

Отчет по лабораторной работе 5

Дисциплина: Информационная безопасность

Савченков Дмитрий Андреевич, НПИбд-02-18

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	15
	Список литературы	16

List of Figures

2.1	Программа <i>simpleid.c</i>	6
2.2	Компиляция и выполнение программы <i>simpleid</i>	7
2.3	Программа <i>simpleid2.c</i>	7
2.4	Компиляция и выполнение программы <i>simpleid2</i>	8
2.5	Смена пользователя. Установка SetUID-бита. Выполнение программы <i>simpl</i>	8
2.6	Установка SetGID-бита. Выполнение программы <i>simpl</i>	9
2.7	Программа <i>readfile.c</i>	9
2.8	Работа с программой <i>readfile.c</i>	9
2.9	Запрет на чтение программы <i>readfile.c</i> для <i>guest</i>	10
2.10	Установка SetUID-бита на программу <i>readfile</i>	10
2.11	Программа <i>readfile</i> читает <i>readfile.c</i>	10
2.12	Программа <i>readfile</i> читает <i>/etc/shadow</i>	11
2.13	Исследование Sticky-бита от имени <i>guest</i>	11
2.14	Работа с <i>file01.txt</i> от имени <i>guest2</i> при наличии Sticky-бита	12
2.15	Снятие Sticky-бита с <i>/tmp</i>	13
2.16	Работа с <i>file01.txt</i> от имени <i>guest2</i> без Sticky-бита	13
2.17	Возвращение Sticky-бита на <i>/tmp</i>	14

List of Tables

1 Цель работы

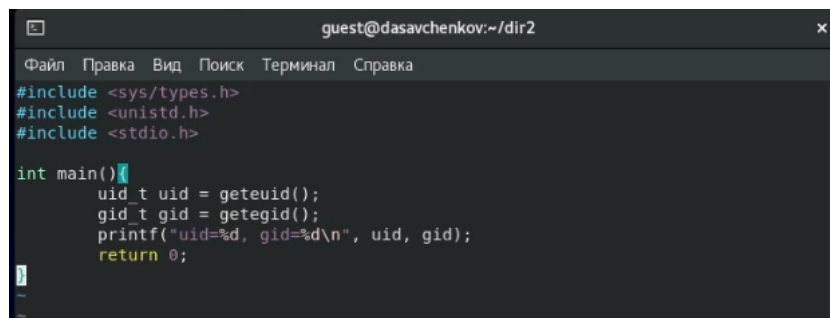
Изучение механизмов изменения идентификаторов, применения SetUID- и Sticky-битов. Получение практических навыков работы в консоли с дополнительными атрибутами. Рассмотрение работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

2 Выполнение лабораторной работы

1. Создание программы

1.1. Вошел в систему от имени пользователя guest.

1.2. Создал программу *simpleid.c* по шаблону из методички. (рис. 2.1)



```
guest@dasavchenkov:~/dir2
Файл Правка Вид Поиск Терминал Справка
#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;
}
```

Figure 2.1: Программа *simpleid.c*

1.3. Скомпилировал программу и убедилась, что файл программы создан: `gcc simpleid.c -o simpl.` (рис. 2.2)

1.4. Выполнил программу *simpleid*: `./simpl.` (рис. 2.2)

