

Отчет по лабораторной работе №5

**Дискреционное разграничение прав в Linux. Исследование влияния
дополнительных атрибутов**

Динькиев Валерий

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	14

List of Tables

List of Figures

2.1	Программа simpleid.c	6
2.2	Компиляция и выполнение программы simpleid	7
2.3	Отредактированная программа simpleid.c	7
2.4	Компиляция и выполнение программы simpleid	8
2.5	Смена пользователя, установка SetUID-бита	8
2.6	Выполнение программы simpleid и команды id	8
2.7	Программа readfile.c	9
2.8	Работа с программой readfile.c	9
2.9	Проверка файла на чтение	10
2.10	Проверка чтения файла readfile.c и установка SetUid-бита	10
2.11	Проверка чтения файла /etc/shadow	11
2.12	Исследование Sticky-бита от имени guest	11
2.13	Работа с file01.txt от имени guest2 при наличии Sticky-бита	12
2.14	Снятие Sticky-бита с директории /tmp	12
2.15	Работа с file01.txt от имени guest2 без Sticky-бита	13
2.16	Возвращение Sticky-бита на /tmp	13

1 Цель работы

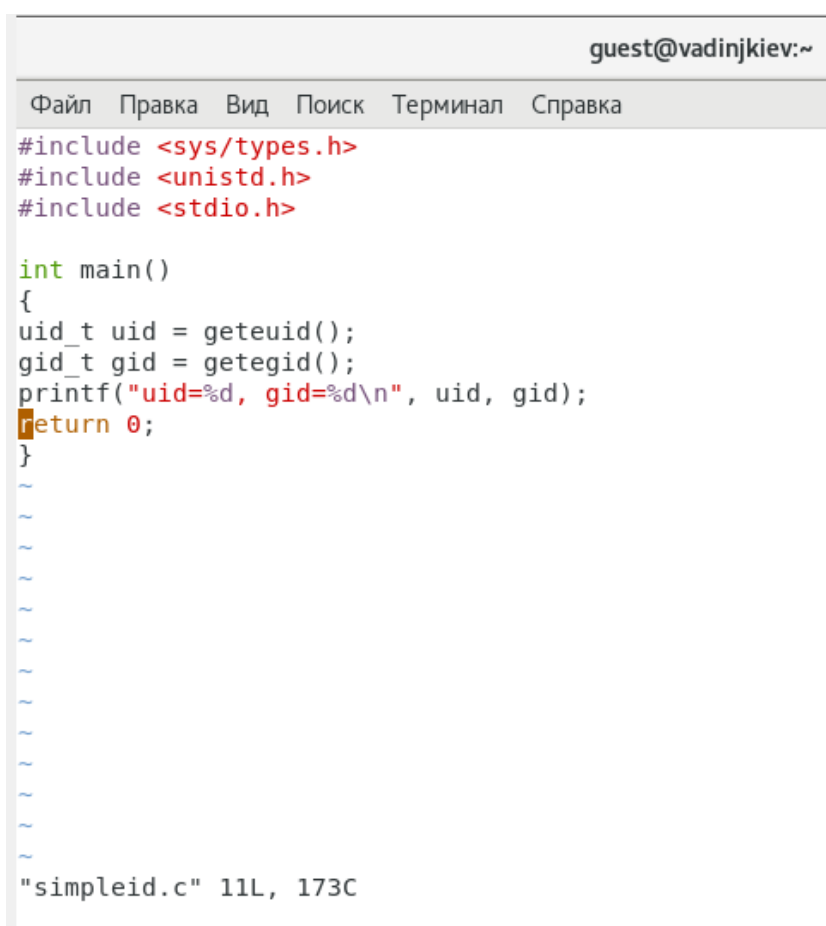
Изучение механизмов изменения идентификаторов, применения SetUID- и Sticky-битов. Получение практических навыков работы в консоли с дополнительными атрибутами. Рассмотрение работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

2 Выполнение лабораторной работы

1. Создание программы

1.1. Вошел в систему от имени пользователя guest.

