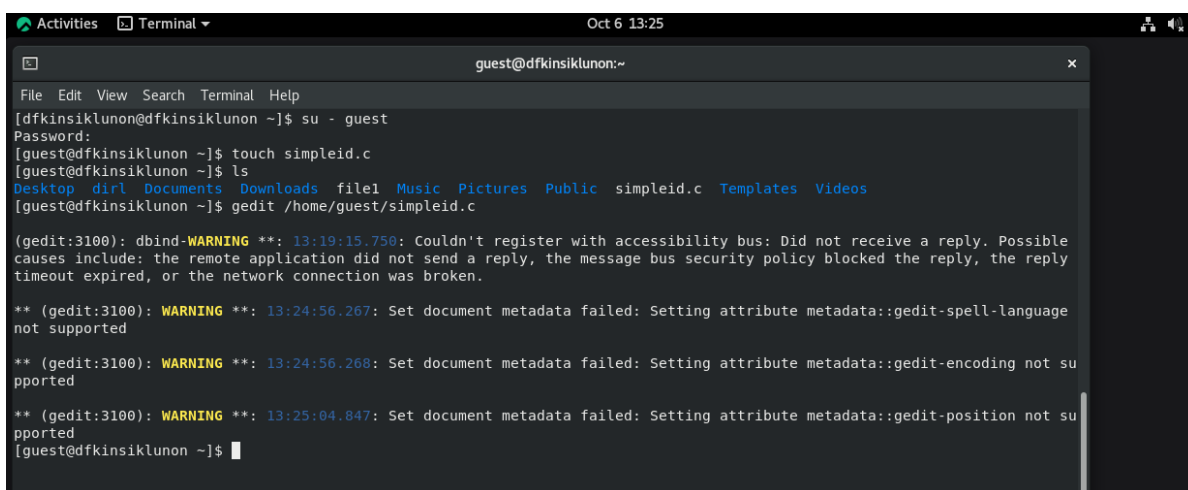
Проверила успешное выполнение команд “whereis gcc” и “whereis g++” (их расположение) (рис. 5.2).



```
dfkinsiklunon@dfkinsiklunon:~$ whereis gcc
gcc: /usr/bin/gcc /usr/lib/gcc /usr/libexec/gcc /usr/share/man/man1/gcc.1.gz /usr/share/info/gcc.info.gz
dfkinsiklunon@dfkinsiklunon:~$ whereis g++
g++: /usr/bin/g++ /usr/share/man/man1/g++.1.gz
dfkinsiklunon@dfkinsiklunon:~$
```

Рис. 2: Рис. 5.2: Команда “whereis”

Вошла в систему от имени пользователя guest командой “su - guest”. Создала программу simpleid.c командой “touch simpleid.c” и открыла её в редакторе командой “gedit /home/guest/simpleid.c” (рис. 5.3).



```
dfkinsiklunon@dfkinsiklunon:~$ su - guest
Password:
[guest@dfkinsiklunon ~]$ touch simpleid.c
[guest@dfkinsiklunon ~]$ ls
Desktop  dirl  Documents  Downloads  file1  Music  Pictures  Public  simpleid.c  Templates  Videos
[guest@dfkinsiklunon ~]$ gedit /home/guest/simpleid.c

(gedit:3100): dbind-WARNING **: 13:19:15.750: Couldn't register with accessibility bus: Did not receive a reply. Possible causes include: the remote application did not send a reply, the message bus security policy blocked the reply, the reply timeout expired, or the network connection was broken.

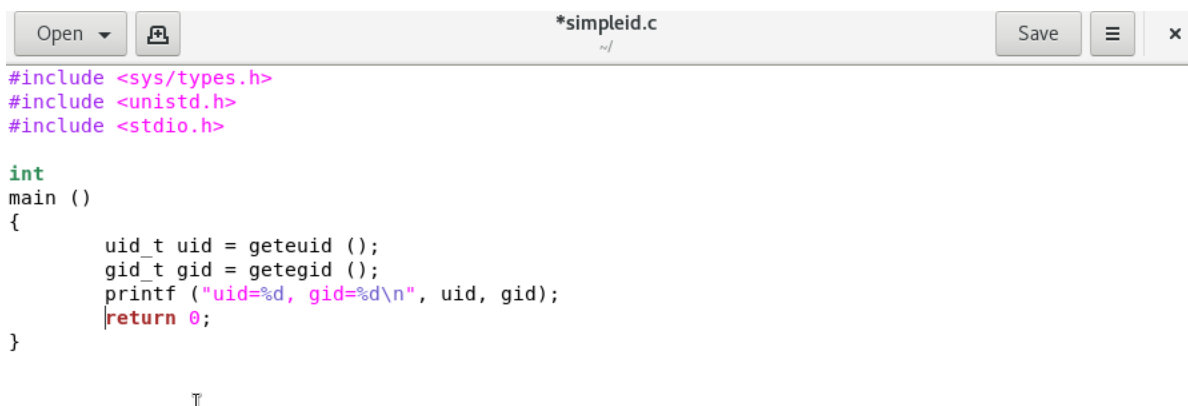
** (gedit:3100): WARNING **: 13:24:56.267: Set document metadata failed: Setting attribute metadata::gedit-spell-language not supported

** (gedit:3100): WARNING **: 13:24:56.268: Set document metadata failed: Setting attribute metadata::gedit-encoding not supported

** (gedit:3100): WARNING **: 13:25:04.847: Set document metadata failed: Setting attribute metadata::gedit-position not supported
[guest@dfkinsiklunon ~]$
```

