

Лабораторная №5

Основы информационной безопасности

Жибицкая Е.Д.

Российский университет дружбы народов, Москва, Россия

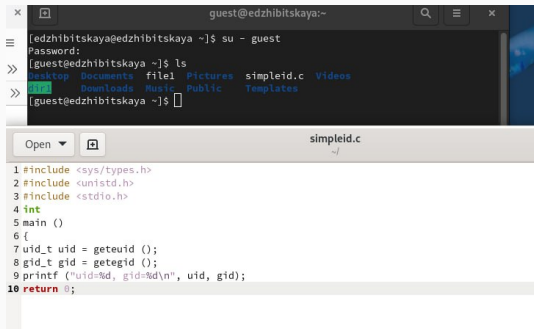
Цель

- Продолжение работы на ОС Rocky. Изучение механизмов изменения идентификаторов, применения SetUID- и Sticky-битов и рассмотрение работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

Ход работы

```
[edzhibitskaya@edzhibitskaya ~]$ gcc -v
Using built-in specs.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/usr/libexec/gcc/x86_64-redhat-linux/11/lto-wrapper
OFFLOAD_TARGET_NAMES=nvptx-none
OFFLOAD_TARGET_DEFAULT=1
Target: x86_64-redhat-linux
Configured with: ../configure --enable-bootstrap --enable-host-pie --enable-host
-bind-now --enable-languages=c,c++,fortran,lto --prefix=/usr --mandir=/usr/share
/man --infodir=/usr/share/info --with-bugurl=https://bugs.rockylinux.org/ --enab
le-shared --enable-threads=posix --enable-checking=release --with-system-zlib --
enable-_cxa_atexit --disable-libunwind-exceptions --enable-gnu-unique-object --
enable-linker-build-id --with-gcc-major-version-only --enable-plugin --enable-in
itfini-array --without-isl --enable-multilib --with-linker-hash-style=gnu --enab
le-offload-targets=nvptx-none --without-cuda-driver --enable-gnu-indirect-functi
on --enable-cet --with-tune=generic --with-arch_64=x86-64-v2 --with-arch_32=x86-
64 --build=x86_64-redhat-linux --with-build-config=bootstrap-lto --enable-link-s
erialization=1
Thread model: posix
Supported LTO compression algorithms: zlib zstd
gcc version 11.5.0 20240719 (Red Hat 11.5.0-5) (GCC)
[edzhibitskaya@edzhibitskaya ~]$ setenforce 0
setenforce: security_setenforce() failed: Permission denied
[edzhibitskaya@edzhibitskaya ~]$ sudo setenforce 0
[sudo] password for edzhibitskaya:
[edzhibitskaya@edzhibitskaya ~]$ getenforce
Permissive
[edzhibitskaya@edzhibitskaya ~]$
```

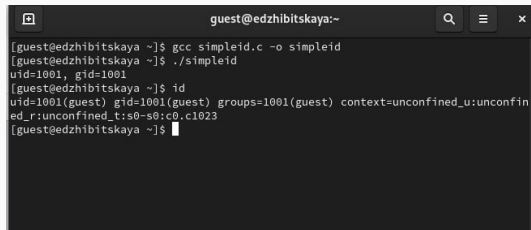
Рис. 1: Проверка gcc



The screenshot shows a terminal window with the prompt `guest@edzhbitskaya:~`. The user enters `su - guest` and provides a password. Then, they enter `ls` and see a list of files including `simpleid.c`. Below the terminal, a file editor window titled `simpleid.c` is open, showing the following code:

```
1 #include <sys/types.h>
2 #include <unistd.h>
3 #include <stdio.h>
4 int
5 main ()
6 {
7     uid_t uid = geteuid ();
8     gid_t gid = getegid ();
9     printf ("uid=%d, gid=%d\n", uid, gid);
10 return 0;
```

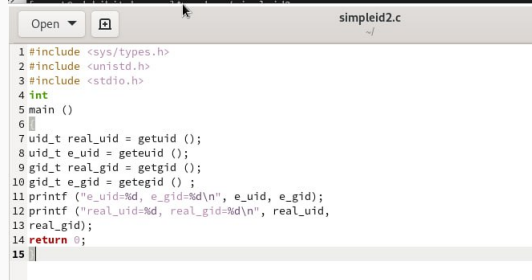
Рис. 2: Создание файла simpleid.c



The screenshot shows a terminal window with the prompt `guest@edzhbitskaya:~`. The user enters `gcc simpleid.c -o simpleid` to compile the file. Then, they enter `./simpleid` to run it. The output shows the user's UID and GID: `uid=1001, gid=1001`. Finally, they enter `id` to see their full identity, which is `uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023`.