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



```
guest@vadinjkiev:~  
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;  
}  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
"simpleid.c" 11L, 173C
```

Figure 2.1: Программа simpleid.c

1.3. Скомпилировал программу и убедился, что файл программы создан (рис. 2.2)

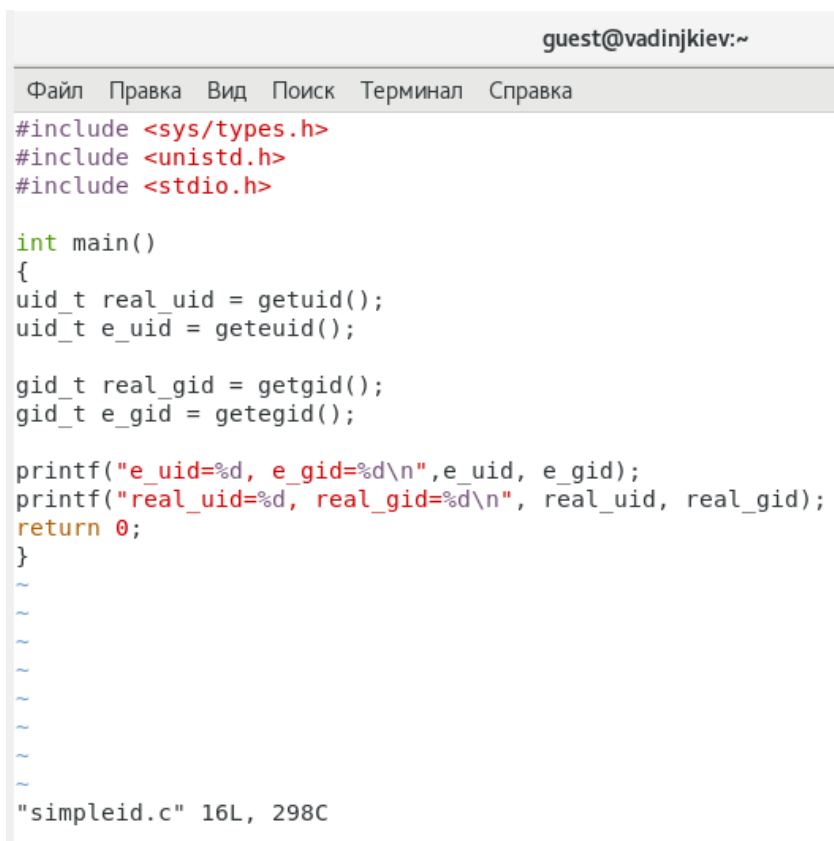
1.4. Выполнил программу simpleid: (рис. 2.2)

1.5. Выполнил системную программу id (рис. 2.2)

```
[guest@vadinjkiiev ~]$ gcc simpleid.c -o simpleid
[guest@vadinjkiiev ~]$ vim simpleid.c
[guest@vadinjkiiev ~]$ ./simpleid
uid=1001, gid=1001
[guest@vadinjkiiev ~]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@vadinjkiiev ~]$
```

Figure 2.2: Компиляция и выполнение программы simpleid

1.6. Усложнил программу, добавив вывод действительных идентификаторов. Я отредактировал программу simpleid.c (рис. 2.3)



```
guest@vadinjkiiev:~
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;
}

"simpleid.c" 16L, 298C
```

Figure 2.3: Отредактированная программа simpleid.c

1.7. Скомпилировал и запустил simpleid2.c (рис. 2.4)

```
[guest@vadinjkiiev ~]$ vim simpleid.c
[guest@vadinjkiiev ~]$ gcc simpleid.c -o simpleid
[guest@vadinjkiiev ~]$ ./simpleid
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
[guest@vadinjkiiev ~]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfined_r:u
nconfined_t:s0-s0:c0.c1023
[guest@vadinjkiiev ~]$
```

Figure 2.4: Компиляция и выполнение программы simpleid

1.8. От имени суперпользователя выполнил команды (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 simpleid2 (рис. 2.5) (рис. 2.6)

1.11. Запустил simpleid2 и id (рис. 2.5). Результаты совпадают.

```
[guest@vadinjkiiev ~]$ su
Пароль:
[root@vadinjkiiev guest]# chown root:guest /home/guest/simpleid
[root@vadinjkiiev guest]# chmod u+s /home/guest/simpleid
[root@vadinjkiiev guest]# ls -l simpleid
-rwsrwxr-x. 1 root guest 8576 ноя 13 18:19 simpleid
```