Рис. 3: Рис. 5.3: Вход в систему и создание программы

Код программы выглядит следующим образом (рис. 5.4).

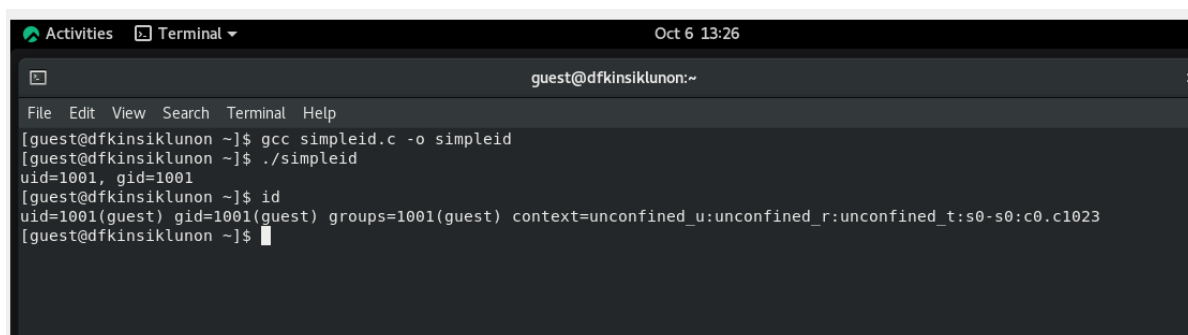


```
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>

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

Рис. 4: Рис. 5.4: Код программы simpleid.c

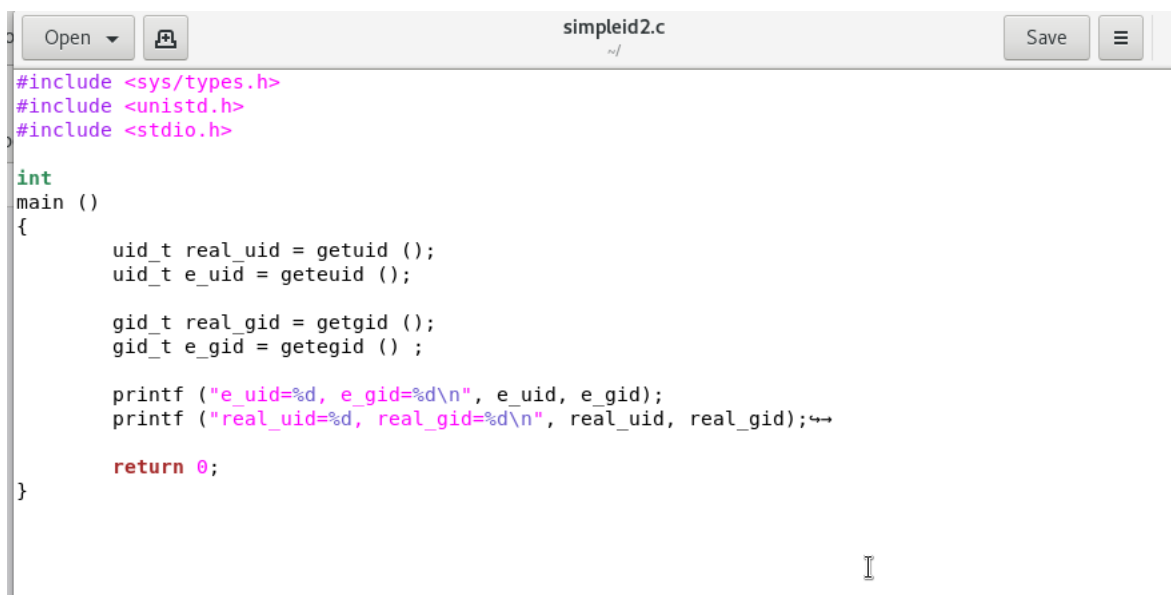
Скомпилировала программу и убедилась, что файл программы был создан командой “gcc simpleid.c -o simpleid”. Выполнила программу simpleid командой “./simpleid”, а затем выполнила системную программу id командой “id”. Результаты, полученные в результате выполнения обеих команд, совпадают (uid=1001 и gid=1001) (рис. 5.5).



