

# Лабораторная работа №5

Nikita A. Toponen

RUDN University, 4 October 2022 Moscow, Russia

# Дискреционное разграничение прав в Linux. Расширенные атрибуты

## Цель выполнения 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

# Выполнение работы

# Выполнение работы

```
[guest@natoponen ~]$ gcc -v
Используются внутренние спецификации.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/usr/libexec/gcc/x86_64-redhat-linux/4.8.5/lto-wrapper
Целевая архитектура: x86_64-redhat-linux
Параметры конфигурации: ../configure --prefix=/usr --mandir=/usr/share/man --inf
odir=/usr/share/info --with-bugurl=http://bugzilla.redhat.com/bugzilla --enable-
bootstrap --enable-shared --enable-threads=posix --enable-checking=release --wit
h-system-zlib --enable-__cxa_atexit --disable-libunwind-exceptions --enable-gnu-
unique-object --enable-linker-build-id --with-linker-hash-style=gnu --enable-lan-
guages=c,c++,objc,obj-c++,java,fortran,ada,go,lto --enable-plugin --enable-initf
ini-array --disable-libgcj --with-isl=/builddir/build/BUILD/gcc-4.8.5-20150702/o
bj-x86_64-redhat-linux/isl-install --with-cloog=/builddir/build/BUILD/gcc-4.8.5-
20150702/obj-x86_64-redhat-linux/cloog-install --enable-gnu-indirect-function --
with-tune=generic --with-arch_32=x86-64 --build=x86_64-redhat-linux
Модель многопоточности: posix
gcc версия 4.8.5 20150623 (Red Hat 4.8.5-44) (GCC)
```

Рис.1 Компилятор gcc

## Выполнение работы

```
[guest@natoponen ~]$ su  
Пароль:  
[root@natoponen guest]# setenforce 0  
[root@natoponen guest]# getenforce  
Permissive  
—
```

Рис.2 Отключение SELinux

## Выполнение работы

```
[guest@natoponen ~]$ touch simpleid.c
[guest@natoponen ~]$ ls
dirl          Видео          Загрузки      Музыка        Рабочий стол
simpleid.c     Документы     Изображения  Общедоступные  Шаблоны
[guest@natoponen ~]$ vim simpleid.c
```

Рис.3 Создание программы simpleid.c

# Выполнение работы

```
#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;
}
```

Код программы simpleid.c



## Выполнение работы

```
[guest@natoponen ~]$ gcc simpleid.c -o simpleid  
[guest@natoponen ~]$ ./simpleid  
uid=1001, gid=1001
```

Рис.4 Компиляция и выполнение simpleid.c

## Выполнение работы

```
[guest@natoponen ~]$ id  
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfi  
ned_r:unconfined_t:s0-s0:c0.c1023
```

Рис.5 Результат команды id

# Выполнение работы

```
#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;
}
```

Код программы simpleid2.c

## Выполнение работы

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

Рис.6 Компиляция и выполнение simpleid2

## Выполнение работы

```
[guest@natoponen ~]$ su
```

Пароль:

```
[root@natoponen guest]# chown root:guest /home/guest/simpleid2
```

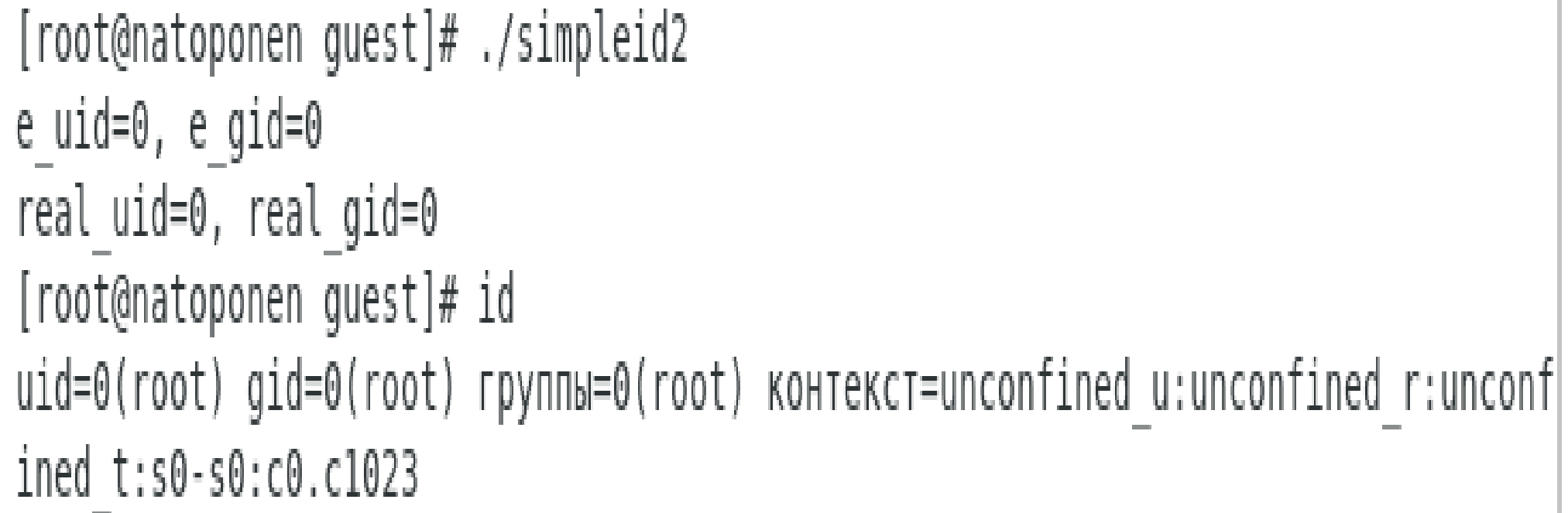
```
[root@natoponen guest]# chmod u+s /home/guest/simpleid2
```

```
[root@natoponen guest]# ls -l simpleid2
```

```
-rwsrwxr-x. 1 root guest 8616 окт  3 22:43 simpleid2
```

Рис.7 Изменение владельца и прав на файл simpleid2

## Выполнение работы



```
[root@natoponen guest]# ./simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@natoponen guest]# id
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

The image shows a terminal window with a red title bar. It displays the execution of two commands: `./simpleid2` and `id`. The output of `./simpleid2` shows effective and real user/group IDs as 0. The output of `id` shows the user is root (UID 0), the group is root (GID 0), and the SELinux context is `unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023`.

Рис.8 Выполнение `simpleid2` и `id`

# Выполнение работы

```
#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;
}
```

## Выполнение работы

```
[guest@natoponen ~]$ gcc readfile.c -o readfile
[guest@natoponen ~]$ chown root:guest readfile.c
chown: изменение владельца «readfile.c»: Операция не позволена
[guest@natoponen ~]$ su
Пароль:
[root@natoponen guest]# chown root:guest readfile.c
[root@natoponen guest]# chmod 700 readfile.c
[root@natoponen guest]# su guest
[guest@natoponen ~]$ vim readfile.c
```

Рис.9 Изменение владельца и прав на файл readfile.c



# Выполнение работы

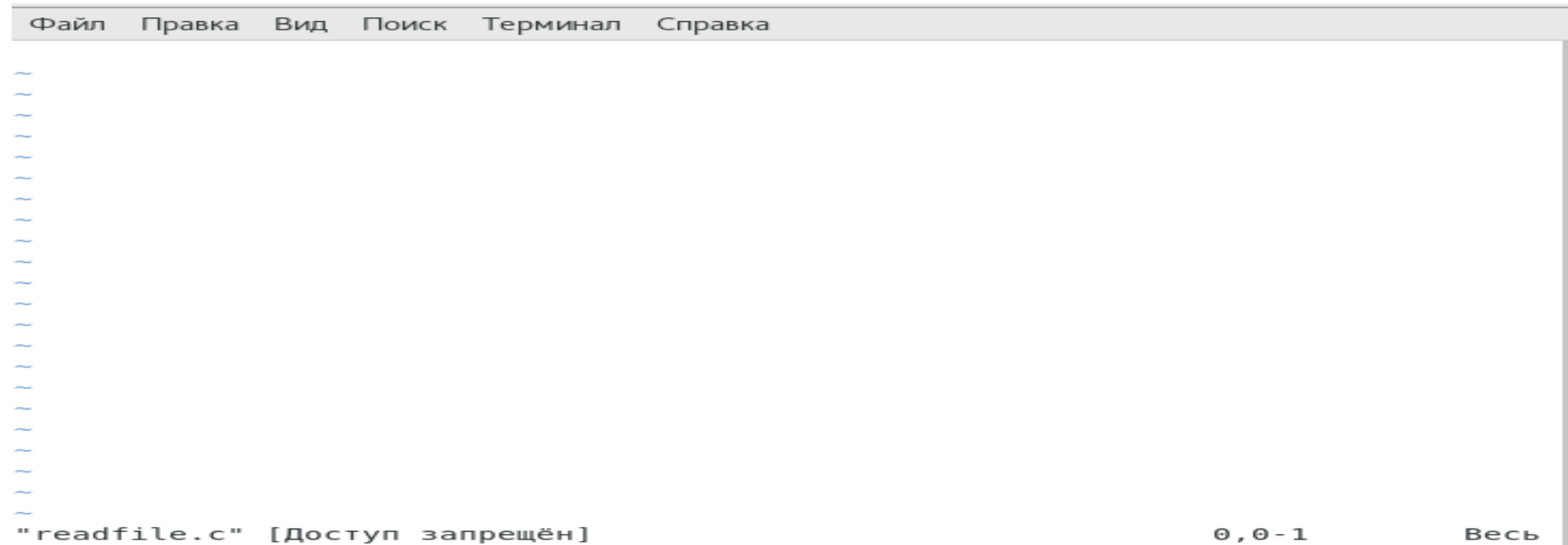


Рис.10 Отказ в чтении пользователю guest

## Выполнение работы

```
[guest@natoponen ~]$ su
```

Пароль:

```
[root@natoponen guest]# chown root:guest readfile
```

```
[root@natoponen guest]# chmod u+s readfile
```

Рис.11 Установка UID бита для readfile.c

# Выполнение работы

```
[root@natoponen 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;
}
```

Рис.12 Выполнение программы для файла readfile.c

# Выполнение работы

```
[root@natoponen guest]# ./readfile /etc/shadow
root:$6$TGptGmRXmrZUPQ.V$TGILQ7bnDoQ6k74Eb0VDTzYTFlyxG8ajLtAB0bjiu5GmN38Ez1Z0aK4
ChUisX9F7H4reBFVUb3mRXn0zF0w0o0::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:!!:19241:::::
dbus:!!:19241:::::
polkitd:!!:19241:::::
libstoragegmt:!!:19241:::::
colord:!!:19241:::::
rpc:!!:19241:0:99999:7:::
saned:!!:19241:::::
```

Рис.13 Выполнение программы для файла /etc/shadow

## Выполнение работы

```
[guest@natoponen ~]$ ls -l / | grep tmp
drwxrwxrwt. 15 root root 4096 окт  3 23:01 tmp
[guest@natoponen ~]$ echo "test" > /tmp/file01.txt
[guest@natoponen ~]$ ls -l /tmp/file01.txt
-rw-rw-r--. 1 guest guest 5 окт  3 23:04 /tmp/file01.txt
[guest@natoponen ~]$ chmod o+rw /tmp/file01.txt
[guest@natoponen ~]$ ls -l /tmp/file01.txt
-rw-rw-rw-. 1 guest guest 5 окт  3 23:04 /tmp/file01.txt
[guest@natoponen ~]$ su guest2
Пароль:
[guest2@natoponen guest]$ cat /tmp/file01.txt
test
[guest2@natoponen guest]$ echo "test2" > /tmp/file01.txt
[guest2@natoponen guest]$ cat /tmp/file01.txt
test2
[guest2@natoponen guest]$ echo "test3" > /tmp/file01.txt
[guest2@natoponen guest]$ cat /tmp/file01.txt
test3
[guest2@natoponen guest]$ rm /tmp/file01.txt
rm: невозможно удалить «/tmp/file01.txt»: Операция не позволена
```

Рис.14 Работа со Sticky битом

## Выполнение работы

```
[root@natoponen guest]# chmod -t /tmp/
[root@natoponen guest]# su guest2
[guest2@natoponen guest]$ ls -l / | grep tmp
drwxrwxrwx. 15 root root 4096 okt  3 23:07 tmp
[guest2@natoponen guest]$ echo "test4" > /tmp/file01.txt
[guest2@natoponen guest]$ cat /tmp/file01.txt
test4
[guest2@natoponen guest]$ rm /tmp/file01.txt
[guest2@natoponen guest]$ cd /tmp/
[guest2@natoponen tmp]$ ls
ssh-VRF8jE9qR0YD
systemd-private-flab0ea4e2b94e828873299a217c90c2-bolt.service-lqpHGa
systemd-private-flab0ea4e2b94e828873299a217c90c2-colord.service-xcczcG
systemd-private-flab0ea4e2b94e828873299a217c90c2-cups.service-wNJwa2
systemd-private-flab0ea4e2b94e828873299a217c90c2-fwupd.service-UWMFVk
systemd-private-flab0ea4e2b94e828873299a217c90c2-rtkit-daemon.service-Tvm02P
tracker-extract-files.1001
yum_save_tx.2022-09-25.17-03.WeLvQ6.yumtx
yum_save_tx.2022-10-03.22-34.R5HCjr.yumtx
```

Рис.15 Работа с файлом без Sticky бита

## Выполнение работы

```
[guest2@natoponen tmp]$ su
```

Пароль:

```
[root@natoponen tmp]# chmod +t /tmp/
```

```
[root@natoponen tmp]# exit
```

```
exit
```

—

Рис.16 Установление атрибута t

# Выводы

В ходе выполнения данной лабораторной работы я:

- Повысил свои навыки использования интерфейса командой строки (CLI)
- Изучил механизмы изменения идентификаторов, применения SetUID-, SetGID- и Sticky-битов
- Рассмотрел работу механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов



**Спасибо за внимание!**