

# **Информационная безопасность.**

**Лабораторная работа №5.**

**Подмогильный Иван Александрович.**

# **Содержание**

<b>1. Цель работы</b>	<b>5</b>
<b>2. Задание</b>	<b>6</b>
<b>3. Выполнение лабораторной работы</b>	<b>7</b>
<b>4. Выводы</b>	<b>21</b>

# Список иллюстраций

3.1. Установка gcc . . . . .	7
3.2. Отключение SELinux . . . . .	8
3.3. Создание файла и программы simpleid.c . . . . .	8
3.4. Компилирование файла . . . . .	9
3.5. Выполнение программы simpleid . . . . .	9
3.6. Выполнение программы id . . . . .	10
3.7. Усложнение программы . . . . .	10
3.8. Компиляция и запуск . . . . .	11
3.9. Добавление SetUID . . . . .	11
3.10. Действия над файлом. . . . .	12
3.11. Установка SetGID . . . . .	12
3.12. Запуск программы с установленным SetGID . . . . .	13
3.13. Создание программы readfile.c . . . . .	13
3.14. Компиляция программы . . . . .	14
3.15. Смена владельца у файла readfile.c . . . . .	14
3.16. Проверка . . . . .	15
3.17. Установил SetU'D бит . . . . .	15
3.18. Проверка . . . . .	16
3.19. Проверка . . . . .	16
3.20. Sticky бит на папке tmp . . . . .	17
3.21. Создание файла . . . . .	17
3.22. Просмотр атрибутов . . . . .	18
3.23. Чтение файла . . . . .	18
3.24. Запись и дозапись . . . . .	19
3.25. Попытка удалить файл . . . . .	19
3.26. Снятие Sticky атрибута . . . . .	20
3.27. Выход из суперпользователя . . . . .	20

# **Список таблиц**

# **1. Цель работы**

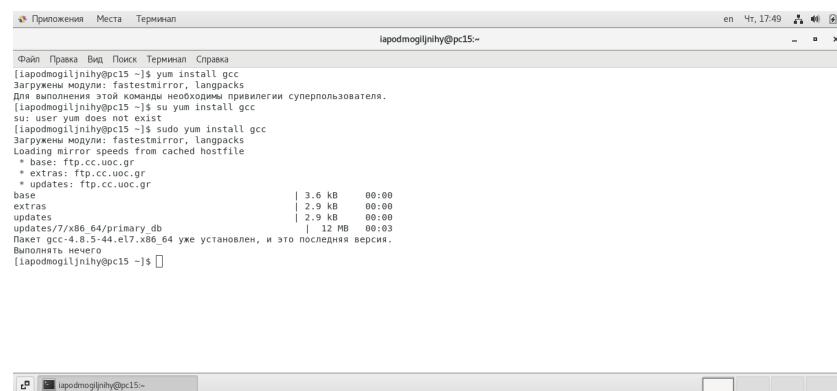
Изучение механизмов изменения идентификаторов, применения SetUID- и Sticky-битов. Получение практических навыков работы в консоли с дополнительными атрибутами. Рассмотрение работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов

## **2. Задание**

- 1) Выполнить пункты из задания по порядку.

### 3. Выполнение лабораторной работы

Установил gcc с помощью команды `yum install gcc`



```
Приложения Места Терминал Справка
[lapodmogiljnihy@pc15 ~]$ yum install gcc
[lapodmogiljnihy@pc15 ~]$ sudo yum install gcc
Загружены модули: fastestmirror, langpacks
Для выполнения этого действия необходимы привилегии суперпользователя.
[lapodmogiljnihy@pc15 ~]$ su yum install gcc
su: user yum does not exist
[lapodmogiljnihy@pc15 ~]$ sudo yum install gcc
Загружены модули: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: ftp.cc.uoc.cz
 * extras: ftp.cc.uoc.cz
 * updates: ftp.cc.uoc.cz
base                                         | 3.6 kB   00:00
extras                                        | 2.9 kB   00:00
updates                                       | 2.9 kB   00:00
updates/7/x86_64/primary_db                   | 12 MB    00:03
Пакет gcc-4.8.5-44.el7.x86_64 уже установлен, и это последняя версия.
Выполнить нечего
[lapodmogiljnihy@pc15 ~]$
```