```
Oct 6 13:26
guest@dfkinsiklunon:~
File Edit View Search Terminal Help
[guest@dfkinsiklunon ~]$ gcc simpleid.c -o simpleid
[guest@dfkinsiklunon ~]$ ./simpleid
uid=1001, gid=1001
[guest@dfkinsiklunon ~]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@dfkinsiklunon ~]$
```

Рис. 5: Рис. 5.5: Компиляция и выполнение программы simpleid

Усложнила программу, добавив вывод действительных идентификаторов (рис. 5.6).



```
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>

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

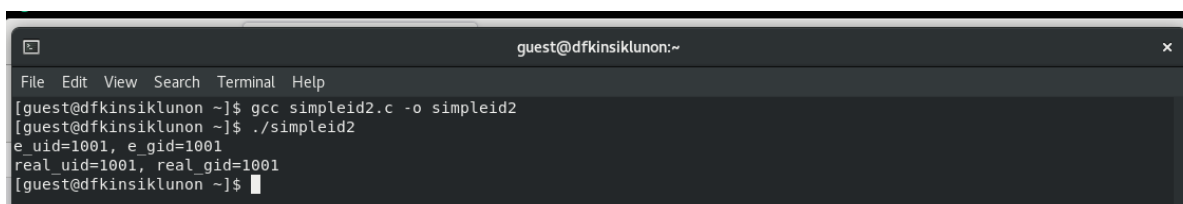
    gid_t real_gid = getgid ();
    gid_t e_gid = getegid ();

    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;
}
```

Рис. 6: Рис. 5.6: Усложнение программы

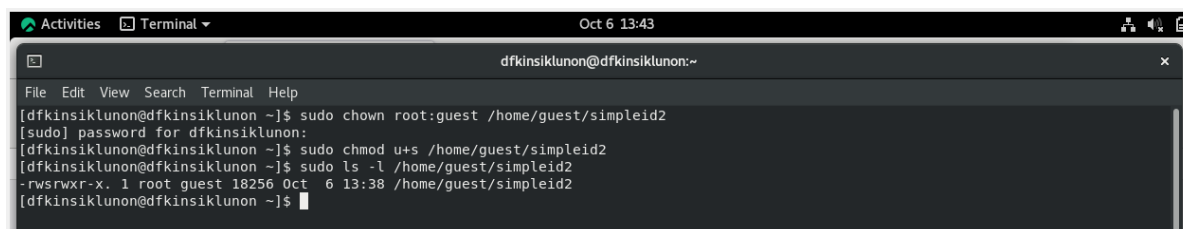
Получившуюся программу назвала simpleid2.c (рис. 3.7).



```
guest@dfkinsiklunon:~
File Edit View Search Terminal Help
[guest@dfkinsiklunon ~]$ gcc simpleid2.c -o simpleid2
[guest@dfkinsiklunon ~]$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
[guest@dfkinsiklunon ~]$
```

Рис. 7: Рис. 5.7: Переименование программы в simpleid2.c

Скомпилировала и запустила simpleid2.c командами “gcc simpleid2.c -o sipleid2” и “./simpleid2” (рис. 3.8).



