

# Лабораторная работа №5

Дискреционное разграничение прав в Linux.


Исследование влияния дополнительных атрибутов

---

Калинина Кристина Сергеевна

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.

1. Выполнение лабораторной работы
2. Оформление отчета и презентации
3. Выгрузка видео на youtube и файлов на GitHub

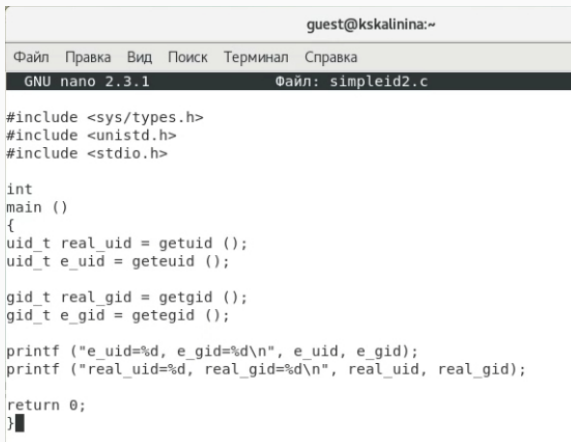


```
guest@kskalinina:~  
Файл Правка Вид Поиск Терминал Справка  
GNU nano 2.3.1 Файл: simpleid.c  
  
#include <sys/types.h>  
#include <unistd.h>  
#include <stdio.h>  
  
int  
main ()  
{  
    uid_t uid = geteuid ();  
    gid_t gid = getegid ();  
    printf ("uid=%d, gid=%d\n", uid, gid);  
    return 0;  
}
```

Figure 1: Код программы simpleid.c

```
[guest@kskalinina ~]$ gcc simpleid.c -o simpleid
[guest@kskalinina ~]$ ./simpleid
uid=1001, gid=1001
[guest@kskalinina ~]$ id
uid=1001(guest) gid=1001(guest) rpyнны=1001(guest) контекст=unconfined_u:unconfi
ned_r:unconfined_t:s0-s0:c0.c1023
```

Figure 2: Выполнение программы и сравнение результата с выводом команды `id`



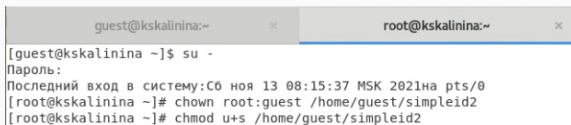
```
guest@kskalina:~  
Файл  Правка  Вид  Поиск  Терминал  Справка  
GNU nano 2.3.1      Файл: simpleid2.c  
  
#include <sys/types.h>  
#include <unistd.h>  
#include <stdio.h>  
  
int  
main ()  
{  
    uid_t real_uid = getuid ();  
    uid_t e_uid = geteuid ();  
  
    gid_t real_gid = getgid ();  
    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;  
}
```

Figure 3: Код программы simpleid2.c

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

Figure 4: Выполнение программы simpleid2.c

# Смена атрибутов и владельца файла simpleid2



```
guest@kskalinina:~$ su -
Пароль:
Последний вход в систему:Сб ноя 13 08:15:37 MSK 2021на pts/0
[root@kskalinina ~]# chown root:guest /home/guest/simpleid2
[root@kskalinina ~]# chmod u+s /home/guest/simpleid2
```

Figure 5: Смена владельца и установка атрибутов

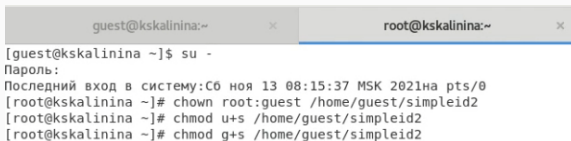


```
[guest@kskalinina ~]$ ls -l simpleid2
-rwsrwxr-x. 1 root guest 8576 ноя 13 08:33 simpleid2
[guest@kskalinina ~]$ ./simpleid2
e_uid=0, e_gid=1001
real_uid=1001, real_gid=1001
[guest@kskalinina ~]$ id
uid=1001(guest) gid=1001(guest) rpyнны=1001(guest) контекст=unconfined_u:unconfi
ned_r:unconfined_t:s0-s0:c0.c1023
```

Figure 6: Проверка и запуск программы

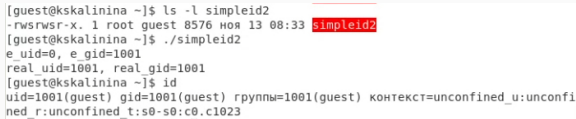


## Повтор действий относительно SetGID-бита



```
guest@kskalinina:~  
[guest@kskalinina ~]$ su -  
Пароль:  
Последний вход в систему:Сб ноя 13 08:15:37 MSK 2021на pts/0  
[root@kskalinina ~]# chown root:guest /home/guest/simpleid2  
[root@kskalinina ~]# chmod u+s /home/guest/simpleid2  
[root@kskalinina ~]# chmod g+s /home/guest/simpleid2
```

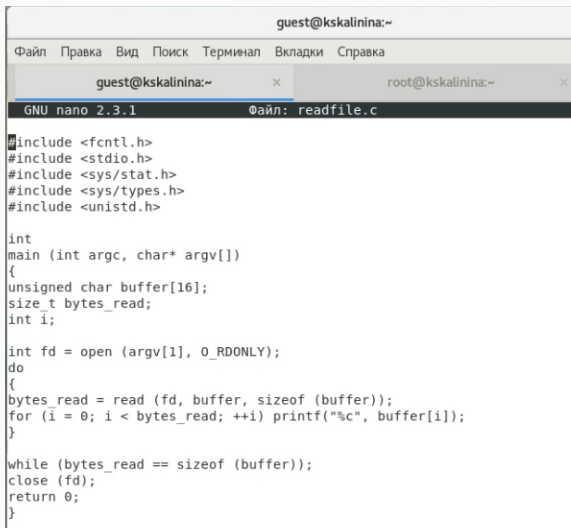
Figure 7: Установка атрибутов



```
[guest@kskalinina ~]$ ls -l simpleid2  
-rwsrwsr-x. 1 root guest 8576 ноя 13 08:33 simpleid2  
[guest@kskalinina ~]$ ./simpleid2  
e_uid=0, e_gid=1001  
real_uid=1001, real_gid=1001  
[guest@kskalinina ~]$ id  
uid=1001(guest) gid=1001(guest) rгруппы=1001(guest) контекст=unconfined_u:unconfi  
ned_r:unconfined_t:s0-s0:c0.c1023
```

Figure 8: Повтор действий с SetGID-битом

# Создание readfile.c



The screenshot shows a terminal window with the title bar 'guest@kskalinina:~'. The menu bar includes 'Файл', 'Правка', 'Вид', 'Поиск', 'Терминал', 'Вкладки', and 'Справка'. There are two tabs: 'guest@kskalinina:~' (active) and 'root@kskalinina:~'. The status bar at the bottom indicates 'GNU nano 2.3.1' and 'Файл: readfile.c'. The code displayed is as follows:

```
#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 9: Код программы readfile.c

```
[guest@kskalinina ~]$ nano readfile.c  
[guest@kskalinina ~]$ gcc readfile.c -o readfile
```

Figure 10: Компиляция программы readfile.c

# Смена прав и владельца файла readfile.c

```
[root@kskalinina ~]# chown root:guest /home/guest/readfile.c
[root@kskalinina ~]# chmod 700 /home/guest/readfile.c
[root@kskalinina ~]# ls -l /home/guest/readfile.c
-rwx-----. 1 root guest 407 ноя 13 08:45 /home/guest/readfile.c
```

Figure 11: Смена владельца файла readfile.c и прав на него

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

Figure 12: Отказ в чтении readfile.c

## Смена прав и владельца файла readfile

```
[root@kskalinina ~]# chown root:guest /home/guest/readfile
[root@kskalinina ~]# chmod u+s /home/guest/readfile
[root@kskalinina ~]# ls -l /home/guest/readfile
-rwsrwxr-x. 1 root guest 8512 ноя 13 08:45 /home/guest/readfile
```

Figure 13: Смена владельца файла readfile и установка SetU'D-бит

# Чтение readfile.c и “/etc/shadow”

```
guest@kskalinina:~ x root@kskalinina:~ x [icon]
[guest@kskalinina ~]$ ./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;
}