Рис. 3.1.: Установка gcc

Отменил на текущую сессию SELinux командой `setenforce 0`

```

Приложения Места Терминал en Чт, 18:44
iaopdmogiljnihy@pc15:/home/iaopdmogiljnihy - x
Файл Правка Вид Поиск Терминал Справка
Загружены модули: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
* base: ftp.cc.uoc.gr
* extras: ftp.cc.uoc.gr
* updates: ftp.cc.uoc.gr
base | 3.6 kB 00:00
extras | 2.9 kB 00:00
updates | 2.9 kB 00:00
updates/7/x86_64/primary_db | 12 MB 00:03
Пакет gcc-4.8.5-44.el7.x86_64 уже установлен, и это последняя версия.
Выполнять нечего
[iaopdmogiljnihy@pc15 ~]$ setenforce 0
setenforce: setenforce() failed
[iaopdmogiljnihy@pc15 ~]$ setenforce 0
setenforce: setenforce() failed
[iaopdmogiljnihy@pc15 ~]$ getenforce
Enforcing
[iaopdmogiljnihy@pc15 ~]$ nosuid
bash: nosuid: команда не найдена...
[iaopdmogiljnihy@pc15 ~]$ setenforce 0
setenforce: setenforce() failed
[iaopdmogiljnihy@pc15 ~]$ su
Пароль:
[root@pc15 iaopdmogiljnihy]# setenforce 0
[root@pc15 iaopdmogiljnihy]# getenforce
Permissive
[root@pc15 iaopdmogiljnihy]# 

```

Рис. 3.2.: Отключение SELinux

Вошёл в систему от имени пользователя guest, создал программу simpleid.c

```

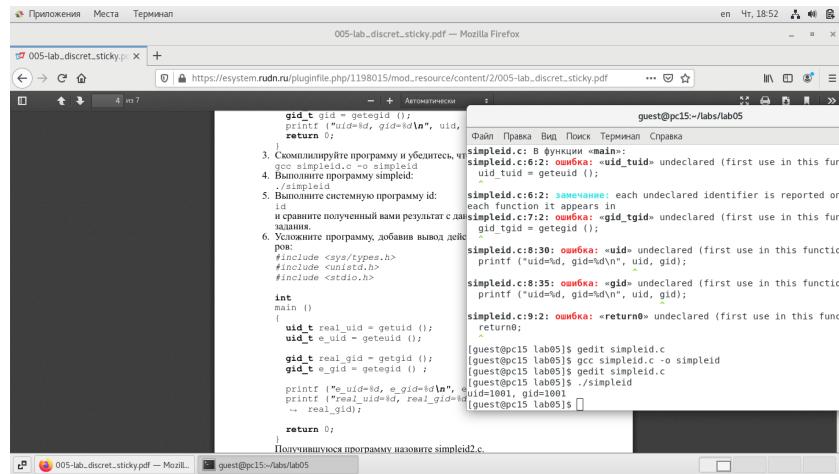
Приложения Места Терминал en Чт, 18:51
simpleid.c Сохранить - x
Файл Правка Вид Поиск Терминал Справка
guest@pc15:~/labs/lab05 - x
simpleid.c:1:1: warning: #include "simpleid.h" is never used [-Wunused-includes]
 #include <sys/types.h>
^
simpleid.c:2:1: warning: #include <unistd.h> is never used [-Wunused-includes]
 #include <unistd.h>
^
simpleid.c:3:1: warning: #include <stdio.h> is never used [-Wunused-includes]
 #include <stdio.h>
^
int main () {
    uid_t uid = geteuid ();
    gid_t gid = getegid ();
    printf ("uid=%d, gid=%d\n", uid, gid);
    return 0;
}
** (gedit:6246): WARNING **: 18:49:30.123: Set document metadata failed: Невозможно установить ключ метаданных
** (gedit:6246): WARNING **: 18:49:52.511: Set document metadata failed: Невозможно установить ключ метаданных
** (gedit:6246): WARNING **: 18:49:52.515: Set document metadata failed: Невозможно установить ключ метаданных
** (gedit:6246): WARNING **: 18:49:55.458: Set document metadata failed: Невозможно установить ключ метаданных
(guest@pc15:~/labs/lab05) $ gcc simpleid.c -o simpleid
simpleid.c: In function 'main':
simpleid.c:6:2: warning: 'uid_t' undeclared (first use in this function)
uid_t uid = geteuid ();
^
simpleid.c:6:2: warning: each undeclared identifier is reported only once
for each function it appears in
simpleid.c:7:2: warning: 'gid_t' undeclared (first use in this function)
gid_t gid = getegid ();
^
simpleid.c:8:38: warning: 'uid' undeclared (first use in this function)
printf ("uid=%d, gid=%d\n", uid, gid);
^
simpleid.c:8:35: warning: 'gid' undeclared (first use in this function)
printf ("uid=%d, gid=%d\n", uid, gid);
^
simpleid.c:9:2: warning: 'return' undeclared (first use in this function)
return;
^
(guest@pc15:~/labs/lab05) $ gedit simpleid.c
(guest@pc15:~/labs/lab05) $ gcc simpleid.c -o simpleid
(guest@pc15:~/labs/lab05) $ gedit simpleid.c