Рис. 3: Запуск файла

```
[guest@edzhibitskaya ~]$ gcc simpleid2.c -o simpleid2
[guest@edzhibitskaya ~]$ ./simpleid2.c
-bash: ./simpleid2.c: Permission denied
```



```
1 #include <sys/types.h>
2 #include <unistd.h>
3 #include <stdio.h>
4 int
5 main ()
6 {
7     uid_t real_uid = getuid ();
8     uid_t e_uid = geteuid ();
9     gid_t real_gid = getgid ();
10    gid_t e_gid = getegid ();
11    printf ("e_uid=%d, e_gid=%d\n", e_uid, e_gid);
12    printf ("real_uid=%d, real_gid=%d\n", real_uid,
13    real_gid);
14    return 0;
15 }
```

Рис. 4: Файл simpleid2.c

Добавление прав и владельца

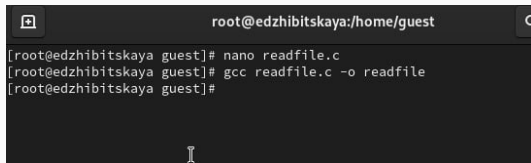
```
[root@edzhibitskaya ~]# chown root:guest /home/guest/simpleid2  
[root@edzhibitskaya ~]# chmod u+s /home/guest/simpleid2
```

Рис. 5: Работа с правами

```
root@edzhibitskaya:/home/guest  
[root@edzhibitskaya guest]# ls -l simpleid2  
-rwsr-xr-x. 1 root guest 17704 Mar 6 14:38 simpleid2  
[root@edzhibitskaya guest]# ./simpleid2  
e_uid=0, e_gid=0  
real_uid=0, real_gid=0  
[root@edzhibitskaya guest]# id  
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023  
[root@edzhibitskaya guest]#
```

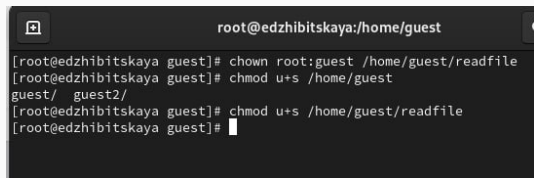
Рис. 6: Исполнение simpleid2.c

Создадим файл, предназначенный для считывания файлов, вставим код и скомпилируем его

A terminal window with a dark background. The title bar shows a window icon, the text 'root@edzhbitskaya:/home/guest', and a search icon. The terminal contains three lines of text: '[root@edzhbitskaya guest]# nano readfile.c', '[root@edzhbitskaya guest]# gcc readfile.c -o readfile', and '[root@edzhbitskaya guest]#'. A cursor is visible on the line following the last command.

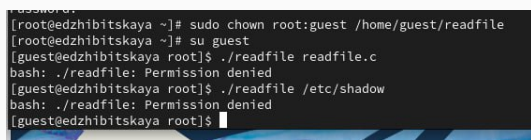
```
root@edzhbitskaya:/home/guest
[root@edzhbitskaya guest]# nano readfile.c
[root@edzhbitskaya guest]# gcc readfile.c -o readfile
[root@edzhbitskaya guest]#
```

Рис. 7: Файл readfile



```
root@edzhibitskaya:/home/guest
[root@edzhibitskaya guest]# chown root:guest /home/guest/readfile
[root@edzhibitskaya guest]# chmod u+s /home/guest
guest/ guest2/
[root@edzhibitskaya guest]# chmod u+s /home/guest/readfile
[root@edzhibitskaya guest]#
```