1.5. Выполнил системную программу *id*: `id`. (рис. 2.2) Полученный мной результат совпадает с данными предыдущего пункта задания.

```

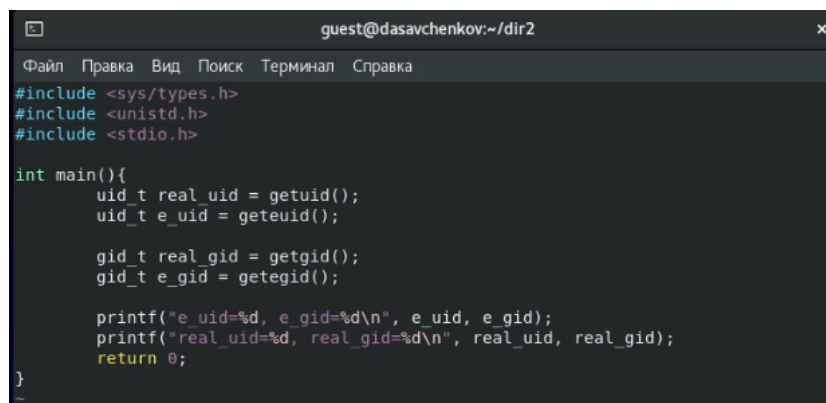
[guest@dasavchenkov dir2]$ cat simpleid.c
[guest@dasavchenkov dir2]$ vim simpleid.c
[guest@dasavchenkov dir2]$ cat simpleid.c
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>

int main(){
    uid_t uid = getuid();
    gid_t gid = getgid();
    printf("uid=%d, gid=%d", uid, gid);
    return 0;
}
[guest@dasavchenkov dir2]$ gcc simpleid.c -o simpl
bash: gcc: команда не найдена...
[guest@dasavchenkov dir2]$ gcc simpleid.c -o simpl
[guest@dasavchenkov dir2]$ ./simpl
uid=1001, gid=1001[guest@dasavchenkov dir2]$ id
uid=1001(guest) gid=1001(guest) rpyнпы=1001(guest) контекст=unconfined_u:unconfi
ned_r:unconfined_t:s0-s0:c0.c1023
[guest@dasavchenkov dir2]$

```

Figure 2.2: Компиляция и выполнение программы *simpleid*

1.6. Усложнил программу, добавив вывод действительных идентификаторов согласно шаблону из методички. Для получившейся программы оставил название *simpleid.c*. (рис. 2.3)



```

guest@dasavchenkov:~/dir2
Файл Правка Вид Поиск Терминал Справка
#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;
}

```

Figure 2.3: Программа *simpleid2.c*

1.7. Скомпилировал и запустил *simpleid.c*: `gcc simpleid.c -o simpl` и `./simpl`. (рис. 2.4)

```
[guest@dasavchenkov dir2]$ vim simpleid.c
[guest@dasavchenkov dir2]$ cat simpleid.c
#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;
}
[guest@dasavchenkov dir2]$ gcc simpleid.c -o simpl
[guest@dasavchenkov dir2]$ ./simpl
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
```

Figure 2.4: Компиляция и выполнение программы simpleid2

1.8. От имени суперпользователя выполнил команды: `chown root:guest /home/guest/simpl` и `chmod u+s /home/guest/simpl`. (рис. 2.5)

1.9. Повысил временно свои права с помощью `su`. (рис. 2.5) Первая команда меняет владельца файла, а вторая добавляет SetUID-бит.

1.10. Выполнил проверку правильности установки новых атрибутов и смены владельца файла `simpl`: `ls -l simpl`. (рис. 2.5)

1.11. Запустил `simpl` и `id`: `./simpl` и `id`. (рис. 2.5)

```
[guest@dasavchenkov dir2]$ su
Пароль:
[root@dasavchenkov dir2]# chown root:guest /home/guest/dir2/simpl
[root@dasavchenkov dir2]# chmod u+s /home/guest/dir2/simpl
[root@dasavchenkov dir2]# ls -l simpl
-rwsrwxr-x. 1 root guest 17648 ноя 12 16:22 simpl
[root@dasavchenkov dir2]# su guest
[guest@dasavchenkov dir2]$ ./simpl
e_uid=0, e_gid=1001
real_uid=1001, real_gid=1001
[guest@dasavchenkov dir2]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfi
ned_r:unconfined_t:s0-s0:c0.c1023
[guest@dasavchenkov dir2]$
```

Figure 2.5: Смена пользователя. Установка SetUID-бита. Выполнение программы simpl

1.12. Проделал то же самое относительно SetGID-бита. (рис. 2.6)