[guest@kskalinina ~]$ ./readfile /etc/shadow
root:$6$IRM6h9jvGIVhR9mE$6q0mS0ntqFo0SDForrcuFmaLAe/bYovJ68Nwd7SUm7Am2HzFgs
a9EIg6WFiWmX8GtxDMESxuz4vkWmyJ.: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:::
```

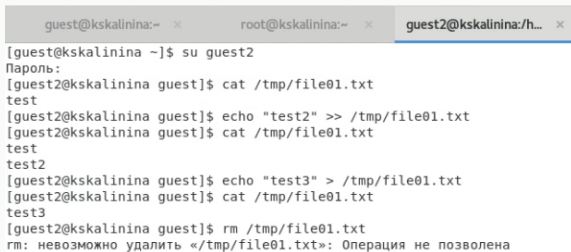
```
[guest@kskalinina ~]$ ls -l / | grep tmp  
drwxrwxrwt. 16 root root 4096 ноя 13 08:49 tmp
```

Figure 15: атрибут Sticky на директории /tmp

```
[guest@kskalinina ~]$ echo "test" > /tmp/file01.txt  
[guest@kskalinina ~]$ ls -l /tmp/file01.txt  
-rw-rw-r--. 1 guest guest 5 ноя 13 08:50 /tmp/file01.txt  
[guest@kskalinina ~]$ chmod o+rw /tmp/file01.txt  
[guest@kskalinina ~]$ ls -l /tmp/file01.txt  
-rw-rw-rw-. 1 guest guest 5 ноя 13 08:50 /tmp/file01.txt
```

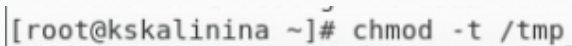
Figure 16: Создание file01.txt и смена атрибутов

## Работа с файлом от пользователя guest2



```
guest@kskalinina:~ × root@kskalinina:~ × guest2@kskalinina:/h... ×
[guest@kskalinina ~]$ su guest2
Пароль:
[guest2@kskalinina guest]$ cat /tmp/file01.txt
test
[guest2@kskalinina guest]$ echo "test2" >> /tmp/file01.txt
[guest2@kskalinina guest]$ cat /tmp/file01.txt
test
test2
[guest2@kskalinina guest]$ echo "test3" > /tmp/file01.txt
[guest2@kskalinina guest]$ cat /tmp/file01.txt
test3
[guest2@kskalinina guest]$ rm /tmp/file01.txt
rm: невозможно удалить «/tmp/file01.txt»: Операция не позволена
```

Figure 17: Работа с file01.txt от guest2



```
[root@kskalinina ~]# chmod -t /tmp_
```

Figure 18: Снятие атрибута Sticky с директории /tmp



## Работа с файлом от пользователя guest2

```
[guest2@kskalinina guest]$ ls -l / | grep tmp
drwxrwxrwx. 16 root root 4096 ноя 13 08:54 tmp
[guest2@kskalinina guest]$ cat /tmp/file01.txt
test3
[guest2@kskalinina guest]$ echo "test2" >> /tmp/file01.txt
[guest2@kskalinina guest]$ cat /tmp/file01.txt
test3
test2
[guest2@kskalinina guest]$ echo "test" > /tmp/file01.txt
[guest2@kskalinina guest]$ cat /tmp/file01.txt
test
[guest2@kskalinina guest]$ rm /tmp/file01.txt
[guest2@kskalinina guest]$ ls /tmp
ssh-p4ccW0zsbZtc
systemd-private-3fdb085f1ef4488ea4acc18d5850a696-bolt.service-mmWSfg
systemd-private-3fdb085f1ef4488ea4acc18d5850a696-chronyd.service-eBERBM
systemd-private-3fdb085f1ef4488ea4acc18d5850a696-colord.service-coEaBA
systemd-private-3fdb085f1ef4488ea4acc18d5850a696-cups.service-g9hvGY
systemd-private-3fdb085f1ef4488ea4acc18d5850a696-fwupd.service-ENFI1L
systemd-private-3fdb085f1ef4488ea4acc18d5850a696-rtkit-daemon.service-anmI76
tracker-extract-files.1001
yum_save_tx.2021-11-13.08-15.DWWMJR.yumtx
```

Figure 19: Повторное выполнение команд от guest2

```
[root@kskalinina ~]# chmod +t /tmp
[root@kskalinina ~]# ls -l / | grep tmp
drwxrwxrwt. 16 root root 4096 ноя 13 08:56 tmp
```

Figure 20: Возвращение атрибута Sticky на директорию /tmp

Таким образом я успешно приобрела изучила механизмы изменения идентификаторов, применения 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.