Figure 2.5: Смена пользователя, установка SetUID-бита

```
[root@vadinjkiiev guest]# ./simpleid
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@vadinjkiiev guest]# id
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:unconfined_t:
s0-s0:c0.c1023
[root@vadinjkiiev guest]#
```

Figure 2.6: Выполнение программы simpleid и команды id

1.12. Проделал то же самое относительно SetGID-бита

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



```
#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;

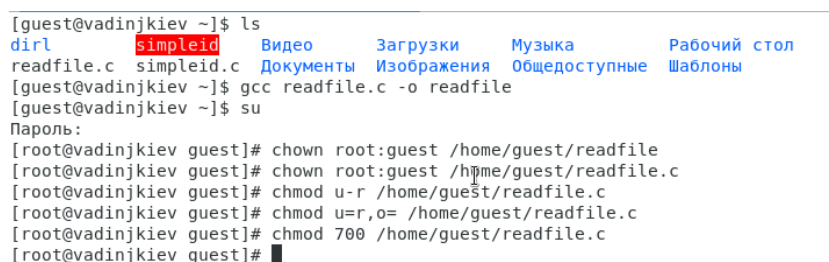
    int fd = open(argv[1], O_RDONLY);
    do
    {
        bytes_read = read(fd, buffer, sizeof(buffer));
        for (i=0; i<bytes_read; ++i) printf("%c", buffer[i]);
    }
    while (bytes_read == sizeof(buffer));
    close(fd);
    return 0;
}

"readfile.c" 24L, 392C
```

Figure 2.7: Программа readfile.c

1.14. Откомпилировал её (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)



```
[guest@vadinjkiy ~]$ ls
dirl simpleid Видео Загрузки Музыка Рабочий стол
readfile.c simpleid.c Документы Изображения Общедоступные Шаблоны
[guest@vadinjkiy ~]$ gcc readfile.c -o readfile
[guest@vadinjkiy ~]$ su
Пароль:
[root@vadinjkiy guest]# chown root:guest /home/guest/readfile
[root@vadinjkiy guest]# chown root:guest /home/guest/readfile.c
[root@vadinjkiy guest]# chmod u-r /home/guest/readfile.c
[root@vadinjkiy guest]# chmod u=r,o= /home/guest/readfile.c
[root@vadinjkiy guest]# chmod 700 /home/guest/readfile.c
[root@vadinjkiy guest]# █
```

Figure 2.8: Работа с программой readfile.c

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

2.9)

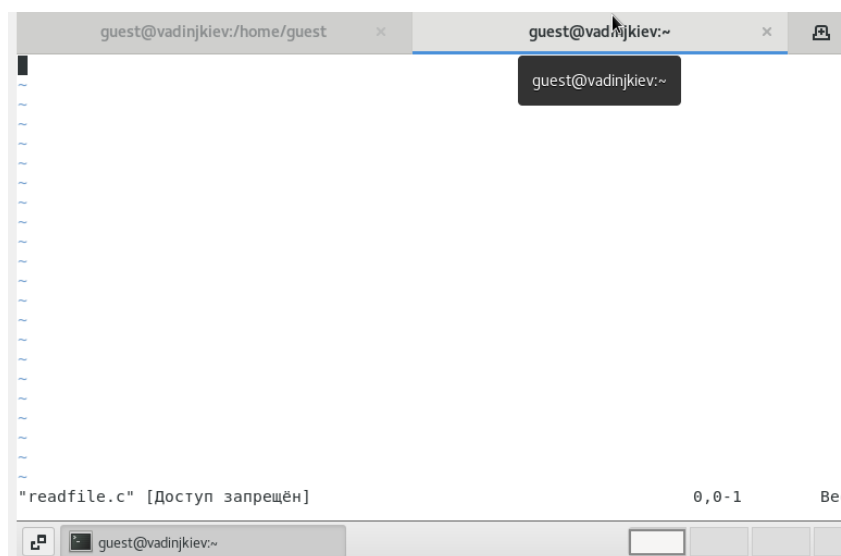


Figure 2.9: Проверка файла на чтение

1.17. Смнил у программы readfile владельца (рис. 2.8) и установил SetUID-бит (рис. ??).