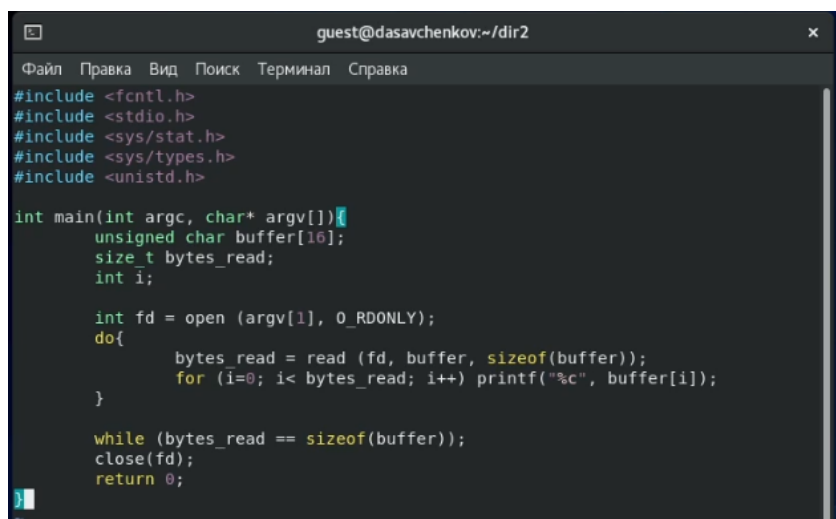

```

[root@dasavchenkov dir2]# chmod g+s /home/guest/dir2/simpl
[root@dasavchenkov dir2]# ls -l simpl
-rwsrwsr-x. 1 root guest 17648 ноя 12 16:22 simpl
[root@dasavchenkov dir2]# su guest
[guest@dasavchenkov dir2]$ ./simpl
e_uid=0, e_gid=1001
real_uid=1001, real_gid=1001
[guest@dasavchenkov dir2]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@dasavchenkov dir2]$ su
Пароль:
[root@dasavchenkov dir2]# ./simpl
e_uid=0, e_gid=1001
real_uid=0, real_gid=0
[root@dasavchenkov dir2]# id
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@dasavchenkov dir2]#

```

Figure 2.6: Установка SetGID-бита. Выполнение программы simpl

1.13. Создал программу readfile.c по шаблону из методички. (рис. 2.7)



```

guest@dasavchenkov:~/dir2
Файл Правка Вид Поиск Терминал Справка
#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;
}

```

Figure 2.7: Программа *readfile.c*

1.14. Откомпилировал её: gcc readfile.c -o readfile. (рис. 2.8)

1.15. Сменил владельца у файла readfile.c и изменил права так, чтобы только суперпользователь (root) мог прочитать его, а guest не мог. (рис. 2.8)

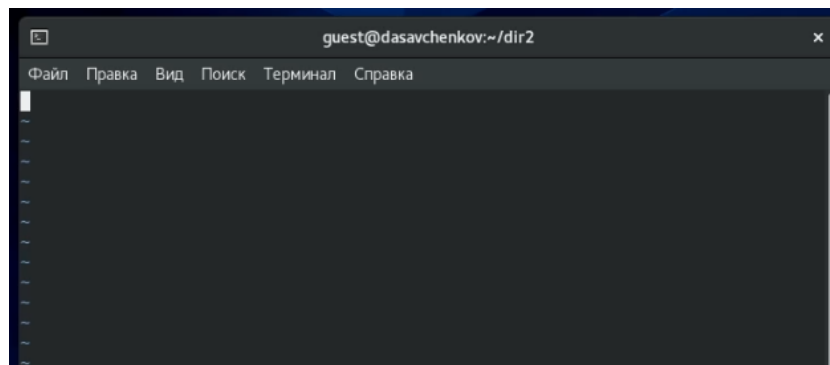
```

[root@dasavchenkov dir2]# chown root:guest /home/guest/dir2/readfile.c
[root@dasavchenkov dir2]# chmod 700 /home/guest/dir2/readfile.c
[root@dasavchenkov dir2]# su guest
[guest@dasavchenkov dir2]$ vim

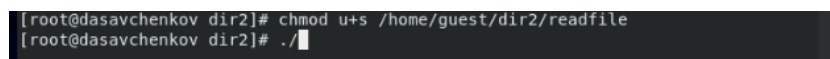
```

