

Отчёт по лабораторной работе № 5

Дисциплина: Основы информационной безопасности

Перелыгин Сергей Викторович

Содержание

1	Цель работы	4
2	Задание	5
3	Теоретическое введение	6
4	Выполнение лабораторной работы	7
5	Выводы	18
6	Библиография	19

Список иллюстраций

4.1	Предварительная подготовка 1	7
4.2	Предварительная подготовка 2	7
4.3	Команда “whereis”	8
4.4	Предварительная подготовка 3	8
4.5	Предварительная подготовка 4	9
4.6	Предварительная подготовка 5	9
4.7	Компиляция и выполнение программы simpleid2	10
4.8	Установка новых атрибутов (SetUID)	10
4.9	Проверка правильности установки новых атрибутов	10
4.10	Запуск simpleid2 после установки SetUID	11
4.11	Запуск simpleid2 после установки SetGID	11
4.12	Программа readfile.c	12
4.13	Компиляция readfile.c	13
4.14	Смена владельца и прав доступа у файла readfile.c	13
4.15	Попытка прочитать файл	13
4.16	Запуск программы readfile	14
4.17	Файл /etc/shadow	14
4.18	Создание файла file01.txt	15
4.19	Попытка выполнить действия над файлом file01.txt от имени пользо- вателя guest2	16
4.20	Удаление атрибута t (Sticky-бита) и повторение действий	16
4.21	Возвращение атрибута t (Sticky-бита)	17

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

2 Задание

- Сделать отчёт по лабораторной работе в формате Markdown.
- В качестве отчёта предоставить отчёты в 3 форматах: pdf, docx и md.

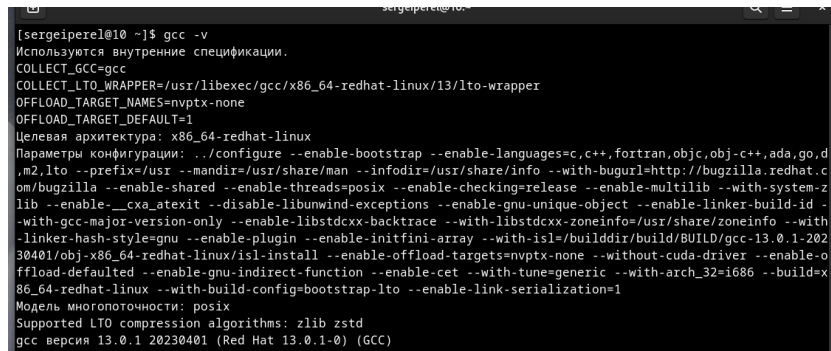
3 Теоретическое введение

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.