```
dfkinsiklunon@dfkinsiklunon:~
File Edit View Search Terminal Help
[dfkinsiklunon@dfkinsiklunon ~]$ sudo chown root:guest /home/guest/simpleid2
[sudo] password for dfkinsiklunon:
[dfkinsiklunon@dfkinsiklunon ~]$ sudo chmod u+s /home/guest/simpleid2
[dfkinsiklunon@dfkinsiklunon ~]$ sudo ls -l /home/guest/simpleid2
-rwsrwxr-x. 1 root guest 18256 Oct  6 13:38 /home/guest/simpleid2
[dfkinsiklunon@dfkinsiklunon ~]$
```

Рис. 8: Рис. 5.8: Компиляция и выполнение программы simpleid2



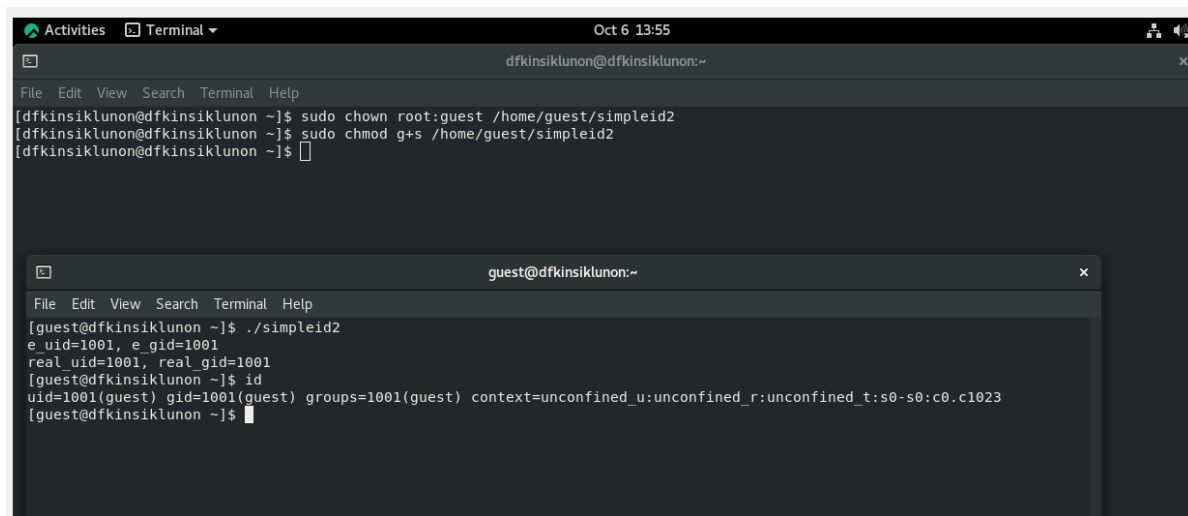
От имени суперпользователя выполнила команды “sudo chown root:guest /home/guest/simpleid2” и “sudo chmod u+s /home/guest/simpleid2”, затем выполнила проверку правильности установки новых атрибутов и смены владельца файла simpleid2 командой “sudo ls -l /home/guest/simpleid2” (рис. 3.9). Этими командами была произведена смена пользователя файла на root и установлен SetUID-бит.



```
guest@dfkinsiklunon:~  
File Edit View Search Terminal Help  
[guest@dfkinsiklunon ~]$ ./simpleid2  
e_uid=0, e_gid=1001  
real_uid=1001, real_gid=1001  
[guest@dfkinsiklunon ~]$ id  
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023  
[guest@dfkinsiklunon ~]$
```

Рис. 9: Рис. 5.9: Установка новых атрибутов (SetUID) и смена владельца файла

Запустила программы simpleid2 и id. Теперь появились различия в uid (рис. 5.10).



```
dfkinsiklunon@dfkinsiklunon:~  
File Edit View Search Terminal Help  
[dfkinsiklunon@dfkinsiklunon ~]$ sudo chown root:guest /home/guest/simpleid2  
[dfkinsiklunon@dfkinsiklunon ~]$ sudo chmod g+s /home/guest/simpleid2  
[dfkinsiklunon@dfkinsiklunon ~]$  
  
guest@dfkinsiklunon:~  
File Edit View Search Terminal Help  
[guest@dfkinsiklunon ~]$ ./simpleid2  
e_uid=1001, e_gid=1001  
real_uid=1001, real_gid=1001  
[guest@dfkinsiklunon ~]$ id  
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023  
[guest@dfkinsiklunon ~]$
```

Рис. 10: Рис. 5.10: Запуск simpleid2 после установки SetUID

Проделала тоже самое относительно SetGID-бита. Также можем заметить различия с предыдущим пунктом (рис. 5.11).