```

Рис. 3.3.: Создание файла и программы simpleid.c

Скомпилировал программу и убедился, что файл программы создан: gcc

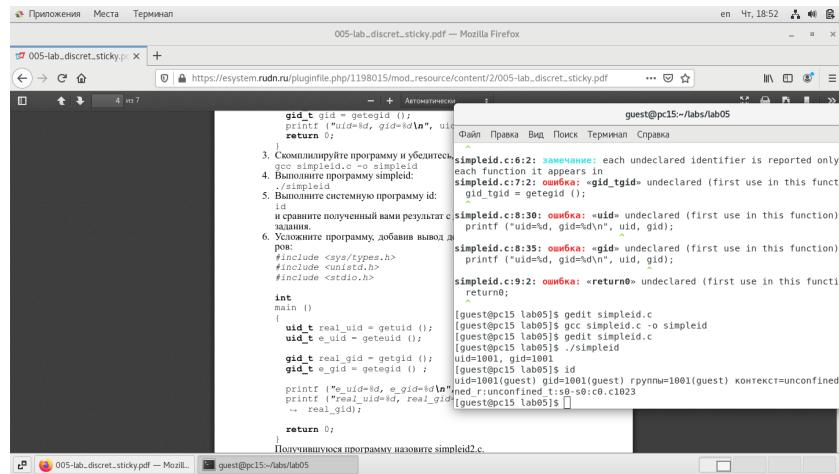
**simpleid.c -o simpleid**



```
005-lab..discret_sticky.pdf -- Mozilla Firefox
https://esystem.rudn.ru/pluginfile.php/1198015/mod_resource/content/2/005-lab..discret_sticky.pdf
guest@pc15:~/labs/lab05
File Правка Вид Поиск Терминал Справка
simpleid.c: В функции «main»:
simpleid.c:6:2: ошибка: «uid_tuid» undeclared (first use in this function)
uid_tuid = geteuid ();
^
simpleid.c:6:2: замечание: each undeclared identifier is reported only
once; declarations it appears in are ignored
simpleid.c:7:2: ошибка: «uid_tgid» undeclared (first use in this function)
uid_tgid = getegid ();
^
simpleid.c:8:30: ошибка: «uid» undeclared (first use in this function)
printf ("uid=%d, gid=%d\n", uid, gid);
^
simpleid.c:8:35: ошибка: «gid» undeclared (first use in this function)
printf ("uid=%d, gid=%d\n", uid, gid);
^
simpleid.c:9:2: ошибка: «return0» undeclared (first use in this function)
return0;
^
Получившуюся программу назовите simpleid2.c.
```

Рис. 3.4.: Компилирование файла

Выполнил программу simpleid: ./simpleid



```
005-lab..discret_sticky.pdf -- Mozilla Firefox
https://esystem.rudn.ru/pluginfile.php/1198015/mod_resource/content/2/005-lab..discret_sticky.pdf
guest@pc15:~/labs/lab05
File Правка Вид Поиск Терминал Справка
simpleid.c:6:2: замечание: each undeclared identifier is reported only
once; declarations it appears in are ignored
simpleid.c:7:2: ошибка: «uid_tgid» undeclared (first use in this function)
uid_tgid = getegid ();
^
simpleid.c:8:30: ошибка: «uid» undeclared (first use in this function)
printf ("uid=%d, gid=%d\n", uid, gid);
^
simpleid.c:8:35: ошибка: «gid» undeclared (first use in this function)
printf ("uid=%d, gid=%d\n", uid, gid);
^
simpleid.c:9:2: ошибка: «return0» undeclared (first use in this function)
return0;
^
Получившуюся программу назовите simpleid2.c.
```

Рис. 3.5.: Выполнение программы simpleid

Выполнил программу id и сравнил полученный результат с данными предыдущего пункта задания. Полученные значения id совпадают

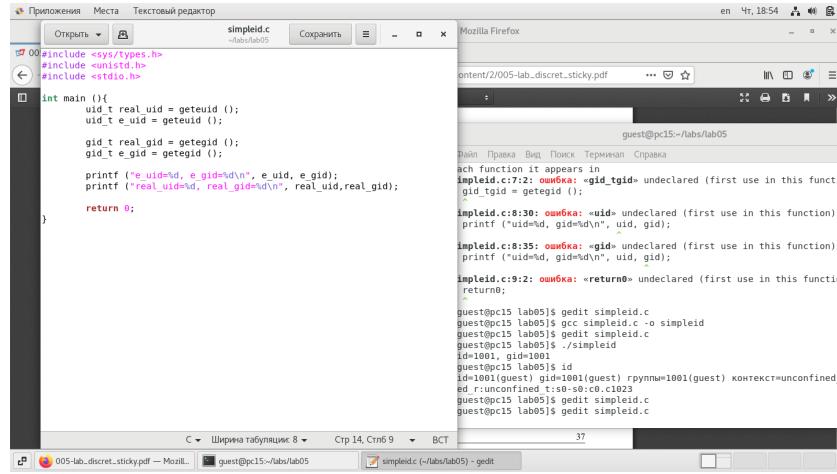


Рис. 3.6.: Выполнение программы id

Усложнил программу, добавив вывод действительных идентификаторов, получившуюся программу назвал `simpleid2.c`