Figure 2.8: Работа с программой *readfile.c*

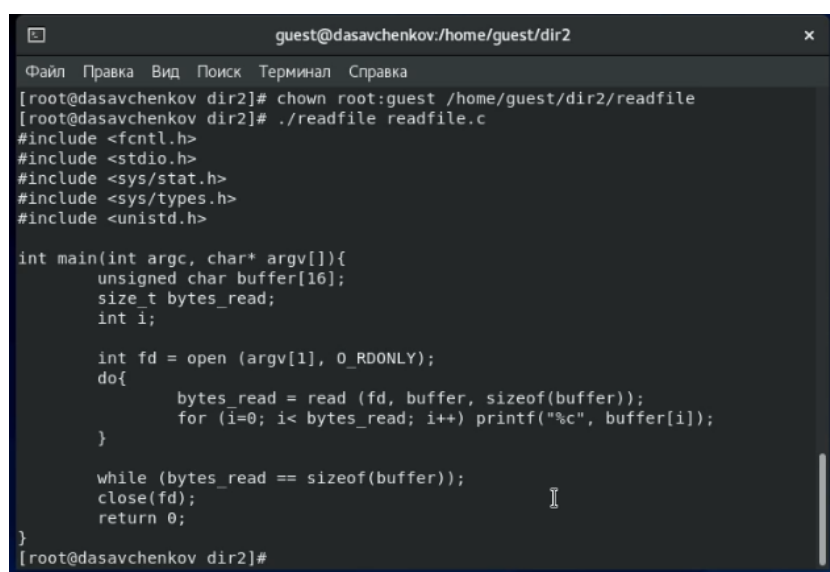
1.16. Проверил, что пользователь guest не может прочитать файл readfile.c. (рис. 2.9)



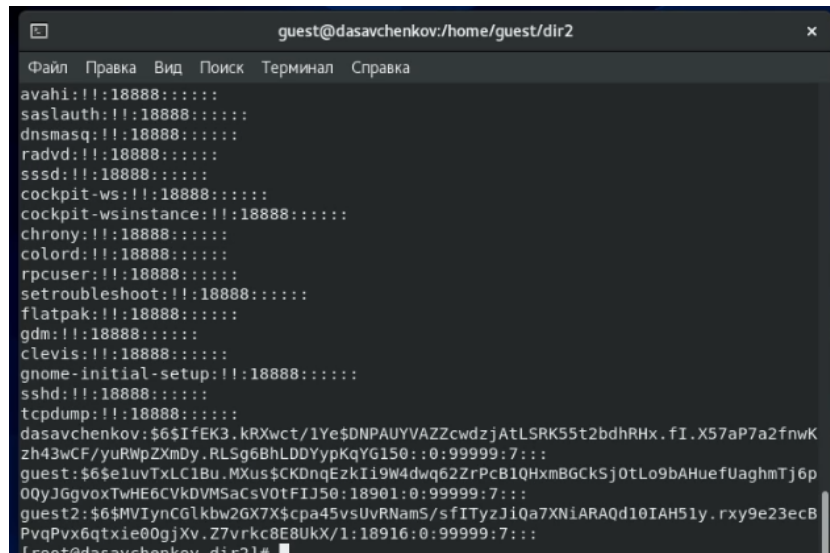
1.17. Сменил у программы `readfile` владельца (рис. 2.8) и установил SetUID-бит (рис. 2.10).



1.18. Проверил, может ли программа `readfile` прочитать файл `readfile.c`. (рис. 2.11)