```

-----
#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;
}

```

Рис. 11: Рис. 5.10: Запуск simpleid2 после установки SetUID

Создаем программу readfile.c (рис. 5.12).

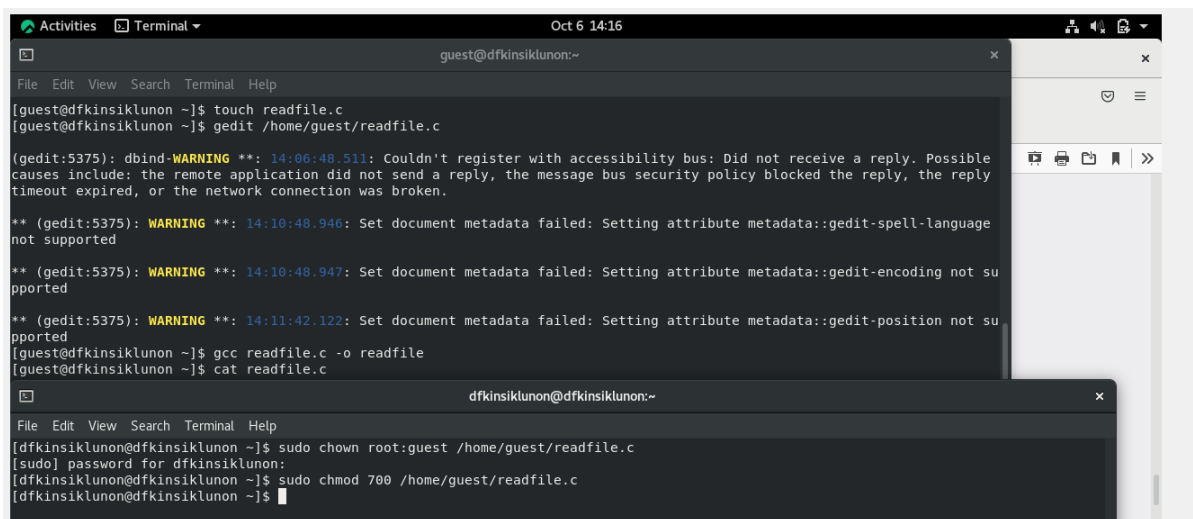


Рис. 12: Рис. 5.12: Код программы readfile.c

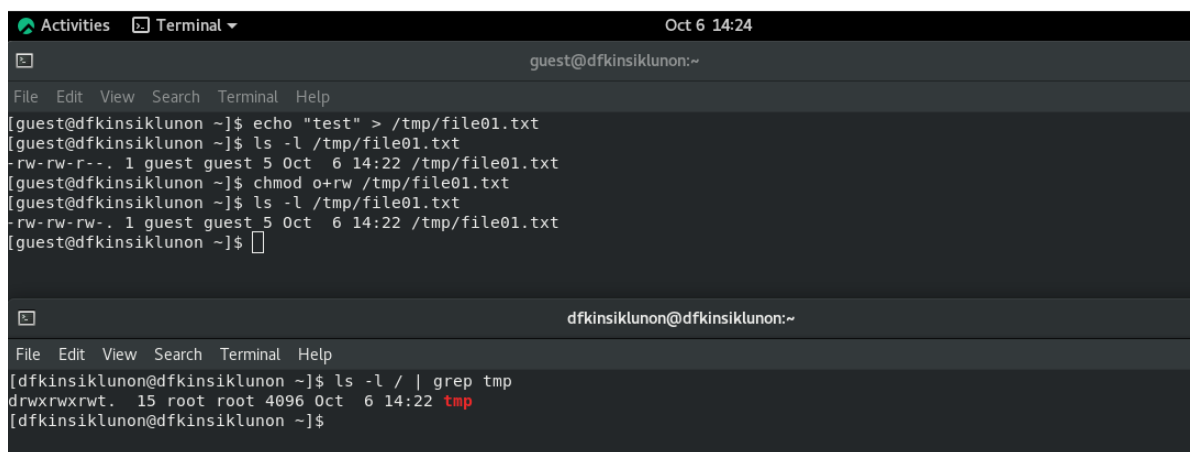
Скомпилировала созданную программу командой “gcc readfile.c -o readfile”. Сменила владельца у файла readfile.c командой “sudo chown root:guest /home/guest/readfile.c” и поменяла права так, чтобы только суперпользователь мог прочитать его, а guest не мог, с помощью команды “sudo chmod 700 /home/guest/readfile.c”. Теперь убедилась, что пользователь guest не может

прочитать файл readfile.c командой “cat readfile.c”, получив отказ в доступе (рис. 5.13).

Рис. 5.13: Смена владельца и прав доступа у файла readfile.c

Рис. 13: Рис. 5.13: Смена владельца и прав доступа у файла readfile.c

Поменяла владельца у программы readfile и установила SetUID. Проверила, может ли программа readfile прочитать файл readfile.c командой “./readfile readfile.c”. Прочитать удалось. Аналогично проверила, можно ли прочитать файл /etc/shadow. Прочитать удалось (рис. 5.14).



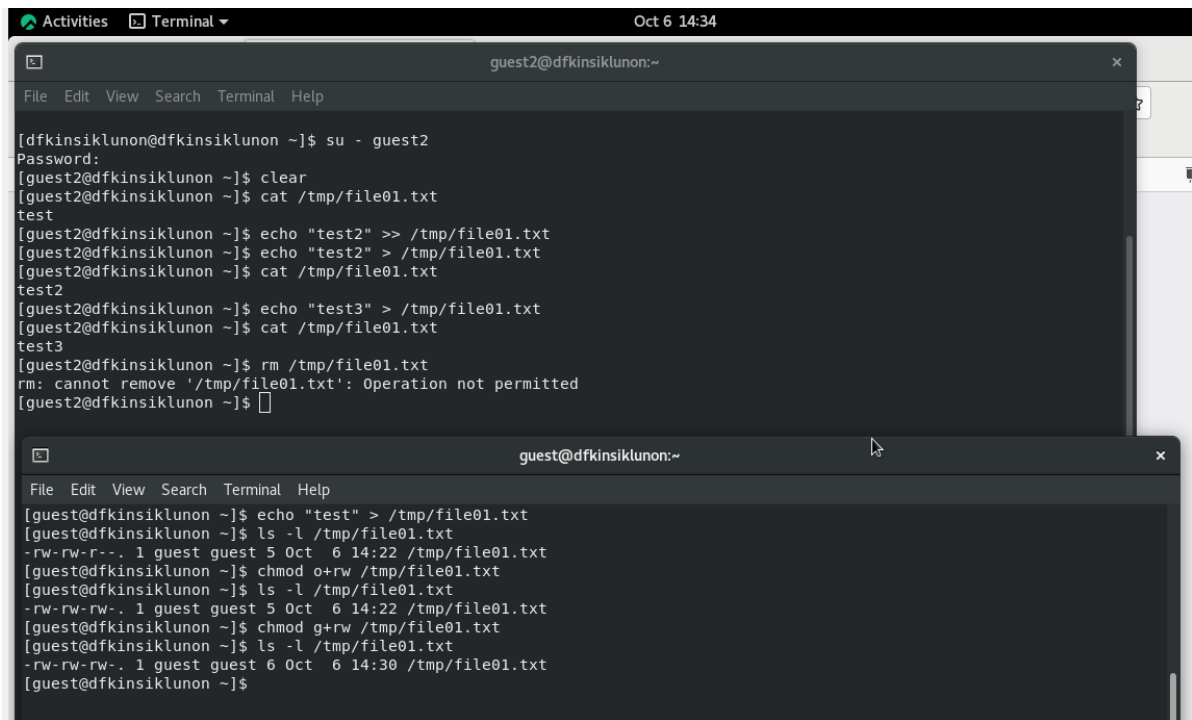
```
Oct 6 14:24
guest@dfkinsiklunon:~
File Edit View Search Terminal Help
[guest@dfkinsiklunon ~]$ echo "test" > /tmp/file01.txt
[guest@dfkinsiklunon ~]$ ls -l /tmp/file01.txt
-rw-rw-r--. 1 guest guest 5 Oct  6 14:22 /tmp/file01.txt
[guest@dfkinsiklunon ~]$ chmod o+rw /tmp/file01.txt
[guest@dfkinsiklunon ~]$ ls -l /tmp/file01.txt
-rw-rw-rw-. 1 guest guest 5 Oct  6 14:22 /tmp/file01.txt
[guest@dfkinsiklunon ~]$