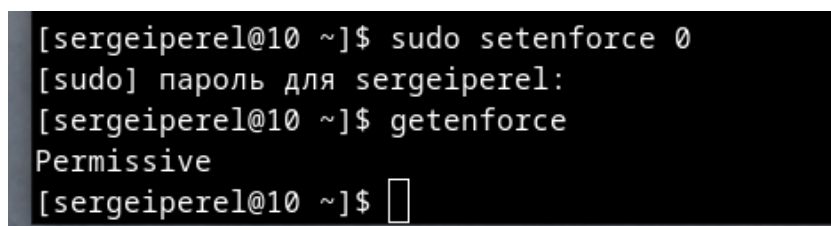
4 Выполнение лабораторной работы

1. Для начала я убедился, что компилятор gcc установлен, используя команду “gcc -v”. Затем отключил систему запретов до очередной перезагрузки системы командой “sudo setenforce 0”, после чего команда “getenforce” вывела “Permissive” (рис. 4.1 и 4.2).



```
[sergeiperel@10 ~]$ gcc -v
Используются внутренние спецификации.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/usr/libexec/gcc/x86_64-redhat-linux/13/lto-wrapper
OFFLOAD_TARGET_NAMES=nvptx-none
OFFLOAD_TARGET_DEFAULT=1
Целевая архитектура: x86_64-redhat-linux
Параметры конфигурации: ../configure --enable-bootstrap --enable-languages=c,c++,fortran,objc,obj-c++,ada,go,d
,m2,lto --prefix=/usr --mandir=/usr/share/man --infodir=/usr/share/info --with-bugurl=http://bugzilla.redhat.c
om/bugzilla --enable-shared --enable-threads=posix --enable-checking=release --enable-multilib --with-system-z
lib --enable-__cxa_atexit --disable-libunwind-exceptions --enable-gnu-unique-object --enable-linker-build-id --
with-gcc-major-version-only --enable-libstdcxx-backtrace --with-libstdcxx-zoneinfo=/usr/share/zoneinfo --with
-linker-hash-style=gnu --enable-plugin --enable-initfini-array --with-isl=/build/isl-0.24/build/BUILD/isl-0.24
30401/obj-x86_64-redhat-linux/isl-install --enable-offload-targets=nvptx-none --without-cuda-driver --enable-o
ffload-defaulted --enable-gnu-indirect-function --enable-cet --with-tune=generic --with-arch_32=i686 --build=x
86_64-redhat-linux --with-build-config=bootstrap-lto --enable-link-serialization=1
Модель многопоточности: posix
Supported LTO compression algorithms: zlib zstd
gcc версия 13.0.1 20230401 (Red Hat 13.0.1-0) (GCC)
```

Рис. 4.1: Предварительная подготовка 1



```
[sergeiperel@10 ~]$ sudo setenforce 0
[sudo] пароль для sergeiperel:
[sergeiperel@10 ~]$ getenforce
Permissive
[sergeiperel@10 ~]$
```

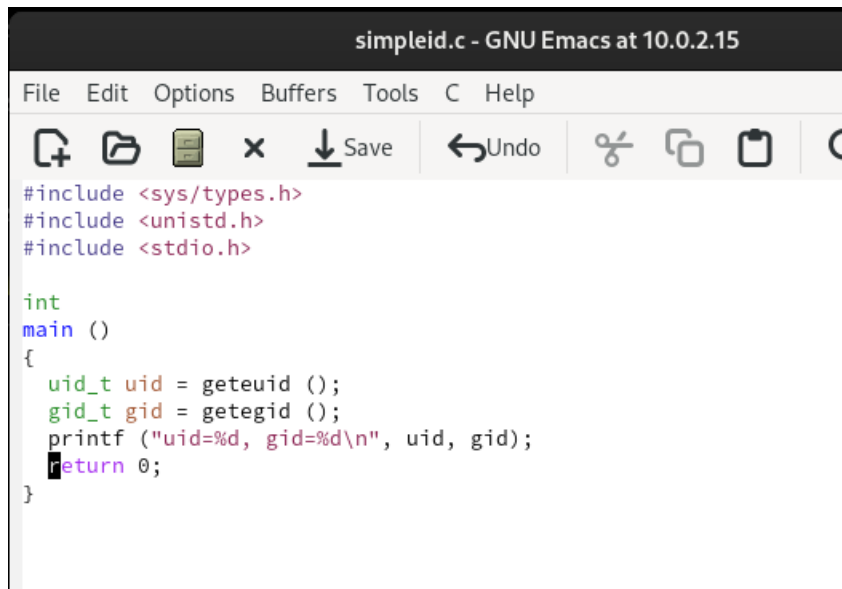
Рис. 4.2: Предварительная подготовка 2

2. Проверил успешное выполнение команд “whereis gcc” и “whereis g++” (их расположение) (рис. 4.3).

```
[sergeiperel@10 ~]$ whereis gcc
gcc: /usr/bin/gcc /usr/lib/gcc /usr/libexec/gcc /usr/s
[sergeiperel@10 ~]$ whereis g++
g++:
[sergeiperel@10 ~]$
```

Рис. 4.3: Команда “whereis”

3. Вошел в систему от имени пользователя guest командой “su - guest”. Создал программу simpleid.c командой “touch simpleid.c” и открыл её в редакторе emacs.
4. Код программы выглядит следующим образом (рис. 4.4).



```
simpleid.c - GNU Emacs at 10.0.2.15
File Edit Options Buffers Tools C Help
[Icons] Save Undo [Icons]
#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. 4.4: Предварительная подготовка 3

5. Скомпилировал программу и убедился, что файл программы был создан командой “gcc simpleid.c -o simpleid”. Выполнил программу simpleid командой “./simpleid”, а затем выполнил системную программу id командой “id”. Результаты, полученные в результате выполнения обеих команд, совпадают (uid=1001 и gid=1001) (рис. 4.5).