1.18. Проверил, может ли программа readfile прочитать файл readfile.c. (рис. 2.10)

```
[root@vadinjkiiev guest]# chmod u+s /home/guest/readfile
[root@vadinjkiiev guest]# ./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;
}
```

Figure 2.10: Проверка чтения файла readfile.c и установка SetUid-бита

1.19. Проверил, может ли программа readfile прочитать файл /etc/shadow. (рис. 2.11)

```
[root@vadinjkiiev guest]# ./readfile /etc/shadow
root:$6$Y.k5lv1PyAs6c7qf$04xYFzhGiJKMh2FfkPuK/.j5iW3QrmWXFHbQIpzcf1UBqbv1D/Ah.0h2Cw308A
9TK/EXJs.0F4bJWj8VJF8vH1::0:99999:7:::
bin:!:18353:0:99999:7:::
daemon:!:18353:0:99999:7:::
adm:!:18353:0:99999:7:::
lp:!:18353:0:99999:7:::
sync:!:18353:0:99999:7:::
shutdown:!:18353:0:99999:7:::
halt:!:18353:0:99999:7:::
mail:!:18353:0:99999:7:::
operator:!:18353:0:99999:7:::
games:!:18353:0:99999:7:::
ftp:!:18353:0:99999:7:::
nobody:!:18353:0:99999:7:::
systemd-network:!!:18886:::
dbus:!!:18886:::
polkitd:!!:18886:::
libstoragemgmt:!!:18886:::
colord:!!:18886:::
rpc:!!:18886:0:99999:7:::
saned:!!:18886:::
saslauth:!!:18886:::
```

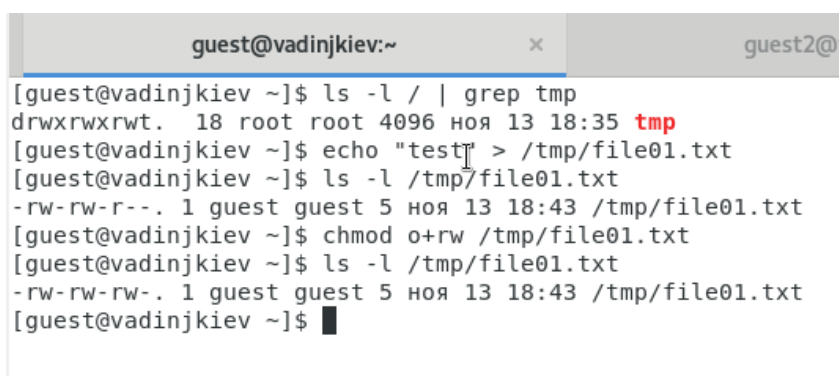
Figure 2.11: Проверка чтения файла /etc/shadow

2. Исследование Sticky-бита

2.1. Выяснил, установлен ли атрибут Sticky на директории /tmp, для чего выполнил команду (рис. 2.12)

2.2. От имени пользователя guest создал файл file01.txt в директории /tmp со словом test (рис. 2.12)

2.3. Просмотрел атрибуты у только что созданного файла и разрешил чтение и запись для категории пользователей «все остальные» (рис. 2.12)



```
guest@vadinjkiiev:~ x guest2@
[guest@vadinjkiiev ~]$ ls -l / | grep tmp
drwxrwxrwt. 18 root root 4096 ноя 13 18:35 tmp
[guest@vadinjkiiev ~]$ echo "test" > /tmp/file01.txt
[guest@vadinjkiiev ~]$ ls -l /tmp/file01.txt
-rw-rw-r--. 1 guest guest 5 ноя 13 18:43 /tmp/file01.txt
[guest@vadinjkiiev ~]$ chmod o+rw /tmp/file01.txt
[guest@vadinjkiiev ~]$ ls -l /tmp/file01.txt
-rw-rw-rw-. 1 guest guest 5 ноя 13 18:43 /tmp/file01.txt
[guest@vadinjkiiev ~]$
```

Figure 2.12: Исследование Sticky-бита от имени guest

2.4. От пользователя guest2 (не являющегося владельцем) попробовал прочитать файл /tmp/file01.txt. (рис. 2.13)

2.5. От пользователя guest2 попробовал дозаписать в файл /tmp/file01.txt слово test2 командой. (рис. 2.13) Операция прошла успешно.