dfkinsiklunon@dfkinsiklunon:~
File Edit View Search Terminal Help
[dfkinsiklunon@dfkinsiklunon ~]$ ls -l / | grep tmp
drwxrwxrwt. 15 root root 4096 Oct  6 14:22 tmp
[dfkinsiklunon@dfkinsiklunon ~]$
```

Рис. 14: Рис. 5.14: Запуск программы readfile

## 0.5 3.2 Исследование Sticky-бита

Командой “ls -l / | grep tmp” убеждалась, что атрибут Sticky на директории /tmp установлен. От имени пользователя guest создала файл file01.txt в директории /tmp со словом test командой “echo”test” > /tmp/file01.txt”. Просмотрела атрибуты у только что созданного файла и разрешаем чтение и запись для категории пользователей “все остальные” командами “ls -l /tmp/file01.txt” и “chmod o+rw /tmp/file01.txt” (рис. 5.15).



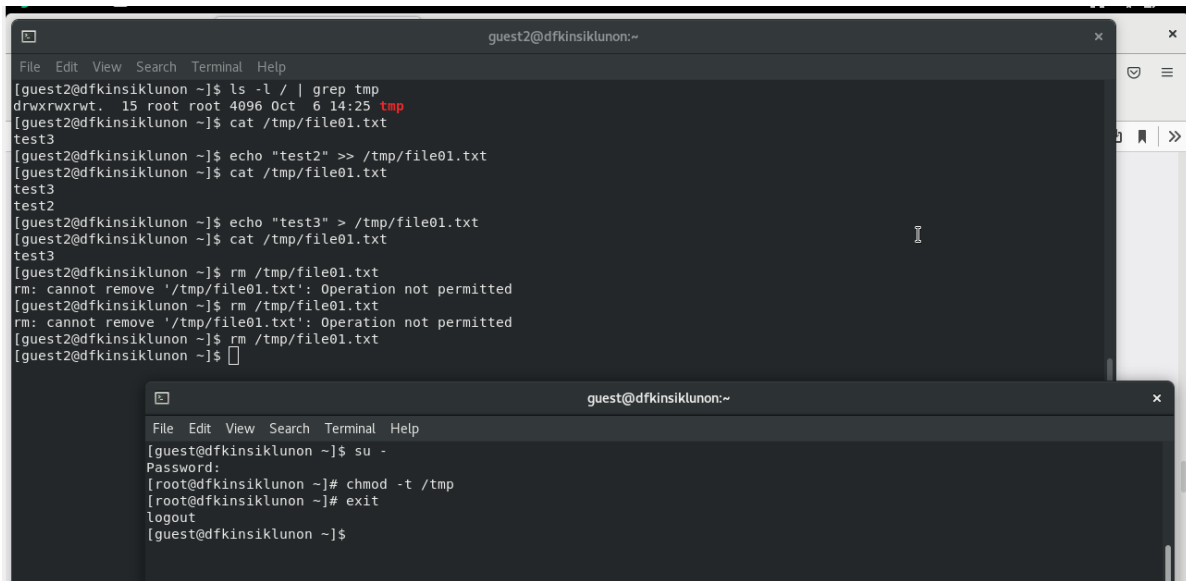
The image shows two terminal windows from a Linux desktop environment. The top window is titled 'guest2@dfkinsiklunon:~' and shows a sequence of commands: 'su - guest2', 'clear', 'cat /tmp/file01.txt' (output: 'test'), 'echo "test2" >> /tmp/file01.txt', 'echo "test2" > /tmp/file01.txt', 'cat /tmp/file01.txt' (output: 'test2'), 'echo "test3" > /tmp/file01.txt', 'cat /tmp/file01.txt' (output: 'test3'), and 'rm /tmp/file01.txt' (error: 'rm: cannot remove "/tmp/file01.txt": Operation not permitted'). The bottom window is titled 'guest@dfkinsiklunon:~' and shows: 'echo "test" > /tmp/file01.txt', 'ls -l /tmp/file01.txt' (output: '-rw-rw-r--. 1 guest guest 5 Oct 6 14:22 /tmp/file01.txt'), 'chmod o+rw /tmp/file01.txt', 'ls -l /tmp/file01.txt' (output: '-rw-rw-rw-. 1 guest guest 5 Oct 6 14:22 /tmp/file01.txt'), 'chmod g+rw /tmp/file01.txt', and 'ls -l /tmp/file01.txt' (output: '-rw-rw-rw-. 1 guest guest 6 Oct 6 14:30 /tmp/file01.txt').