```
guest@10:~  
[sergeiperel@10 ~]$ su - guest  
Пароль:  
[guest@10 ~]$ touch simpleid.c  
[guest@10 ~]$ ls  
simpleid.c  
[guest@10 ~]$ emacs simpleid.c  
  
(emacs:4493): dbind-WARNING **: 19:56:21.288: Couldn't connect to accessibility bus: Отказано в доступе  
[guest@10 ~]$ gcc simpleid.c -o simpleid  
[guest@10 ~]$ ./simpleid  
uid=1001, gid=1001  
[guest@10 ~]$ id  
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfined_r:sandbox_t:s0  
[guest@10 ~]$ ls  
simpleid simpleid.c simpleid.c~  
[guest@10 ~]$
```

Рис. 4.5: Предварительная подготовка 4

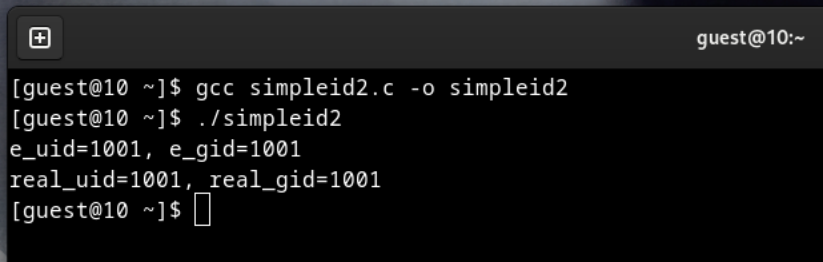
6. Усложнил программу, добавив вывод действительных идентификаторов (рис. 4.6).

```
simpleid.c - GNU Emacs at 10.0.2.15  
File Edit Options Buffers Tools C Help  
[Icons] Save Undo [Icons] Search  
  
#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. 4.6: Предварительная подготовка 5

7. Получившуюся программу назвал simpleid2.c.

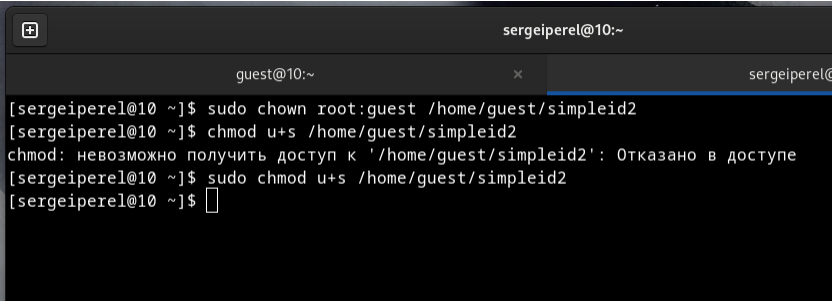
8. Скомпилировал и запустил simpleid2.c командами “gcc simpleid2.c -o simpleid2” и “./simpleid2” (рис. 4.7).



```
guest@10:~  
[guest@10 ~]$ gcc simpleid2.c -o simpleid2  
[guest@10 ~]$ ./simpleid2  
e_uid=1001, e_gid=1001  
real_uid=1001, real_gid=1001  
[guest@10 ~]$
```

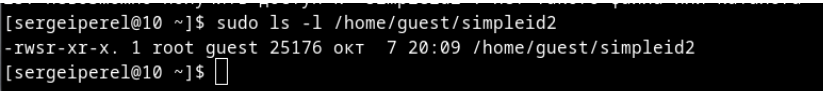
Рис. 4.7: Компиляция и выполнение программы simpleid2

9. От имени суперпользователя выполнил команды “sudo chown root:guest /home/guest/simpleid2” и “sudo chmod u+s /home/guest/simpleid2” (рис. 4.8), затем выполнил проверку правильности установки новых атрибутов и смены владельца файла simpleid2 командой “sudo ls -l /home/guest/simpleid2” (рис. 4.9). Этими командами была произведена смена пользователя файла на root и установлен SetUID-бит.



```
sergeiperel@10:~  
[sergeiperel@10 ~]$ sudo chown root:guest /home/guest/simpleid2  
[sergeiperel@10 ~]$ chmod u+s /home/guest/simpleid2  
chmod: невозможно получить доступ к '/home/guest/simpleid2': Отказано в доступе  
[sergeiperel@10 ~]$ sudo chmod u+s /home/guest/simpleid2  
[sergeiperel@10 ~]$
```