1.19. Проверил, может ли программа readfile прочитать файл */etc/shadow*. (рис. 2.12)



```
guest@dasavchenkov:/home/guest/dir2
Файл Правка Вид Поиск Терминал Справка
avahi:!:18888:!:~:
saslauth:!:18888:!:~:
dnsmasq:!:18888:!:~:
raddvd:!:18888:!:~:
sssd:!:18888:!:~:
cockpit-ws:!:18888:!:~:
cockpit-wsinstance:!:18888:!:~:
chrony:!:18888:!:~:
colord:!:18888:!:~:
rpcuser:!:18888:!:~:
setroubleshoot:!:18888:!:~:
flatpak:!:18888:!:~:
gdm:!:18888:!:~:
clemis:!:18888:!:~:
gnome-initial-setup:!:18888:!:~:
sshd:!:18888:!:~:
tcpdump:!:18888:!:~:
dasavchenkov:$6$IfeK3.kRXwct/1Ye$DNPAUYVAZZcWdzjAtLSRK55t2bdhRHx.fI.X57aP7a2fnwK
zh43wCF/yuRWPZxmDy.RLSg6BhLDDYypKqYG150::0:99999:7:::
guest:$6$e1uvTxLC1Bu.MXus$CKDnqEzKiI9W4dwq62ZrPcB1QHxmBGckSJ0tLo9bAHuefUaghmTj6p
00yJGgvoxTwHE6CVKdVMSaCsV0tFIJ50:18901:0:99999:7:::
guest2:$6$MVIynCGlkbw2GX7X$cpa45vsUvRNamS/sfITyzJiQa7XNiARAQd10IAH51y.rxy9e23ecB
PvqPx6qtXie00gjXv.Z7vrkc8E8UKX/1:18916:0:99999:7:::
[root@dasavchenkov dir2]#
```

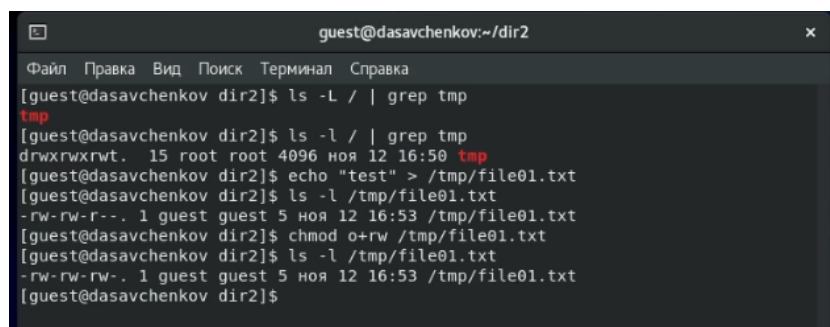
Figure 2.12: Программа readfile читает /etc/shadow

2. Исследование Sticky-бита

2.1. Выяснил, установлен ли атрибут Sticky на директории /tmp, для чего выполнил команду: `ls -l / | grep tmp`. (рис. 2.13)

2.2. От имени пользователя guest создал файл file01.txt в директории /tmp со словом test: `echo "test" > /tmp/file01.txt`. (рис. 2.13)

2.3. Просмотрел атрибуты у только что созданного файла и разрешил чтение и запись для категории пользователей «все остальные»: `ls -l /tmp/file01.txt`, `chmod o+rw /tmp/file01.txt` и `ls -l /tmp/file01.txt`. (рис. 2.13)



```
guest@dasavchenkov:~/dir2
Файл Правка Вид Поиск Терминал Справка
[guest@dasavchenkov dir2]$ ls -L / | grep tmp
tmp
[guest@dasavchenkov dir2]$ ls -l / | grep tmp
drwxrwxrwt. 15 root root 4096 ноя 12 16:50 tmp
[guest@dasavchenkov dir2]$ echo "test" > /tmp/file01.txt
[guest@dasavchenkov dir2]$ ls -l /tmp/file01.txt
-rw-rw-r--. 1 guest guest 5 ноя 12 16:53 /tmp/file01.txt
[guest@dasavchenkov dir2]$ chmod o+rw /tmp/file01.txt
[guest@dasavchenkov dir2]$ ls -l /tmp/file01.txt
-rw-rw-rw-. 1 guest guest 5 ноя 12 16:53 /tmp/file01.txt
[guest@dasavchenkov dir2]$
```

Figure 2.13: Исследование Sticky-бита от имени guest

2.4. От пользователя guest2 (не являющегося владельцем) попробовал прочитать файл /tmp/file01.txt: `cat /tmp/file01.txt`. (рис. 2.14)

2.5. От пользователя `guest2` попробовал дозаписать в файл `/tmp/file01.txt` слово `test2` командой: `echo "test2" >> /tmp/file01.txt`. (рис. 2.14) Операция прошла успешно.