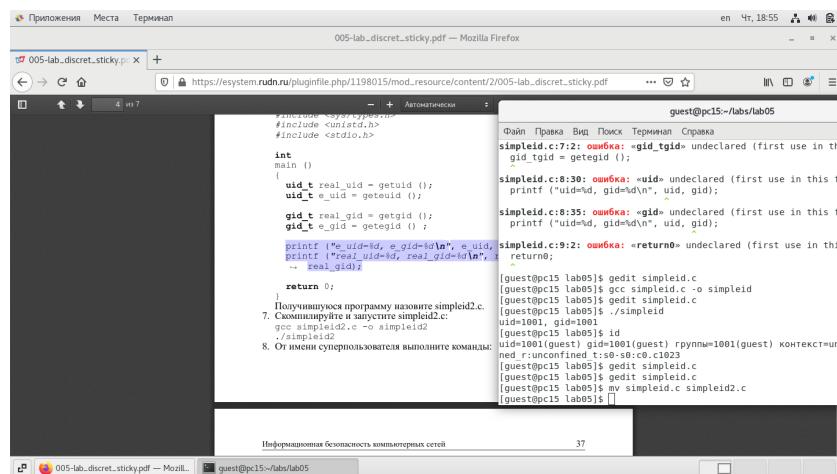


Рис. 3.7.: Усложнение программы

Скомпилировал и запустил simpleid2.c gcc simpleid2.c -o simpleid2, а затем ./simpleid2

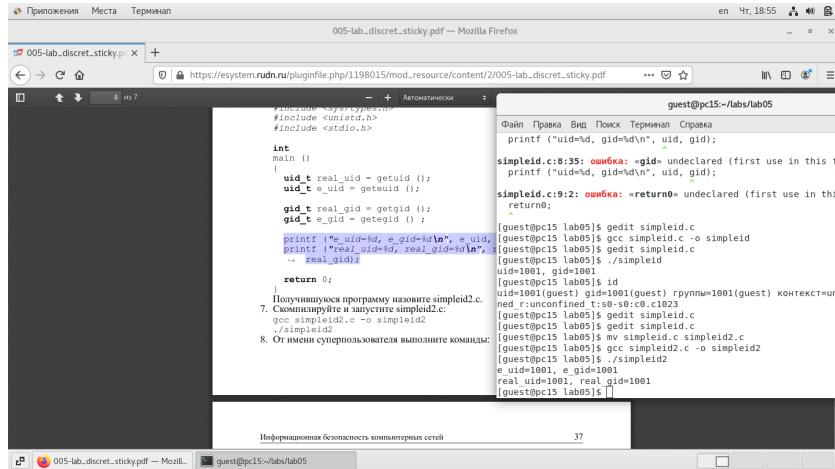


Рис. 3.8.: Компиляция и запуск

От имени суперпользователя выполнил команды: `chown root:guest /home/guest/simpleid2`, а затем `chmod u+s /home/guest/simpleid2`. Первая команда изменяет права на файл с guest на root. А затем устанавливает атрибут SetUID, который запускает программу не с правами пользователя, а с правами владельца файла. Затем выполнил проверку изменений с помощью команды `ls -l simpleid2`

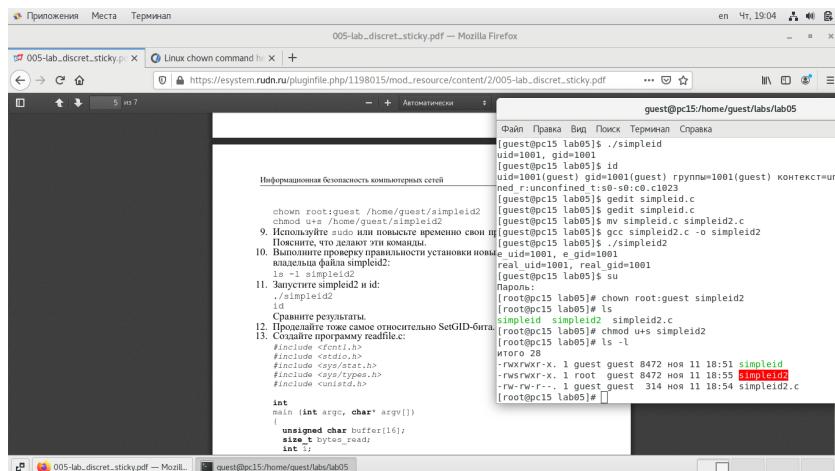


Рис. 3.9.: Добавление SetUID

Запустил simpleid2 и id: ./simpleid2, id. При данном запуску выводы совпа-

дают.

The terminal session shows the following steps:

- Creating a directory and copying files:

```
shawn root@pc15: /home/guest/simpleid2
shawn user@pc15: /home/guest/simpleid2
```

- Compiling the exploit:

```
[guest@pc15 lab05]$ gcc simpleid.c simpleid2.c
[guest@pc15 lab05]$ gcc simpleid2.c -o simpleid2
```

- Running the exploit with root privileges:

```
[guest@pc15 lab05]$ ./simpleid2
e uid=1001, e gid=1001
real uid=1001, real gid=1001
[guest@pc15 lab05]$ su
Password:
```

- Comparing results:

```
[root@pc15 lab05]# chown root:guest simpleid2
[root@pc15 lab05]# ls
simpleid simpleid2 simpleid2.c
[root@pc15 lab05]# chmod u+s simpleid2
[root@pc15 lab05]# ls -l
итого 28
-rwxr-x--x . 1 guest quest 8472 моя 11 18:51 simpleid2
-rwsr-x--x . 1 guest quest 314 моя 11 18:54 simpleid2.c
[root@pc15 lab05]# ./simpleid2
[root@pc15 lab05]#
```

- Running the exploit again to show it's still SetGID:

```
[root@pc15 lab05]# ./simpleid2
e uid=0, e gid=0
real uid=0, real gid=0
[root@pc15 lab05]# id
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:u:u
[root@pc15 lab05]#
```

Рис. 3.10.: Действия над файлом.

Проделал то же самое с атрибутом SetGID (установление прав для владеющей группы).

The terminal session shows the following steps:

- Copying files:

```
shawn root@pc15: /home/guest/simpleid2
shawn user@pc15: /home/guest/simpleid2
```

- Setting the SetGID attribute:

```
[guest@pc15 lab05]$ chmod u+s simpleid2
[guest@pc15 lab05]$
```

- Comparing results:

```
[root@pc15 lab05]# ./simpleid2
[root@pc15 lab05]# ls
simpleid simpleid2 simpleid2.c
[root@pc15 lab05]# chmod u+s simpleid2
[root@pc15 lab05]# ls -l
итого 28
-rwxr-x--x . 1 guest quest 8472 моя 11 18:51 simpleid2
-rwsr-x--x . 1 root  guest 8472 моя 11 18:55 simpleid2.c
[root@pc15 lab05]# ./simpleid2
[root@pc15 lab05]#
```

- Running the exploit again to show it's still SetGID:

```
[root@pc15 lab05]# ./simpleid2
[root@pc15 lab05]#
```

Рис. 3.11.: Установка SetGID

Запустил файл. Теперь выводы для группы различны.

```

Приложения Места Терминал
005-lab..discret_sticky.pdf — Mozilla Firefox
Linux chown command | x | +
https://esystem.rudn.ru/pluginfile.php/1198015/mod_resource/content/2/005-lab..discret_sticky.pdf
... Помощь Терминал Справка
guest@pc15:/home/guest/labs/lab05
Файл Правка Вид Поиск Терминал Справка
[root@pc15 lab05]# ls
simpleid simpleid2 simpleid2.c
[root@pc15 lab05]# chmod u+s simpleid2
[root@pc15 lab05]# ls -l
total 8
-rwxr-x--x . 1 guest guest 8472 ноя 11 18:51 simpleid2
-rwsrwxr-- . 1 root guest 8472 ноя 11 18:55 simpleid2
[root@pc15 lab05]# ./simpleid2
9. Используйте root или guest временно свои права.
Проверьте, что делает эта команда.
10. Выполните проверку правильности установки новых прав.
изменения файла simpleid:
ls -l simpleid2
11. Запустите simpleid2 и id:
./simpleid2
id
Сравните результаты.
12. Проверьте тоже самое относительно SetGID-битка.
13. Создайте программу readfile.c:
#include <fcntl.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <stropts.h>
#include <unistd.h>

int
main (int argc, char* argv[])
{
    unsigned char buffer[16];
    size_t bytes_read;
    int fd;
    int fd = open (argv[1], O_RDONLY);
    bytes_read = read (fd, buffer, bytes_read);
    for (i=0; i < bytes_read; i++)
        write (1, buffer[i], 1);
    close (fd);
    return 0;
}
14. Откомпилируйте её:
gcc readfile.c -o readfile
[root@pc15 lab05]# mv readfile.c readfile.c
15. Смените владельца у файла readfile.c или любого другого текстового файла в системе и исполните так, чтобы только суперпользователь (root) мог прочитать его, а разные пользователи не могли.
16. Проверьте, что пользователь guest не может прочитать файл readfile.c.
17. Смените у программы readfile владельца и установите SetGID-бит.
18. Проверьте, может ли пользователь guest прочитать файл readfile.c?
19. Проделайте то же самое, но с программой readfile прочитать файла /etc/shadow.
Организуйте полученный результат и ваши объяснения в отчёте.

```

Рис. 3.12.: Запуск программы с установленным SetGID

Создал программу `readfile.c`

```

Приложения Места Терминал
005-lab..discret_sticky.pdf — Mozilla Firefox
Linux chown command | x | +
https://esystem.rudn.ru/pluginfile.php/1198015/mod_resource/content/2/005-lab..discret_sticky.pdf
... Помощь Терминал Справка
guest@pc15:/home/guest/labs/lab05
Файл Правка Вид Поиск Терминал Справка
12. Проверьте тоже самое относительно SetGID-битка.
13. Создайте программу readfile.c:
#include <fcntl.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <stropts.h>
#include <unistd.h>

int
main (int argc, char* argv[])
{
    unsigned char buffer[16];
    size_t bytes_read;
    int fd;
    int fd = open (argv[1], O_RDONLY);
    bytes_read = read (fd, buffer, bytes_read);
    for (i=0; i < bytes_read; i++)
        write (1, buffer[i], 1);
    close (fd);
    return 0;
}
14. Откомпилируйте её:
gcc readfile.c -o readfile
[root@pc15 lab05]# mv readfile.c readfile.c
15. Смените владельца у файла readfile.c или любого другого текстового файла в системе и исполните так, чтобы только суперпользователь (root) мог прочитать его, а разные пользователи не могли.
16. Проверьте, что пользователь guest не может прочитать файл readfile.c.
17. Смените у программы readfile владельца и установите SetGID-бит.
18. Проверьте, может ли пользователь guest прочитать файл readfile.c?
19. Проделайте то же самое, но с программой readfile прочитать файла /etc/shadow.
Организуйте полученный результат и ваши объяснения в отчёте.

```

Рис. 3.13.: Создание программы `readfile.c`

Откомпилировал программу: `gcc readfile.c -o readfile`

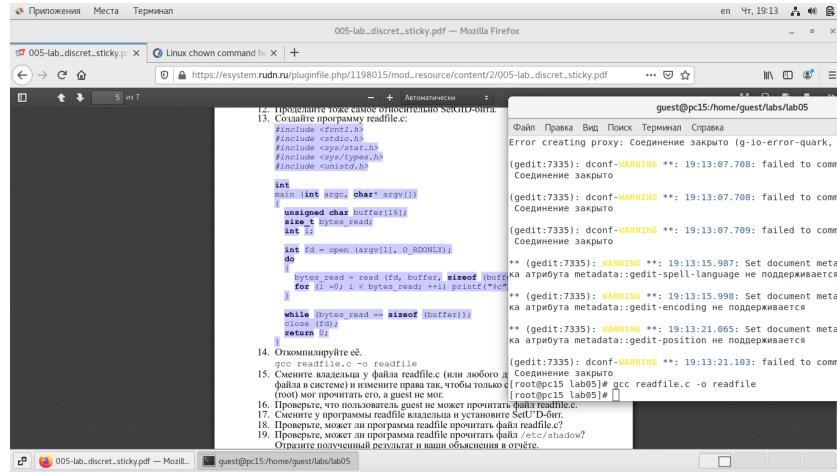


Рис. 3.14.: Компиляция программы

Сменил владельца у файла `readfile.c` и изменил права так, чтобы только суперпользователь(`root`) мог прочитать его, а `guest` не мог.

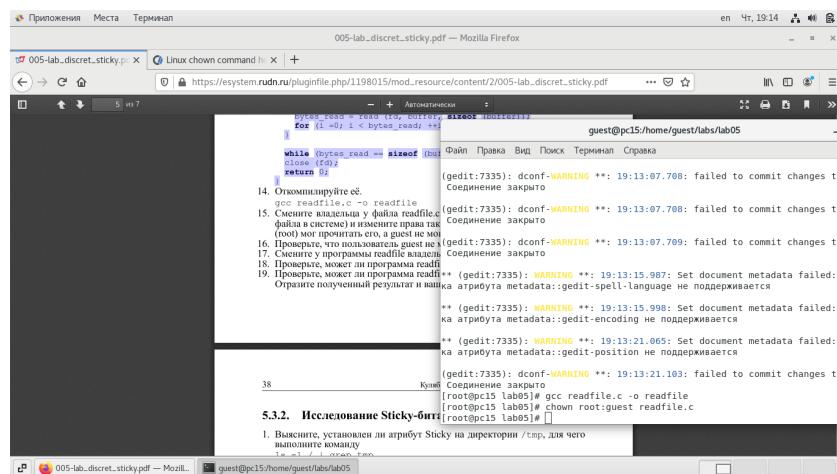


Рис. 3.15.: Смена владельца у файла `readfile.c`

Проверил, что пользователь guest не может прочитать файл readfile.c

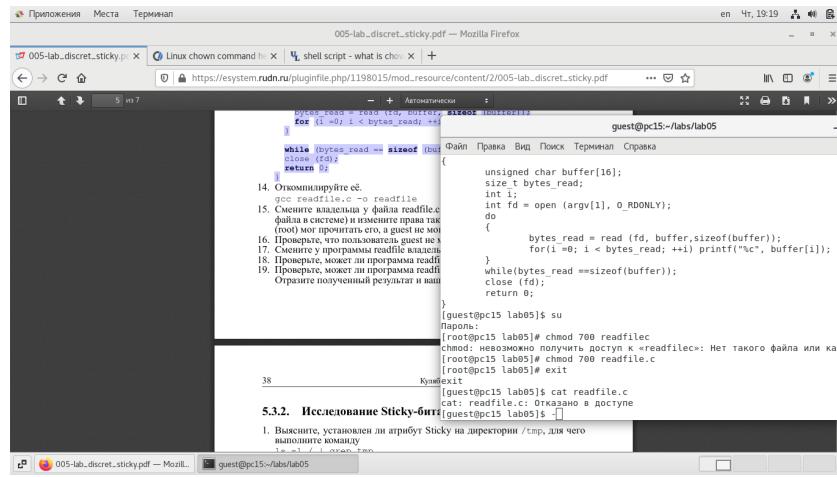


Рис. 3.16.: Проверка

Сменил у программы readfile владельца и установил SetU'D-бит.