Рис. 4.8: Установка новых атрибутов (SetUID)



```
[sergeiperel@10 ~]$ sudo ls -l /home/guest/simpleid2  
-rwsr-xr-x. 1 root guest 25176 окт 7 20:09 /home/guest/simpleid2  
[sergeiperel@10 ~]$
```

Рис. 4.9: Проверка правильности установки новых атрибутов

10. Запустил программы `simpleid2` и `id`. Теперь появились различия в `uid` (рис. 4.10).

```
[guest@10 ~]$ ./simpleid2
e_uid=0, e_gid=1001
real_uid=1001, real_gid=1001
[guest@10 ~]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfined_r:023
[guest@10 ~]$
```

Рис. 4.10: Запуск `simpleid2` после установки `SetUID`

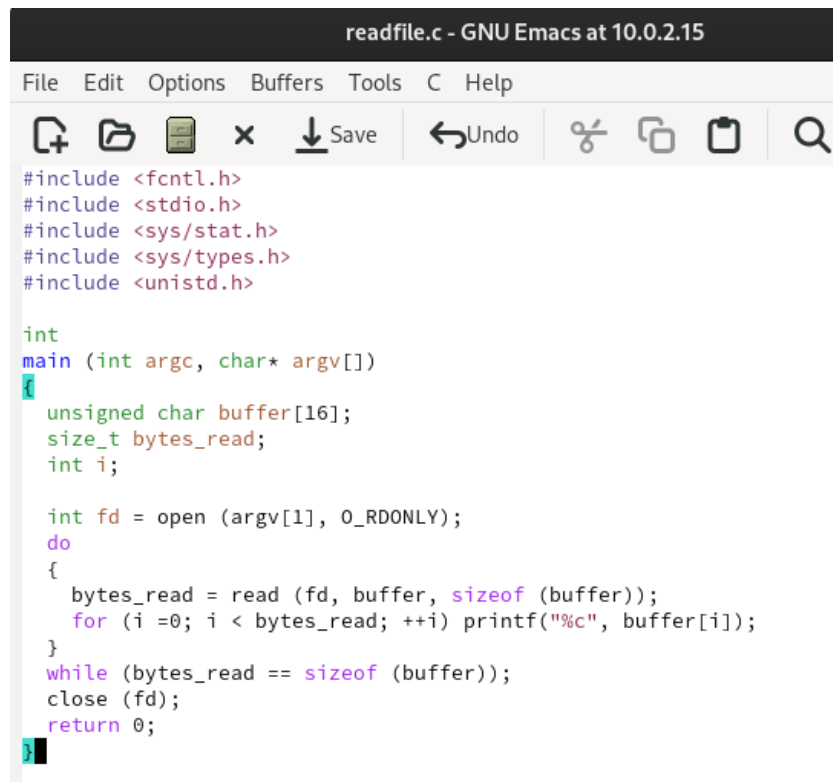
11. Прodelал тоже самое относительно `SetGID`-бита. Также можем заметить различия с предыдущим пунктом (рис. 4.11).

```
sergeiperel@10:~
[sergeiperel@10 ~]$ sudo chown root:guest /home/guest/simpleid2
[sergeiperel@10 ~]$ sudo chmod g+s /home/guest/simpleid2
[sergeiperel@10 ~]$

guest@10:~
[guest@10 ~]$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
[guest@10 ~]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfined_r:0:c0.c1023
[guest@10 ~]$
```

Рис. 4.11: Запуск `simpleid2` после установки `SetGID`

12. Создаем программу `readfile.c` (рис. 4.12).



```
#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. 4.12: Программа readfile.c

13. Скомпилировал созданную программу командой “gcc readfile.c -o readfile” (рис. 4.13). Сменил владельца у файла readfile.c командой “sudo chown root:guest /home/guest/readfile.c” и поменял права так, чтобы только суперпользователь мог прочитать его, а guest не мог, с помощью команды “sudo chmod 700 /home/guest/readfile.c” (рис. 4.14). Теперь убедился, что пользователь guest не может прочитать файл readfile.c командой “cat readfile.c”, получив отказ в доступе (рис. 4.15).

```
[guest@10 ~]$ emacs readfile.c&
[1] 6593
[guest@10 ~]$
(emacs:6593): dbind-WARNING **: 20:31:45.621: Couldn't connect to accessibility
 в доступе

[1]+  Завершён      emacs readfile.c
[guest@10 ~]$ gcc readfile.c -o readfile
[guest@10 ~]$
```

Рис. 4.13: Компиляция readfile.c

```
[sergeiperel@10 ~]$ sudo chown root:guest /home/guest/readfile.c
[sudo] пароль для sergeiperel:
[sergeiperel@10 ~]$ sudo chmod 700 /home/guest/readfile.c
[sergeiperel@10 ~]$
```

Рис. 4.14: Смена владельца и прав доступа у файла readfile.c

```
[guest@10 ~]$ cat readfile.c
cat: readfile.c: Отказано в доступе
[guest@10 ~]$
```

Рис. 4.15: Попытка прочитать файл

14. Поменял владельца у программы readfile и установила SetUID. Проверил, может ли программа readfile прочитать файл readfile.c командой “./readfile readfile.c”. Прочитать удалось (рис. 4.16). Аналогично проверил, можно ли прочитать файл /etc/shadow. Прочитать не удалось (рис. 4.17).

```

[guest@10 ~]$ ./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;
}

```

Рис. 4.16: Запуск программы readfile

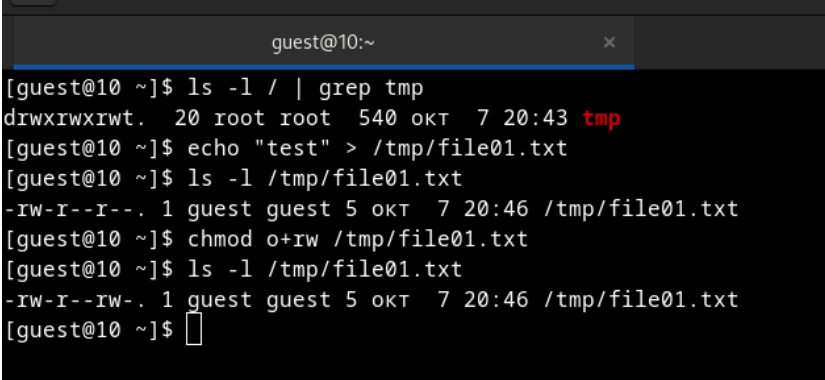
```

[guest@10 ~]$ ./readfile /etc/shadow
root:!:0:99999:7:::
bin:*.19378:0:99999:7:::
daemon:*.19378:0:99999:7:::
adm:*.19378:0:99999:7:::
lp:*.19378:0:99999:7:::
sync:*.19378:0:99999:7:::
shutdown:*.19378:0:99999:7:::
halt:*.19378:0:99999:7:::
mail:*.19378:0:99999:7:::
operator:*.19378:0:99999:7:::
games:*.19378:0:99999:7:::
ftp:*.19378:0:99999:7:::
nobody:*.19378:0:99999:7:::
dbus:!!:19460:::::
apache:!!:19460:::::
tss:!!:19460:::::
systemd-network:!*:19460:::::
systemd-oom:!*:19460:::::
systemd-resolve:!*:19460:::::
qemu:!!:19460:::::
polkitd:!!:19460:::::
avahi:!!:19460:::::
geoclue:!!:19460:::::
nm-openconnect:!!:19460:::::
colord:!!:19460:::::
usbmuxd:!!:19460:::::
gluster:!!:19460:::::
rtkit:!!:19460:::::

```

Рис. 4.17: Файл /etc/shadow

15. Командой “ls -l / | grep tmp” убедился, что атрибут Sticky на директории /tmp установлен. От имени пользователя guest создал файл file01.txt в директории /tmp со словом test командой “echo test” > /tmp/file01.txt”. Просмотрел 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. 4.18).



```
guest@10:~  
[guest@10 ~]$ ls -l / | grep tmp  
drwxrwxrwt. 20 root root 540 окт 7 20:43 tmp  
[guest@10 ~]$ echo "test" > /tmp/file01.txt  
[guest@10 ~]$ ls -l /tmp/file01.txt  
-rw-r--r--. 1 guest guest 5 окт 7 20:46 /tmp/file01.txt  
[guest@10 ~]$ chmod o+rw /tmp/file01.txt  
[guest@10 ~]$ ls -l /tmp/file01.txt  
-rw-r--rw-. 1 guest guest 5 окт 7 20:46 /tmp/file01.txt  
[guest@10 ~]$
```

Рис. 4.18: Создание файла file01.txt

16. От имени пользователя guest2 попробовал прочитать файл командой “cat /tmp/file01.txt” - это удалось. Далее попытался дозаписать в файл слово test2, проверить содержимое файла и записать в файл слово test3, стеревав при этом всю имеющуюся в файле информацию - эти операции выполнить не удалось. От имени пользователя guest2 попробовал удалить файл - это не удастся, возникает ошибка (рис. 4.19).