2.6. Проверил содержимое файла командой: `cat /tmp/file01.txt`. (рис. 2.14)

2.7. 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`. (рис. 2.14) Операция прошла успешно.

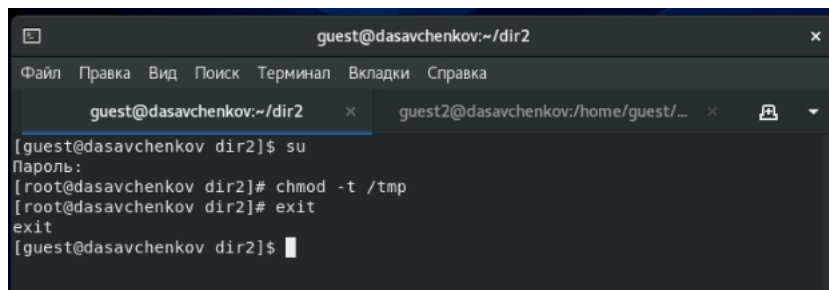
2.8. Проверил содержимое файла командой: `cat /tmp/file01.txt`. (рис. 2.14)

2.9. От пользователя `guest2` попробовал удалить файл `/tmp/file01.txt` командой: `rm /tmp/file01.txt`. (рис. 2.14) Операция была не позволена.

Figure 2.14: Работа с `file01.txt` от имени `guest2` при наличии Sticky-бита

2.10. Повысил свои права до суперпользователя следующей командой: `su -`, и выполнил после этого команду, снимающую атрибут `t` (Sticky-бит) с директории `/tmp`: `chmod -t /tmp`. (рис. 2.15)

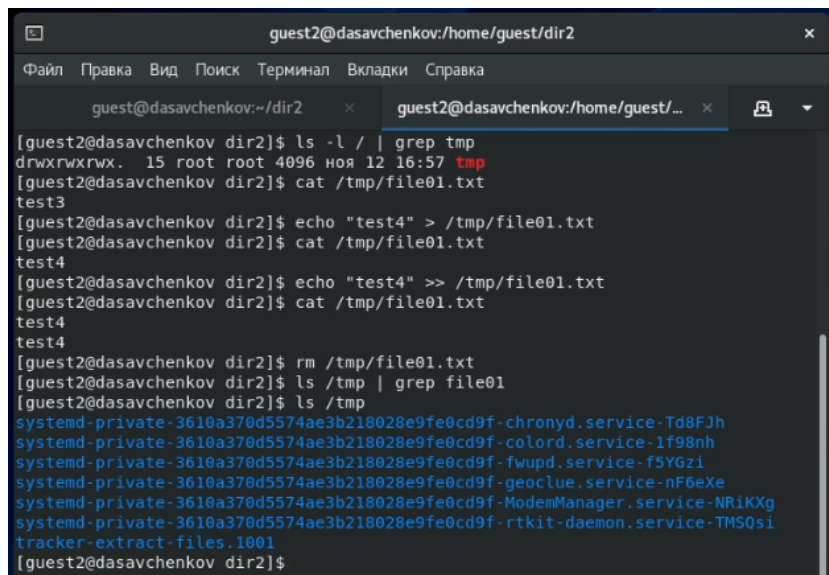
2.11. Покинул режим суперпользователя командой: `exit`. (рис. 2.15)



```
guest@dasavchenkov:~/dir2
[guest@dasavchenkov dir2]$ su
Пароль:
[root@dasavchenkov dir2]# chmod -t /tmp
[root@dasavchenkov dir2]# exit
exit
[guest@dasavchenkov dir2]$
```