2.6. Проверил содержимое файла командой (рис. 2.13)

2.7. От пользователя guest2 попробовал записать в файл /tmp/file01.txt слово test3, стерев при этом всю имеющуюся в файле информацию командой (рис. 2.13) Операция прошла успешно.

2.8. Проверил содержимое файла командой (рис. 2.13)

2.9. От пользователя guest2 попробовал удалить файл /tmp/file01.txt командой (рис. 2.13) Операция была не позволена.

```
[guest2@vadinjkiiev ~]$ cat /tmp/file01.txt
test2
[guest2@vadinjkiiev ~]$ echo "test2" >> /tmp/file01.txt
[guest2@vadinjkiiev ~]$ cat /tmp/file01.txt
test2
test2
[guest2@vadinjkiiev ~]$ echo "test3" > /tmp/file01.txt
[guest2@vadinjkiiev ~]$ cat /tmp/file01.txt
test3
[guest2@vadinjkiiev ~]$ rm /tmp/file01.txt
rm: невозможно удалить «/tmp/file01.txt»: Операция не позволена
[guest2@vadinjkiiev ~]$
```

Figure 2.13: Работа с file01.txt от имени guest2 при наличии Sticky-бита

2.10. Повысил свои права до суперпользователя следующей командой и выполнил после этого команду, снимающую атрибут t (Sticky-бит) с директории (рис. 2.14)

2.11. Покинул режим суперпользователя командой. (рис. 2.14)

```
[guest@vadinjkiiev ~]$ su -
Пароль:
Последний вход в систему:Сб ноя 13 18:35:30 MSK 2021на pts/0
[root@vadinjkiiev ~]# chmod -t /tmp
[root@vadinjkiiev ~]# exit
logout
[guest@vadinjkiiev ~]$ █
```

Figure 2.14: Снятие Sticky-бита с директории /tmp

2.12. От пользователя guest2 проверил, что атрибута t у директории /tmp. (рис. 2.15)

2.13. Повторил предыдущие шаги. (рис. 2.15) Теперь удалось удалить файл.

```
[guest2@vadinjkiiev ~]$ ls -l / | grep tmp
drwxrwxrwx. 18 root root 4096 ноя 13 18:49 tmp
[guest2@vadinjkiiev ~]$ cat /tmp/file01.txt
test3
[guest2@vadinjkiiev ~]$ echo "test2" >> /tmp/file01.txt
[guest2@vadinjkiiev ~]$ cat /tmp/file01.txt
test3
test2
[guest2@vadinjkiiev ~]$ echo "test4" > /tmp/file01.txt
[guest2@vadinjkiiev ~]$ cat /tmp/file01.txt
test4
[guest2@vadinjkiiev ~]$ rm /tmp/file01.txt
[guest2@vadinjkiiev ~]$ ls /tmp
ssh-Ah2rv507Ey4Y
systemd-private-7d2a8a40af4d4c15a1cdebb118bfda02-bolt.service-ncg1hp
systemd-private-7d2a8a40af4d4c15a1cdebb118bfda02-chronyd.service-M74Sgl
systemd-private-7d2a8a40af4d4c15a1cdebb118bfda02-colord.service-pQwXB2
systemd-private-7d2a8a40af4d4c15a1cdebb118bfda02-cups.service-yzwlE
systemd-private-7d2a8a40af4d4c15a1cdebb118bfda02-fwupd.service-njKRXJ
systemd-private-7d2a8a40af4d4c15a1cdebb118bfda02-rtkit-daemon.service-8AFXck
tracker-extract-files.1000
tracker-extract-files.1001
yum_save_tx.2021-11-09.17-31.BFeJZp.yumtx
yum_save_tx.2021-11-13.18-08.WrPNth.yumtx
```

Figure 2.15: Работа с file01.txt от имени guest2 без Sticky-бита

2.14. Удалось удалить файл от имени пользователя, не являющегося его владельцем.

2.15. Повысил свои права до суперпользователя и вернул атрибут t на директорию /tmp (рис. 2.16)

```
[guest@vadinjkiiev ~]$ su -
Пароль:
Последний вход в систему:Сб ноя 13 18:48:42 MSK 2021на pts/1
[root@vadinjkiiev ~]# chmod +t /tmp
[root@vadinjkiiev ~]# exit
logout
[guest@vadinjkiiev ~]$
```

Figure 2.16: 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