

Лабораторная работа 5.
Дискреционное разграничение прав в Linux.
Исследование влияния дополнительных
атрибутов.

Поленикова Анна Алексеевна

Москва, 2021

Российский Университет Дружбы Народов

Цель лабораторной работы

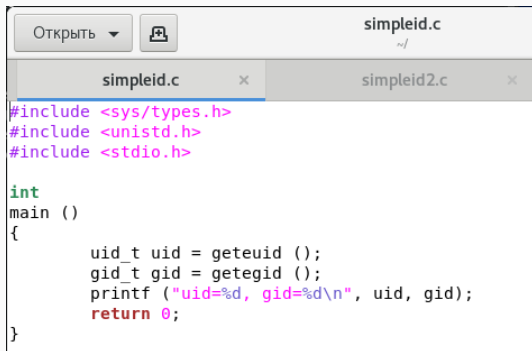
Изучение механизмов изменения идентификаторов, применения SetUID- и Sticky-битов. Получение практических навыков работы в консоли с дополнительными атрибутами. Рассмотрение работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

Подготовка лабораторного стенда

```
[guest@aapolenikova ~]$ gcc -v
Используются внутренние спецификации.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/usr/libexec/gcc/x86_64-redhat-linux/8/lto-wrapper
OFFLOAD_TARGET_NAMES=nvptx-none
OFFLOAD_TARGET_DEFAULT=1
Целевая архитектура: x86_64-redhat-linux
Параметры конфигурации: ../configure --enable-bootstrap --enable-languages=c,c++,fortran,lto --prefix=/usr --mandir=/usr/share/man --infodir=/usr/share/info --with-bugurl=http://bugzilla.redhat.com/bugzilla --enable-shared --enable-threads=posix --enable-checking=release --enable-multilib --with-system-zlib --enable-_cxa_atexit --disable-libunwind-exceptions --enable-gnu-unique-object --enable-linker-build-id --with-gcc-major-version-only --with-linker-hash-style=gnu --enable-plugin --enable-initfini-array --with-isl --disable-libmpx --enable-offload-targets=nvptx-none --without-cuda-driver --enable-gnu-indirect-function --enable-cet --with-tune=generic --with-arch_32=x86-64 --build=x86_64-redhat-linux
Модель многопоточности: posix
gcc версия 8.5.0 20210514 (Red Hat 8.5.0-3) (GCC)
[guest@aapolenikova ~]$ getenforce
Permissive
```

Рис. 1: Подготовка

Создание программы 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;
}
```

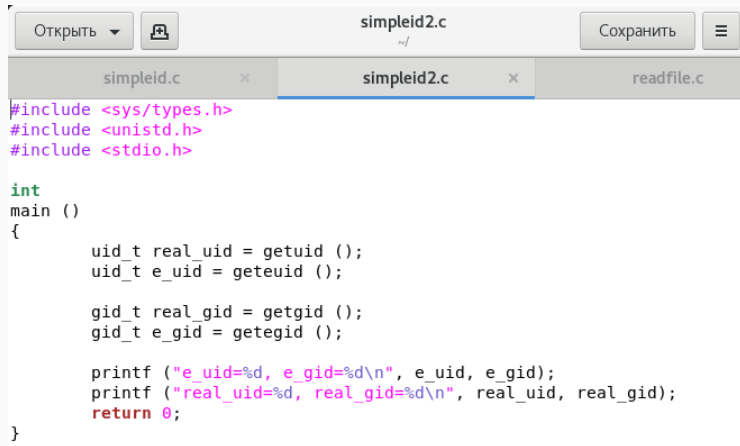
Рис. 2: Код simpleid.c

Компиляция и выполнение программы simpleid

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

Рис. 3: Компиляция и выполнение программы simpleid

Создание программы 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;
}
```

Рис. 4: Код simpleid2.c

Компиляция и выполнение программы simpleid2

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

Рис. 5: Компиляция и выполнение программы simpleid2

Изменение владельца и атрибутов simpleid2

```
[root@aapolenikova guest]# chown root:guest /home/guest/simpleid2  
[root@aapolenikova guest]# chmod u+s /home/guest/simpleid2
```

Рис. 6: Изменение владельца и атрибутов simpleid2

```
[guest@aapolenikova ~]$ ls -l simpleid2  
-rwsrwxr-x. 1 root guest 17696 ноя 13 17:35 simpleid2  
[guest@aapolenikova ~]$ ./simpleid2  
e_uid=0, e_gid=1001  
real uid=1001, real_gid=1001  
[guest@aapolenikova ~]$ id  
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

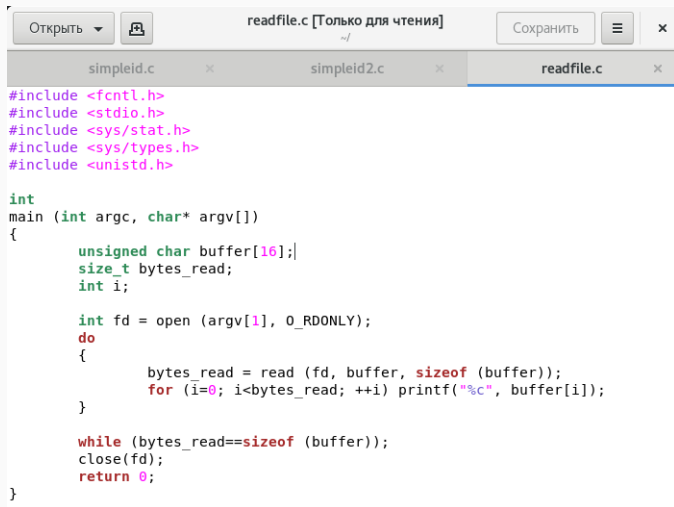
Рис. 7: Запуск simpleid2 и id

Повторение операций для SetGID-бита