Figure 2.15: Снятие Sticky-бита с */tmp*

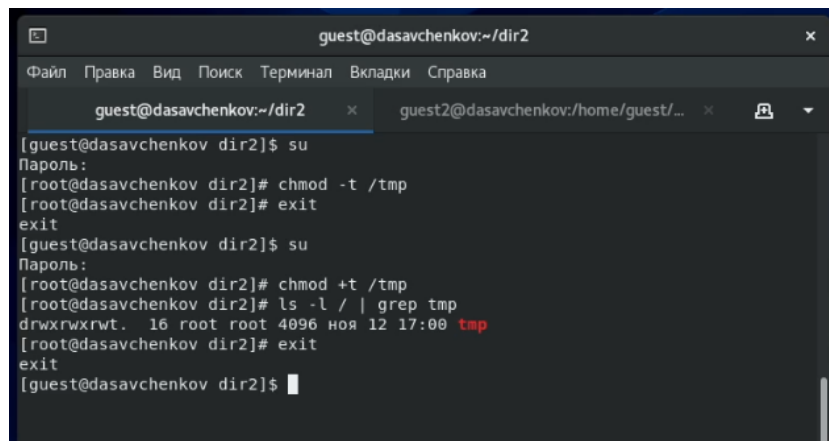
- 2.12. От пользователя *guest2* проверил, что атрибута *t* у директории */tmp* нет:
`ls -l / | grep tmp`. (рис. 2.16)
- 2.13. Повторил предыдущие шаги. (рис. 2.16) Теперь удалось удалить файл.



```
guest2@dasavchenkov:/home/guest/dir2
[guest2@dasavchenkov dir2]$ ls -l / | grep tmp
drwxrwxrwx. 15 root root 4096 ноя 12 16:57 tmp
[guest2@dasavchenkov dir2]$ cat /tmp/file01.txt
test3
[guest2@dasavchenkov dir2]$ echo "test4" > /tmp/file01.txt
[guest2@dasavchenkov dir2]$ cat /tmp/file01.txt
test4
[guest2@dasavchenkov dir2]$ echo "test4" >> /tmp/file01.txt
[guest2@dasavchenkov dir2]$ cat /tmp/file01.txt
test4
test4
[guest2@dasavchenkov dir2]$ rm /tmp/file01.txt
[guest2@dasavchenkov dir2]$ ls /tmp | grep file01
[guest2@dasavchenkov dir2]$ ls /tmp
systemd-private-3610a370d5574ae3b218028e9fe0cd9f-chrond.service-Td8FJh
systemd-private-3610a370d5574ae3b218028e9fe0cd9f-colord.service-1f98nh
systemd-private-3610a370d5574ae3b218028e9fe0cd9f-fwupd.service-f5YGzi
systemd-private-3610a370d5574ae3b218028e9fe0cd9f-geoclue.service-nF6eXe
systemd-private-3610a370d5574ae3b218028e9fe0cd9f-ModemManager.service-NRiKXg
systemd-private-3610a370d5574ae3b218028e9fe0cd9f-rtkit-daemon.service-TMSQsi
tracker-extract-files.1801
[guest2@dasavchenkov dir2]$
```

Figure 2.16: Работа с *file01.txt* от имени *guest2* без Sticky-бита

- 2.14. Да, мне удалось удалить файл от имени пользователя, не являющегося его владельцем.
- 2.15. Повысил свои права до суперпользователя и вернул атрибут *t* на директорию */tmp*: `su -, chmod +t /tmp` и `exit`. (рис. 2.17)



The image shows a terminal window titled 'guest@dasavchenkov:~/dir2'. It contains two tabs: 'guest@dasavchenkov:~/dir2' (active) and 'guest2@dasavchenkov:/home/guest/...'. The terminal output shows a sequence of commands and their results:

```
[guest@dasavchenkov dir2]$ su
Пароль:
[root@dasavchenkov dir2]# chmod -t /tmp
[root@dasavchenkov dir2]# exit
exit
[guest@dasavchenkov dir2]$ su
Пароль:
[root@dasavchenkov dir2]# chmod +t /tmp
[root@dasavchenkov dir2]# ls -l / | grep tmp
drwxrwxrwt. 16 root root 4096 ноя 12 17:00 tmp
[root@dasavchenkov dir2]# exit
exit
[guest@dasavchenkov dir2]$
```

Figure 2.17: Возвращение Sticky-бита на */tmp*

3 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.

Список литературы

1. Кулябов Д. С., Королькова А. В., Геворкян М. Н. Информационная безопасность компьютерных сетей. Лабораторная работа № 5. Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов