

Лабораторная работа 5

Дискреционноеразграничение прав в Linux. Исследование влияния дополнительных атрибутов. - Казакова Виктория Алексеевна

7 октября 2023

Российский университет дружбы народов, Москва, Россия

- Казакова Виктория Алексеевна
- студент кафедры математического
модулирования и искусственного интеллекта
- Российский университет дружбы народов
- 1032201659@rudn.ru

Вводная часть

- Получение практических навыков работы в консоли с атрибутами файлов
- Закрепление теоретических основ дискреционного разграничения доступа в современных системах с открытым кодом на базе ОС Linux

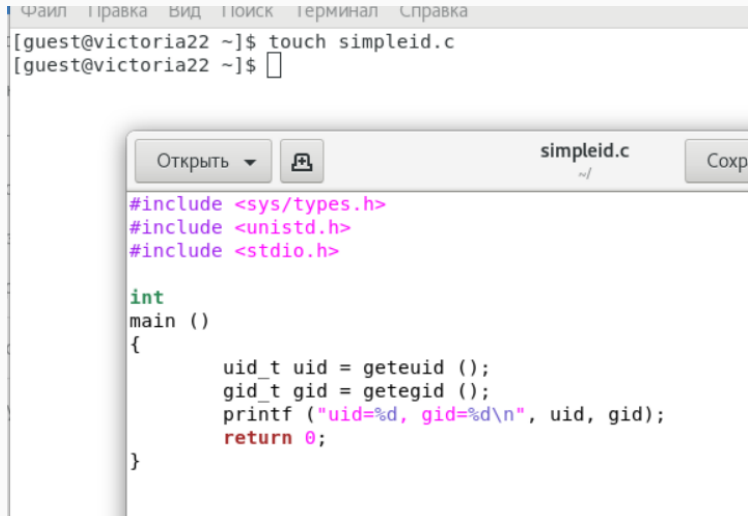
- Процессор pandoc для входного формата Markdown
- Операционная система Centos 7.0
- Сервис для хостинга IT-проектов GitHub
- Методические материалы

Результаты выполнения работы

Подготовка лабораторного стенда

```
[root@victoria22 victoria22]# 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 --infodir=
/usr/share/info --with-bugurl=http://bugzilla.redhat.com/bugzilla --enable-bootstrap
--enable-shared --enable-threads=posix --enable-checking=release --with-system-zlib -
-enable-__cxa_atexit --disable-libunwind-exceptions --enable-gnu-unique-object --enab
le-linker-build-id --with-linker-hash-style=gnu --enable-languages=c,c++,objc,obj-c++
,java,fortran,ada,go,lto --enable-plugin --enable-initfini-array --disable-libgcj --w
ith-isl=/builddir/build/BUILD/gcc-4.8.5-20150702/obj-x86_64-redhat-linux/isl-install
--with-cloog=/builddir/build/BUILD/gcc-4.8.5-20150702/obj-x86_64-redhat-linux/cloog-i
ninstall --enable-gnu-indirect-function --with-tune=generic --with-arch_32=x86_64 --bui
ld=x86_64-redhat-linux
Модель многопоточности: posix
gcc версия 4.8.5 20150623 (Red Hat 4.8.5-44) (GCC)
[root@victoria22 victoria22]# setenforce 0
[root@victoria22 victoria22]# getenforce
Permissive
```