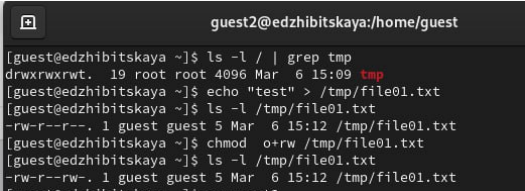
Рис. 8: Работа с правами



```
password:
[root@edzhibitskaya ~]# sudo chown root:guest /home/guest/readfile
[root@edzhibitskaya ~]# su guest
[guest@edzhibitskaya root]$ ./readfile readfile.c
bash: ./readfile: Permission denied
[guest@edzhibitskaya root]$ ./readfile /etc/shadow
bash: ./readfile: Permission denied
[guest@edzhibitskaya root]$
```

Рис. 9: Запуск

Сначала проверим установлен ли на директорию stiky-бит, запишем в него сообщение. Посмотрим на установленные права, разрешим чтение и запись для всех остальных пользователей.



```
guest2@edzhbitskaya:/home/guest  
[guest@edzhbitskaya ~]$ ls -l / | grep tmp  
drwxrwxrwt. 19 root root 4096 Mar  6 15:09 tmp  
[guest@edzhbitskaya ~]$ echo "test" > /tmp/file01.txt  
[guest@edzhbitskaya ~]$ ls -l /tmp/file01.txt  
-rw-r--r--. 1 guest guest 5 Mar  6 15:12 /tmp/file01.txt  
[guest@edzhbitskaya ~]$ chmod o+rw /tmp/file01.txt  
[guest@edzhbitskaya ~]$ ls -l /tmp/file01.txt  
-rw-r--rw-. 1 guest guest 5 Mar  6 15:12 /tmp/file01.txt
```

Рис. 10: Sticky-бит

Выполнение команд от guest2

```
[guest@edzhbitskaya ~]$ su guest2
Password:
[guest2@edzhbitskaya guest]$ cat /tmp/file01.txt
test
[guest2@edzhbitskaya guest]$ echo "test2" > /tmp/file01.txt
bash: /tmp/file01.txt: Permission denied
[guest2@edzhbitskaya guest]$ echo "test3" > /tmp/file01.txt
bash: /tmp/file01.txt: Permission denied
[guest2@edzhbitskaya guest]$ cat /tmp/file01.txt
test
[guest2@edzhbitskaya guest]$
```

Рис. 11: Чтение от guest2

От пользователя guest2 прочитаем файл, попробуем записать туда текст(безуспешно) Попробуем удалить файл - также безуспешно.

```
[guest2@edzhbitskaya guest]$ su -  
Password:  
[root@edzhbitskaya ~]# chmod -t /tmp  
[root@edzhbitskaya ~]# exit  
logout  
[guest2@edzhbitskaya guest]$ ls -l / | grep tmp  
drwxrwxrwx. 18 root root 4096 Mar  6 15:18 tmp  
[guest2@edzhbitskaya guest]$ echo "test2" > /tmp/file01.txt  
bash: /tmp/file01.txt: Permission denied  
[guest2@edzhbitskaya guest]$ echo "test3" > /tmp/file01.txt  
bash: /tmp/file01.txt: Permission denied  
[guest2@edzhbitskaya guest]$ rm /tmp/file01.txt  
rm: remove write-protected regular file '/tmp/file01.txt'? y
```

Рис. 12: Удаление файла без sticky-бита

```
[guest2@edzhibitskaya guest]$ su -  
Password:  
[root@edzhibitskaya ~]# chmod +t /tmp  
[root@edzhibitskaya ~]# exit  
logout  
[guest2@edzhibitskaya guest]$ ls -l / | grep tmp  
drwxrwxrwt. 20 root root 4096 Mar  6 15:19 tmp  
[guest2@edzhibitskaya guest]$
```

Рис. 13: Возвращение sticky-бита

Выводы

- В ходе работы были изучены механизмы изменения идентификаторов, применения SetUID- и Sticky-битов а также влияние бита Sticky на запись и удаление файлов.