```
[root@aapolenikova guest]# chmod g+s /home/guest/simpleid2
[root@aapolenikova guest]# exit
exit
[guest@aapolenikova ~]$ ls -l simpleid2
-rwsrwsr-x. 1 root guest 17696 ноя 13 17:35 simpleid2
[guest@aapolenikova ~]$ ./simpleid2
e_uid=0, e_gid=1001
real_uid=1001, real_gid=1001
[guest@aapolenikova ~]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Рис. 8: Повторение операций для SetGID-бита

Создание программы 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. 9: Код readfile.c

Компиляция. Смена владельца и изменение прав файла readfile.c

```
[guest@aapolenikova ~]$ gcc readfile.c -o readfile
[guest@aapolenikova ~]$ su
Пароль:
[root@aapolenikova guest]# chown root:guest /home/guest/readfile.c
[root@aapolenikova guest]# chmod ug-r /home/guest/readfile.c
[root@aapolenikova guest]# exit
exit
[guest@aapolenikova ~]$ ls -l readfile.c
--w---r--. 1 root guest 414 ноя 13 18:17 readfile.c
[guest@aapolenikova ~]$ cat readfile.c
cat: readfile.c: Отказано в доступе
```

Рис. 10: Компиляция. Смена владельца и изменение прав
файла readfile.c

Операции с программой readfile

[illegible]

Рис. 11: Смена владельца программы readfile и установка SetUID-бита. Чтение файла readfile.c

[illegible]

Рис. 12: Чтение файла /etc/shadow

Проверка Sticky атрибута. Создание файла file1

```
[guest@aapolenikova ~]$ ls -l / | grep tmp
drwxrwxrwt. 13 root root 4096 ноя 13 18:41 tmp
[guest@aapolenikova ~]$ echo "test" > /tmp/file01.txt
[guest@aapolenikova ~]$ ls -l /tmp/file01.txt
-rw-rw-r--. 1 guest guest 5 ноя 13 18:57 /tmp/file01.txt
[guest@aapolenikova ~]$ chmod o+rw /tmp/file01.txt
[guest@aapolenikova ~]$ ls -l /tmp/file01.txt
-rw-rw-rw-. 1 guest guest 5 ноя 13 18:57 /tmp/file01.txt
```

Рис. 13: Проверка Sticky атрибута. Создание файла file1.
Изменение атрибутов для категории пользователей «все остальные»

Операции с файлом file01.txt

```
[guest2@aapolenikova guest]$ cat /tmp/file01.txt
test
[guest2@aapolenikova guest]$ echo "test" > /tmp/file01.txt
[guest2@aapolenikova guest]$ cat /tmp/file01.txt
test
[guest2@aapolenikova guest]$ echo "test2" > /tmp/file01.txt
[guest2@aapolenikova guest]$ cat /tmp/file01.txt
test2
[guest2@aapolenikova guest]$ echo "test3" > /tmp/file01.txt
[guest2@aapolenikova guest]$ cat /tmp/file01.txt
test3
[guest2@aapolenikova guest]$ rm /tmp/file01.txt
rm: невозможно удалить '/tmp/file01.txt': Операция не позволена
```

Рис. 14: Операции с файлом file01.txt

Снятие t атрибута с директории /tmp

```
[guest2@aapolenikova guest]$ su -  
Пароль:  
[root@aapolenikova ~]# chmod -t /tmp  
[root@aapolenikova ~]# exit
```

Рис. 15: Снятие t атрибута с директории /tmp

Операции с файлом file01.txt после снятия t атрибута

```
[guest2@aaapolenikova guest]$ ls -l / | grep tmp
drwxrwxrwx.  13 root  root 4096 ноя 13 19:03 tmp
[guest2@aaapolenikova guest]$ cat /tmp/file01.txt
test3
[guest2@aaapolenikova guest]$ echo "test" > /tmp/file01.txt
[guest2@aaapolenikova guest]$ cat /tmp/file01.txt
test
[guest2@aaapolenikova guest]$ echo "test2" > /tmp/file01.txt
[guest2@aaapolenikova guest]$ cat /tmp/file01.txt
test2
[guest2@aaapolenikova guest]$ rm /tmp/file01.txt
```

Рис. 16: Операции с файлом file01.txt после снятия t атрибута

Возвращение t атрибута директории /tmp

```
[guest2@aapolenikova guest]$ su -  
Пароль:  
[root@aapolenikova ~]# chmod +t /tmp  
[root@aapolenikova ~]# exit  
выход
```

Рис. 17: Возвращение t атрибута директории /tmp

В результате проделанной лабораторной работы были изучены механизмы изменения идентификаторов, применения SetUID- и Sticky-битов, получены практические навыки работы в консоли с дополнительными атрибутами, а также рассмотрены работы механизма смены идентификатора процессов пользователей и влияние бита Sticky на запись и удаление файлов.