Создание, компиляция и запуск файла simpleid.c

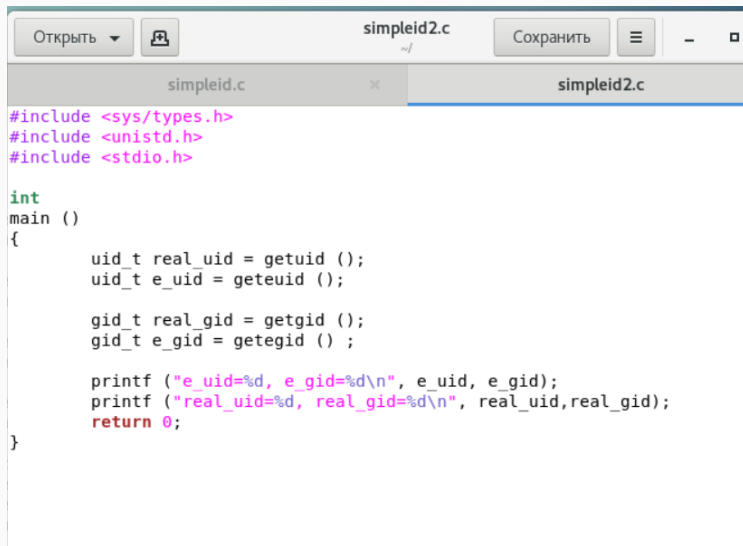


The image shows a terminal window at the top and a code editor window below it. The terminal window has a menu bar with 'Файл', 'Правка', 'Вид', 'Поиск', 'Терминал', and 'Справка'. The prompt is '[guest@victoria22 ~]\$' and the command 'touch simpleid.c' has been executed. The code editor window has a title bar with 'simpleid.c' and buttons for 'Открыть', a file icon, and 'Сохранить'. The code in the editor is as follows:

```
#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;
}
```


Аналогичная процедура для файла 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;
}
```

```
[quest@victoria22 ~]$ gcc simpleid2.c -o simpleid2
```

От имени суперпользователя выполнила команды и проверила их правильность

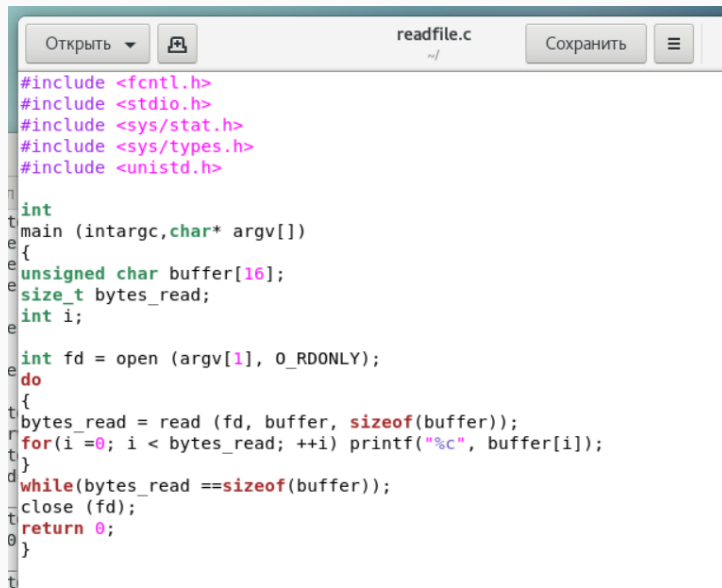
```
[guest@victoria22 ~]$ su
Пароль:
[root@victoria22 guest]# chown root:guest /home/guest/simpleid2
[root@victoria22 guest]# chmod u+s /home/guest/simpleid2
[root@victoria22 guest]# sudo
usage: sudo -h | -K | -k | -V
usage: sudo -v [-AknS] [-g group] [-h host] [-p prompt] [-u user]
usage: sudo -l [-AknS] [-g group] [-h host] [-p prompt] [-U user] [-u user]
[command]
usage: sudo [-AbEHknPS] [-r role] [-t type] [-C num] [-g group] [-h host] [-p
prompt] [-T timeout] [-u user] [VAR=value] [-i|-s] [<command>]
usage: sudo -e [-AknS] [-r role] [-t type] [-C num] [-g group] [-h host] [-p
prompt] [-T timeout] [-u user] file ...
[root@victoria22 guest]#
```

```
[root@victoria22 guest]# ls -l simpleid2
-rwsrwxr-x. 1 root guest 8616 окт  7 17:01 simpleid2
[root@victoria22 guest]#
```

Проделала тоже самое относительно SetGID-бита