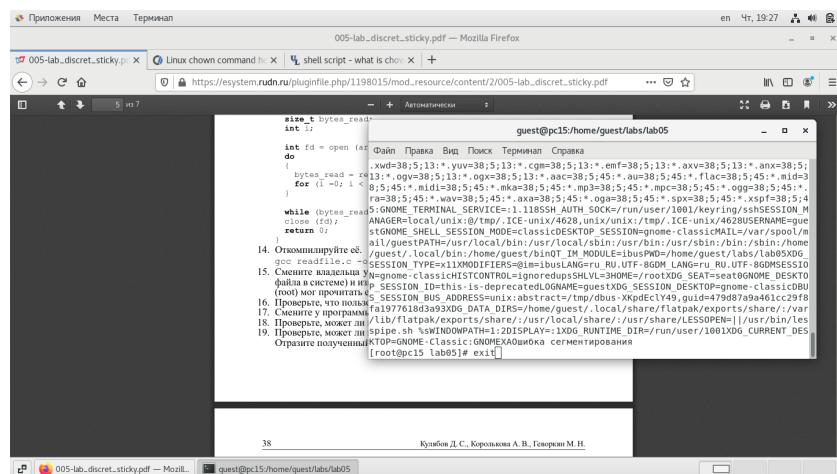


Рис. 3.17.: Установил SetU'D бит

Проверил, может ли программа readfile прочитать файл readfile.c. Может.

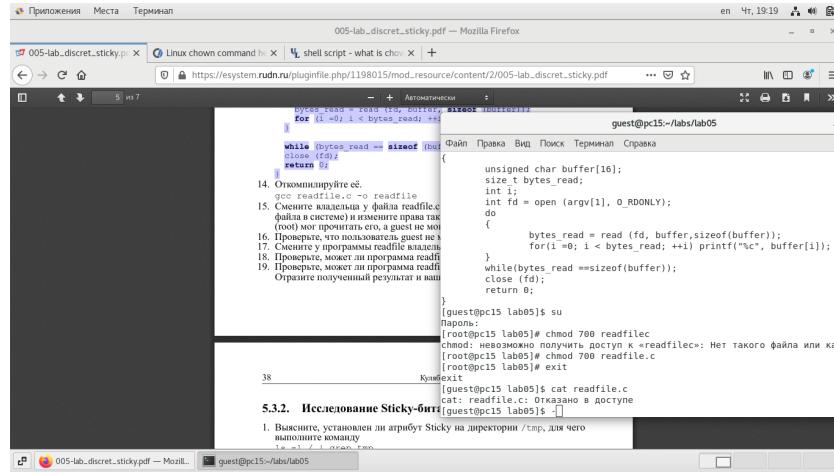


Рис. 3.18.: Проверка

Проверил, может ли программа `readfile` прочитать файл `/etc/shadow`. Может

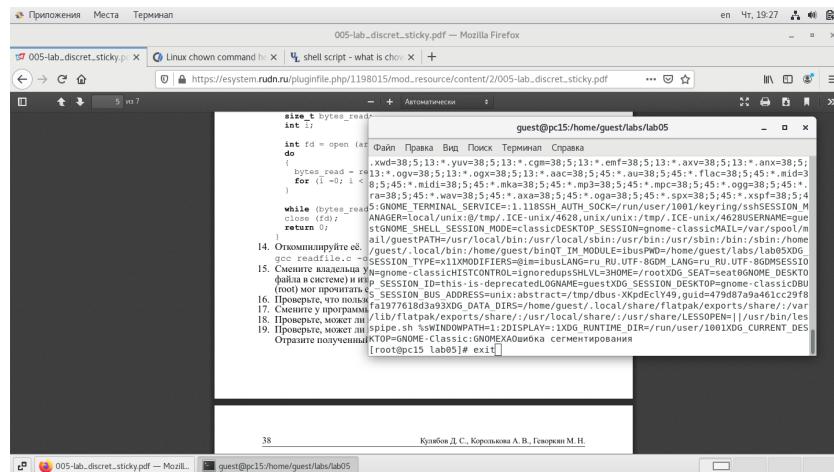


Рис. 3.19.: Проверка

Исследование Sticky-бита. Узнал, установлен ли атрибут Sticky на директории /tmp, для чего выполнил команду `ls -l / | grep tmp`

Рис. 3.20.: Sticky бит на папке tmp

От имени пользователя guest создал файл file01.txt в директории /tmp со словом test echo "test" > /tmp/file01.txt

Рис. 3.21.: Создание файла

Просмотрел атрибуты у только что созданного файла и разрешил чтение и запись для категории пользователей «все остальные»: ls -l /tmp/file01.txt, chmod o+rwx /tmp/file01.txt, ls -l /tmp/file01.txt