```

[sergeiperel@10 ~]$ su - guest2
Пароль:
[guest2@10 ~]$ cat /tmp/file01.txt
test
[guest2@10 ~]$ echo "test2" > /tmp/file01.txt
-bash: /tmp/file01.txt: Отказано в доступе
[guest2@10 ~]$ cat /tmp/file01.txt
test
[guest2@10 ~]$ echo "test3" > /tmp/file01.txt
-bash: /tmp/file01.txt: Отказано в доступе
[guest2@10 ~]$ cat /tmp/file01.txt
test
[guest2@10 ~]$ rm /tmp/file01.txt
rm: невозможно удалить '/tmp/file01.txt': Нет такого файла или каталога
[guest2@10 ~]$

```

Рис. 4.19: Попытка выполнить действия над файлом file01.txt от имени пользователя guest2

17. Повысил права до суперпользователя командой “sudo su -” и выполнил команду, снимающую атрибут t с директории /tmp “chmod -t /tmp”. После чего покинул режим суперпользователя командой “exit”. Повторил предыдущие шаги. Теперь мне удалось выполнить все, кроме команды удалить файл file01.txt от имени пользователя, не являющегося его владельцем (рис. 4.20).

```

[guest2@10 ~]$ ls -l / | grep tmp
drwxrwxrwx. 20 root root 560 окт 7 21:07 tmp
[guest2@10 ~]$ cat /tmp/file01.txt
test
[guest2@10 ~]$ echo "test2" > /tmp/file01.txt
[guest2@10 ~]$ cat /tmp/file01.txt
test2
[guest2@10 ~]$ echo "test3" >> /tmp/file01.txt
[guest2@10 ~]$ cat /tmp/file01.txt
test2
test3
[guest2@10 ~]$ rm /tmp/file01.txt
rm: невозможно удалить '/tmp/file01.txt': Нет такого файла или каталога

```

Рис. 4.20: Удаление атрибута t (Sticky-бита) и повторение действий

18. Повысил свои права до суперпользователя и вернул атрибут t на директорию /tmp (рис. 4.21).


```
[sergeiperel@10 ~]$ sudo su -  
[sudo] пароль для sergeiperel:  
[root@10 ~]# chmod +t /tmp  
[root@10 ~]# exit  
выход  
[sergeiperel@10 ~]$
```

Рис. 4.21: Возвращение атрибута t (Sticky-бита)

5 Выводы

Вывод: В ходе выполнения данной лабораторной работы я изучил механизмы изменения идентификаторов, применение 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.

6 Библиография

- Медведовский И.Д., Семьянов П.В., Платонов В.В. Атака через Internet. — НПО “Мир и семья-95”, 1997. — URL: <http://bugtraq.ru/library/books/attack1/index.html>
- Медведовский И.Д., Семьянов П.В., Леонов Д.Г. Атака на Internet. — Издательство ДМК, 1999. — URL: <http://bugtraq.ru/library/books/attack/index.html>
- Запечников С. В. и др. Информационная безопасность открытых систем. Том 1. — М.: Горячая линия -Телеком, 2006.
- Введение в информационную безопасность. Типы уязвимостей. (Д.Гамаюнов, МГУ)
- Практические аспекты сетевой безопасности. Вводная лекция. Сетевая безопасность. Стек протоколов TCP/IP. (Д. Гамаюнов, МГУ)
- Практические аспекты сетевой безопасности. Сетевая безопасность. Межсетевые экраны. (В. Иванов, МГУ)
- Практические аспекты сетевой безопасности. Сетевая безопасность. Системы обнаружения и фильтрации компьютерных атак (IDS/IPS). (Д. Гамаюнов, МГУ)
- Практические аспекты сетевой безопасности. Контроль нормального поведения приложений. Security Enhanced Linux (SELinux) (В. Сахаров, МГУ)