```
[root@victoria22 guest]# chmod g+s /home/guest/simpleid2  
[root@victoria22 guest]# ls -l simpleid2  
-rwsrwsr-x. 1 root guest 8616 окт  7 17:01 simpleid2  
[root@victoria22 guest]#
```

Создала программу readfile.c и откомпилировала его



```
#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;
}
```

Сменила владельца у файла readfile.c и изменила права так, чтобы только суперпользователь (root) мог прочитать его, а guest не мог. Убедилась, что пользователь guest не может прочитать файл readfile.c

```
[guest@victoria22 ~]$ su
Пароль:
[root@victoria22 guest]# chown root:guest readfile.c
[root@victoria22 guest]# chmod 700 readfile.c
[root@victoria22 guest]# exit
exit
[guest@victoria22 ~]$ cat readfile.c
cat: readfile.c: Отказано в доступе
[guest@victoria22 ~]$
```

Убедилась, что программа readfile может прочитать файл

```
[guest@victoria22 ~]$ ./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@victoria22 ~]$
```

Выяснила, установлен ли атрибут Sticky на директории /tmp

```
[guest@victoria22 ~]$ ls -l / | grep tmp
drwxrwxrwt. 28 root root 4096 окт  7 17:36 tmp
[guest@victoria22 ~]$ echo "test" > /tmp/file01.txt
[guest@victoria22 ~]$
```

Просмотрела атрибуты у только что созданного файла и разрешила чтение и запись для категории пользователей «все остальные»

```
[guest@victoria22 ~]$ ls -l /tmp/file01.txt
-rw-rw-r--. 1 guest guest 5 окт  7 17:43 /tmp/file01.txt
[guest@victoria22 ~]$ chmod o+rw /tmp/file01.txt
[guest@victoria22 ~]$ ls -l /tmp/file01.txt
-rw-rw-rw-. 1 guest guest 5 окт  7 17:43 /tmp/file01.txt
[guest@victoria22 ~]$
```


Попытка просмотра файла

От пользователя guest2 попробовала прочитать файл /tmp/file01.txt и дозаписать в файл слово test2. Снова проверила содержимое файла

```
[guest2@victoria22 ~]$ cat /tmp/file01.txt
test
[guest2@victoria22 ~]$ echo "test2" > /tmp/file01.txt
[guest2@victoria22 ~]$ cat /tmp/file01.txt
test2
```

От пользователя guest2 записала в файл слово test3, стерев при этом всю имеющуюся в файле информацию командой, а затем прочитала его

```
~~~~~  
[guest2@victoria22 ~]$ echo "test3" > /tmp/file01.txt  
[guest2@victoria22 ~]$ cat /tmp/file01.txt  
test3  
[guest2@victoria22 ~]$
```

Попытка удаления файла

От пользователя guest2 попробовала удалить файл /tmp/file01.txt командой rm

```
[guest2@victoria22 ~]$ rm /tmp/file01.txt  
rm: невозможно удалить «/tmp/file01.txt»: Операция не позволена  
[guest2@victoria22 ~]$
```

/tmp/file01.txt

Повторная попытка удаления файла

Повысила права до суперпользователя и выполнила после этого команду, снимающую атрибут t (Sticky-бит) с директории /tmp. Затем покинула режим суперпользователя и от пользователя guest2 проверьте, что атрибута t у директории /tmp нет. Затем снова попыталась удалить файл

```
[guest2@victoria22 ~]$ su
Пароль:
[root@victoria22 guest2]# chmod -t /tmp
[root@victoria22 guest2]# exit
exit
[guest2@victoria22 ~]$ ls -l / | grep tmp
drwxrwxrwx. 31 root root 4096 окт 7 17:52 tmp
[guest2@victoria22 ~]$ rm /tmp/file01.txt
[guest2@victoria22 ~]$ cat /tmp/file01.txt
cat: /tmp/file01.txt: Нет такого файла или каталога
[guest2@victoria22 ~]$
```

Вывод

В результате выполнения работы я изучила механизм изменения идентификаторов, применения SetUID- и Sticky-битов. Получила практические навыки работы в консоли с дополнительными атрибутами. Рассмотрела работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов