

Презентация по лабораторной работе №5.

Информационная безопасность

Ле Тиен Винь

Информация

- Ле Тиен Винь
- Студент
- Российский университет дружбы народов
- 1032215241@pfur.ru
- <https://github.com/tvle2000/information>



vinh

I.Цель работы

Исследовать механизм изменения идентификаторов с помощью битов SetUID и Sticky. Получить практические навыки работы в консолях с дополнительными свойствами. Рассмотреть механизм изменения идентификатора процесса пользователя, а также влияние бита Sticky на запись и удаление файлов.

II. Выполнение работы

1. Подготовка лабораторного стенда

Установить gcc командой “yum install gcc”

Отключить систему запретов до очередной перезагрузки системы командой “setenforce 0”

```
[root@ltvinh ltvinh]# yum install gcc
Rocky Linux 9 - BaseOS                8.0 kB/s | 4.1 kB      00:00
Rocky Linux 9 - BaseOS                1.2 MB/s | 2.3 MB      00:01
Rocky Linux 9 - AppStream             13 kB/s | 4.5 kB      00:00
Rocky Linux 9 - AppStream             2.6 MB/s | 8.0 MB      00:03
Rocky Linux 9 - Extras                 8.4 kB/s | 2.9 kB      00:00
Package gcc-11.4.1-3.el9.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@ltvinh ltvinh]# setenforce 0
```

II. Выполнение работы

2. Создание программы и исследование

- Создать программу `simpleid.c` от имени пользователя `guest`, которая будет печатать на экране значения UID и GID после запуска
- Скомпилировать программу и выполнить программу
- Сравнить значения UID и GID, результат программы и команды `id` одинаковые.
- Создать программу `simpleid2.c`, которая будет печатать на экране значения действительных идентификаторов
- Скомпилировать и запустить `simpleid2.c` получить

II. Выполнение работы

2. Создание программы и исследование

```
#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;
}
```

```
[guest@ltvinh lab]$ gcc simpleid.c -o simpleid
[guest@ltvinh lab]$ ls
simpleid  simpleid.c
[guest@ltvinh lab]$ ./simpleid
uid=1001, gid=1001
```

```
[guest@ltvinh lab]$ id  
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```



```
#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;
}
```

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

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

II. Выполнение работы

2. Создание программы и исследование

“chown root:guest /home/guest/lab/simpleid2” - команда используется для смены владельца файлов и каталогов. Здесь файл “simpleid2” будет принадлежать пользователю «root» и группе «guest»

“chmod u+s /home/guest/lab/simpleid2” - Бит SetUID устанавливает владельца исполняемого файла. Когда он установлен, файл будет выполняться с идентификатором пользователя владельца файла, а не того, кто его запустил

- Проверять правильность установки новых атрибутов и смены владельца файла simpleid2, здесь владельца является root и группой guest. И атрибут s установлен для пользователя

II. Выполнение работы

2. Создание программы и исследование

Запустить simpleid2 и id, сравнить результат вывода мы увидим они одинаковые

Проделать тоже самое относительно SetGID-бита

Установить SetGID Бит для файла

Проверять правильность установки новых атрибутов и смены владельца файла simpleid2, здесь владельца является root и группой guest. И атрибут s установлен для группы

Запустить simpleid2 и id, сравнить результат вывода мы увидим они одинаковые

Создать и откомпилировать программу readfile.c, которая читать файл

```
[guest@ltvinh lab]$ ls -l simpleid2  
-rwsr-xr-x. 1 root guest 17720 Oct  5 00:31 simpleid2
```

II. Выполнение работы

2. Создание программы и исследование

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

```
[root@ltvinh \ltvinh]# sudo chown root:root /home/guest/lab/readfile  
[root@ltvinh \ltvinh]# sudo chmod u+s /home/guest/lab/readfile
```

```
[guest@ltnvinh lab]$ ls -l simpleid2  
-rwxr-sr-x. 1 root guest 17720 Oct  5 14:57 simpleid2
```



```
[guest@ltvinh lab]$ ./simpleid2
```

```
e_uid=1001, e_gid=1001
```

```
real_uid=1001, real_gid=1001
```

```
[guest@danguen lab]$ id
```

```
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

```
#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;
}
```

II. Выполнение работы

2. Создание программы и исследование

- Сменить владельца у файла `readfile.c`, чтобы только суперпользователь мог прочитать его, а `guest` не мог, и проверять
- Сменить у программы `readfile` владельца и установить SetUID-бит и проверять
- Проверять, может ли программа `readfile` прочитать файл `readfile.c`
- Проверять, может ли программа `readfile` прочитать файл `/etc/shadow`

II. Выполнение работы

2. Создание программы и исследование

```
[root@ltvinh: ~]# chown root:root /home/guest/lab/readfile.c  
[root@ltvinh: ~]# chmod 400 /home/guest/lab/readfile.c
```

```
[guest@ltvinh lab]$ ls -l readfile.c
-r----- . 1 root root 402 Oct  5 00:40 readfile.c
[guest@ltvinh lab]$ cat readfile.c
cat: readfile.c: Permission denied
```

```
[root@ltvinh lab]# chown root:guest /home/guest/lab/readfile  
[root@ltvinh lab]# chmod u+s /home/guest/lab/readfile
```

```
[guest@ltvinh1 lab]$ ls -l readfile
```

```
-rwxr-xr-x. 1 root guest 17664 Oct  5 15:06 readfile
```

```
[guest@ltvinh lab]$ ./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@ltvinh lab]$ ./readfile /etc/shadow
root:$6$4Vd3He7cyG3mMYHw$yaA9iyvUza8xQTUjNbWdQ.6XwKjd1Gs0EoZmD1fkJ37DLdD2K933x86
jHw0zbmlCaWCCISL8CMM0yI92f0tFC.:0:99999:7:::
bin:*:19820:0:99999:7:::
daemon:*:19820:0:99999:7:::
adm:*:19820:0:99999:7:::
lp:*:19820:0:99999:7:::
sync:*:19820:0:99999:7:::
shutdown:*:19820:0:99999:7:::
halt:*:19820:0:99999:7:::
mail:*:19820:0:99999:7:::
operator:*:19820:0:99999:7:::
games:*:19820:0:99999:7:::
ftp:*:19820:0:99999:7:::
nobody:*:19820:0:99999:7:::
systemd-coredump:!!:19970:::
dbus:!!:19970:::
polkitd:!!:19970:::
avahi:!!:19970:::
```

II. Выполнение работы

3. Исследование Sticky-бита

- Проверять установлен ли атрибут Sticky на директории /tmp командой `"ls -l / | grep tmp"`
- От имени пользователя guest создать файл file01.txt в директории /tmp со словом test
- Разрешить file01.txt прав чтения и записи для категории пользователей «все остальные»
- От пользователя guest2 (не является владельцем) попробовать прочитать файл /tmp/file01.txt
- От пользователя guest2 попробовать дозаписать в файл /tmp/file01.txt слово test2, и нам не удалось выполнить операцию

II. Выполнение работы

3. Исследование Sticky-бита

```
[root@ltnvinh danguen]# ls -l / | grep tmp  
drwxrwxrwt. 15 root root 4096 Oct  5 00:52 tmp
```

```
[guest@ltvinh lab]$ echo "test" > /tmp/file01.txt  
[guest@ltvinh lab]$ ls -l /tmp/file01.txt  
-rw-r--r--. 1 guest guest 5 Oct  5 00:54 /tmp/file01.txt
```

```
[guest@ltvinhlab]$ chmod o+rw /tmp/file01.txt
```

```
[guest@ltvinhlab]$ ls -l /tmp/file01.txt
```

```
-rw-r--rw-. 1 guest guest 5 Oct  5 00:54 /tmp/file01.txt
```

```
[guest2@ltvinh ltvinh]$ cat /tmp/file01.txt  
test
```

```
guest2@ltvinh [ltvinh]$ echo "test2" > /tmp/file01.txt
ash: /tmp/file01.txt: Permission denied
guest2@ltvinh [ltvinh]$ cat /tmp/file01.txt
est
```

II. Выполнение работы

3. Исследование Sticky-бита

- От пользователя guest2 попробовать удалить файл /tmp/file01.txt, и нам не удалось выполнить операцию
- Снимать атрибут t (Sticky-бит) с директории /tmp от имени суперпользователя
- От пользователя guest2 проверять, что атрибута t у директории /tmp нет
- Снова от пользователя guest2 попробовать дозаписать в файл /tmp/file01.txt слово test2, и нам не удалось выполнить операцию
- Снова от пользователя guest2 попробовать удалить файл /tmp/file01.txt, и нам удалось выполнить операцию
- Вернуть атрибут t на директорию /tmp от имени суперпользователя

II. Выполнение работы

3. Исследование Sticky-бита

```
[guest2@ltvinh ltvinh]$ rm /tmp/file01.txt  
rm: remove write-protected regular file '/tmp/file01.txt'? y  
rm: cannot remove '/tmp/file01.txt': Operation not permitted
```

```
[root@ltvinh ltvinh]# chmod -t /tmp  
[root@ltvinh ltvinh]# exit  
exit
```

```
[guest2@ltvinh ltvinh]$ ls -l / | grep tmp  
drwxrwxrwx. 15 root root 4096 Oct  5 00:57 tmp
```

```
[root@ltvinh ltvinh]$ echo "test2" > /tmp/file01.txt  
bash: /tmp/file01.txt: Permission denied
```

```
[root@ltvinh ltvinh]# chmod +t /tmp
[root@ltvinh ltvinh]# ls -l / | grep tmp
drwxrwxrwt. 16 root root 4096 Oct  5 15:21 tmp
```

III. Вывод

После лабораторной работы я получил практические навыки работы в консолях с дополнительными свойствами.