```
[dfkinsiklunon@dfkinsiklunon ~]$ su - guest2
Password:
[guest2@dfkinsiklunon ~]$ clear
[guest2@dfkinsiklunon ~]$ cat /tmp/file01.txt
test
[guest2@dfkinsiklunon ~]$ echo "test2" >> /tmp/file01.txt
[guest2@dfkinsiklunon ~]$ echo "test2" > /tmp/file01.txt
[guest2@dfkinsiklunon ~]$ cat /tmp/file01.txt
test2
[guest2@dfkinsiklunon ~]$ echo "test3" > /tmp/file01.txt
[guest2@dfkinsiklunon ~]$ cat /tmp/file01.txt
test3
[guest2@dfkinsiklunon ~]$ rm /tmp/file01.txt
rm: cannot remove '/tmp/file01.txt': Operation not permitted
[guest2@dfkinsiklunon ~]$
```

```
[guest@dfkinsiklunon ~]$ echo "test" > /tmp/file01.txt
[guest@dfkinsiklunon ~]$ ls -l /tmp/file01.txt
-rw-rw-r--. 1 guest guest 5 Oct 6 14:22 /tmp/file01.txt
[guest@dfkinsiklunon ~]$ chmod o+rw /tmp/file01.txt
[guest@dfkinsiklunon ~]$ ls -l /tmp/file01.txt
-rw-rw-rw-. 1 guest guest 5 Oct 6 14:22 /tmp/file01.txt
[guest@dfkinsiklunon ~]$ chmod g+rw /tmp/file01.txt
[guest@dfkinsiklunon ~]$ ls -l /tmp/file01.txt
-rw-rw-rw-. 1 guest guest 6 Oct 6 14:30 /tmp/file01.txt
[guest@dfkinsiklunon ~]$
```

Рис. 15: Рис. 5.15: Создание файла file01.txt

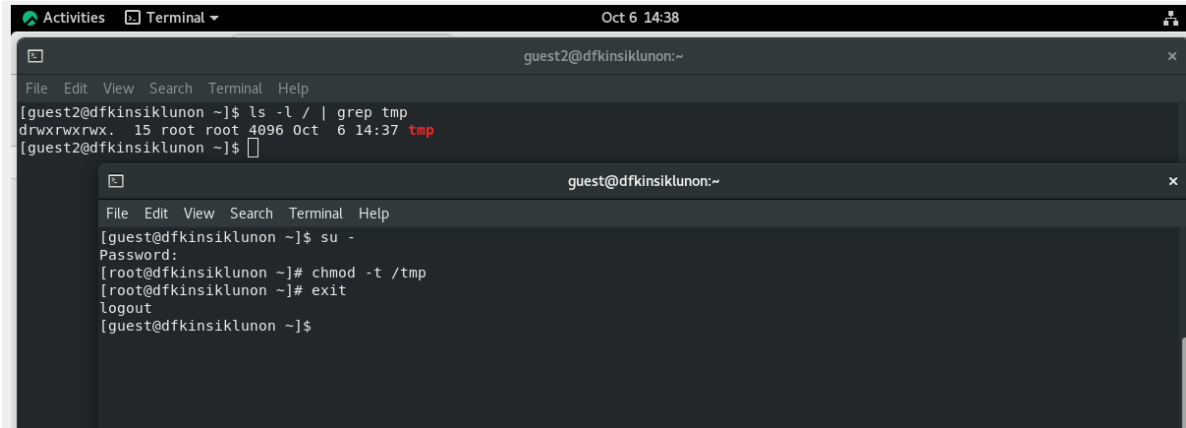
От имени пользователя guest2 попробовала прочитать файл командой “cat /tmp/file01.txt” - это удалось. Далее попыталась дозаписать в файл слово test2, проверить содержимое файла и записать в файл слово test3, стерев при этом всю имеющуюся в файле информацию - эти операции удалось выполнить только в случае, если еще дополнительно разрешить чтение и запись для группы пользователей командой “chmod g+rw /tmp/file01.txt”. От имени пользователя guest2 попробовала удалить файл - это не удастся ни в каком из случаев, возникает ошибка (рис. 5.16).



```
guest2@dfkinsiklunon:~  
File Edit View Search Terminal Help  
[guest2@dfkinsiklunon ~]$ ls -l / | grep tmp  
drwxrwxrwt. 15 root root 4096 Oct 6 14:25 tmp  
[guest2@dfkinsiklunon ~]$ cat /tmp/file01.txt  
test3  
[guest2@dfkinsiklunon ~]$ echo "test2" >> /tmp/file01.txt  
[guest2@dfkinsiklunon ~]$ cat /tmp/file01.txt  
test3  
test2  
[guest2@dfkinsiklunon ~]$ echo "test3" > /tmp/file01.txt  
[guest2@dfkinsiklunon ~]$ cat /tmp/file01.txt  
test3  
[guest2@dfkinsiklunon ~]$ rm /tmp/file01.txt  
rm: cannot remove '/tmp/file01.txt': Operation not permitted  
[guest2@dfkinsiklunon ~]$ rm /tmp/file01.txt  
rm: cannot remove '/tmp/file01.txt': Operation not permitted  
[guest2@dfkinsiklunon ~]$ rm /tmp/file01.txt  
[guest2@dfkinsiklunon ~]$  
  
guest@dfkinsiklunon:~  
File Edit View Search Terminal Help  
[guest@dfkinsiklunon ~]$ su -  
Password:  
[root@dfkinsiklunon ~]# chmod -t /tmp  
[root@dfkinsiklunon ~]# exit  
logout  
[guest@dfkinsiklunon ~]$
```

Рис. 16: Попытка выполнить действия над файлом file01.txt от имени пользователя guest2

Повысила права до суперпользователя командой “su -” и выполнила команду, снимающую атрибут t с директории /tmp “chmod -t /tmp”. После чего покинула режим суперпользователя командой “exit”. Повторила предыдущие шаги. Теперь мне удалось удалить файл file01.txt от имени пользователя, не являющегося его владельцем (рис. 5.17).



```
Activities Terminal Oct 6 14:38  
guest2@dfkinsiklunon:~  
File Edit View Search Terminal Help  
[guest2@dfkinsiklunon ~]$ ls -l / | grep tmp  
drwxrwxrwx. 15 root root 4096 Oct 6 14:37 tmp  
[guest2@dfkinsiklunon ~]$  
  
guest@dfkinsiklunon:~  
File Edit View Search Terminal Help  
[guest@dfkinsiklunon ~]$ su -  
Password:  
[root@dfkinsiklunon ~]# chmod -t /tmp  
[root@dfkinsiklunon ~]# exit  
logout  
[guest@dfkinsiklunon ~]$
```

Рис. 17: Удаление атрибута t (Sticky-бита) и повторение действий

Повысила свои права до суперпользователя и вернула атрибут t на директорию

/tmp (рис. 5.18).

Рис. 5.18: Возвращение атрибута t (Sticky-бита)

Рис. 18: Рис. 5.18: Возвращение атрибута t (Sticky-бита)

## 0.6 Выводы

ходе выполнения данной лабораторной работы я изучила механизмы изменения идентификаторов, применение SetUID- и Sticky-битов. Получила практические навыки работы в консоли с дополнительными атрибутами. Рассмотрела работу механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.