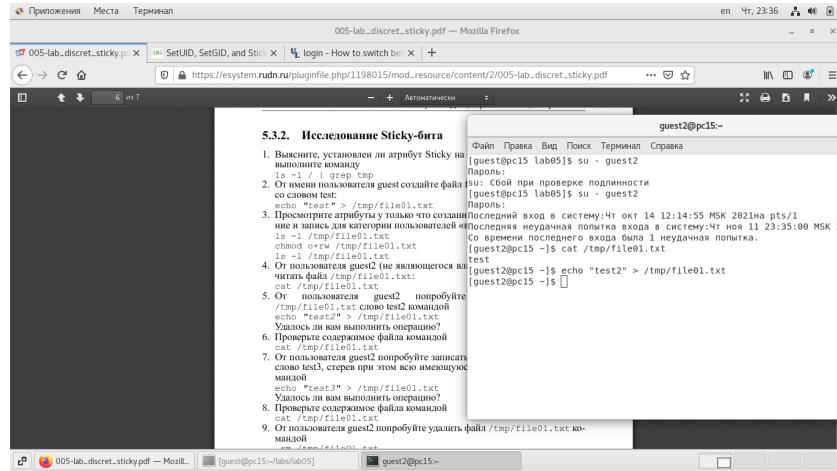


Рис. 3.22.: Просмотр атрибутов

От пользователя guest2 (не являющегося владельцем) попробовал прочитать файл /tmp/file01.txt: cat /tmp/file01.txt

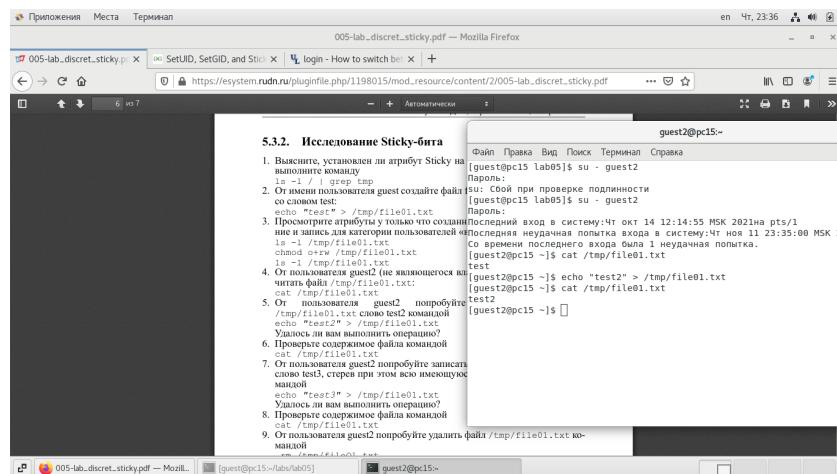


Рис. 3.23.: Чтение файла

От пользователя guest2 попробовал записать в файл /tmp/file01.txt слово test3, стерев при этом всю имеющуюся в файле информацию командой echo "test3" > /tmp/file01.txt

Проверил содержимое файла командой cat /tmp/file01.txt

От пользователя guest2 попробовал дозаписать в файл /tmp/file01.txt слово

test2 командой echo "test2" >> /tmp/file01.txt

Проверил содержимое файла командой cat /tmp/file01.txt

The screenshot shows a terminal window titled 'guest2@pc15:'. The user has run the command 'cat /tmp/file01.txt' and is prompted for permission to write to the file. The terminal shows the following interaction:

```
[guest2@pc15 ~]$ cat /tmp/file01.txt
[guest2@pc15 ~]$ ls -l /tmp/file01.txt
[guest2@pc15 ~]$ chmod +t /tmp/file01.txt
[guest2@pc15 ~]$ cat /tmp/file01.txt
[guest2@pc15 ~]$ echo "test2" >> /tmp/file01.txt
[guest2@pc15 ~]$ ls -l /tmp/file01.txt
[guest2@pc15 ~]$ rm /tmp/file01.txt
[guest2@pc15 ~]$
```

Рис. 3.24.: Запись и дозапись

От пользователя guest2 попробовал удалить файл /tmp/file01.txt командой rm /tmp/file01.txt Файл удалить не удалось.

The screenshot shows a terminal window titled 'guest2@pc15:'. The user has run the command 'rm /tmp/file01.txt' and is prompted for permission to delete the file. The terminal shows the following interaction:

```
[guest2@pc15 ~]$ rm /tmp/file01.txt
[guest2@pc15 ~]$ ls -l /tmp/file01.txt
[guest2@pc15 ~]$
```

Рис. 3.25.: Попытка удалить файл

Повысил свои права до суперпользователя следующей командой su - и выполнил после этого команду, снимающую атрибут t (Sticky-bit) с директории /tmp: chmod -t /tmp

Рис. 3.26.: Снятие Sticky атрибута

Повысил свои права до суперпользователя и вернул атрибут t на директорию /tmp: su -, chmod +t /tmp, exit

Рис. 3.27.: Выход из суперпользователя

## **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.