

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ЛАБОРАТОРНОЙ РАБОТЕ № 5

дисциплина: Информационная безопасность

Студент: Нгуен Фыок Дат Группа: НФИбд-01-20

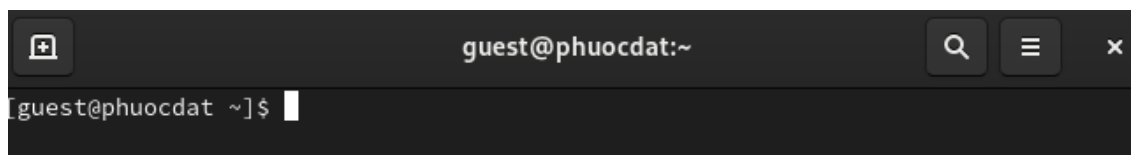
МОСКВА

2023 г

Постановка задачи 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.

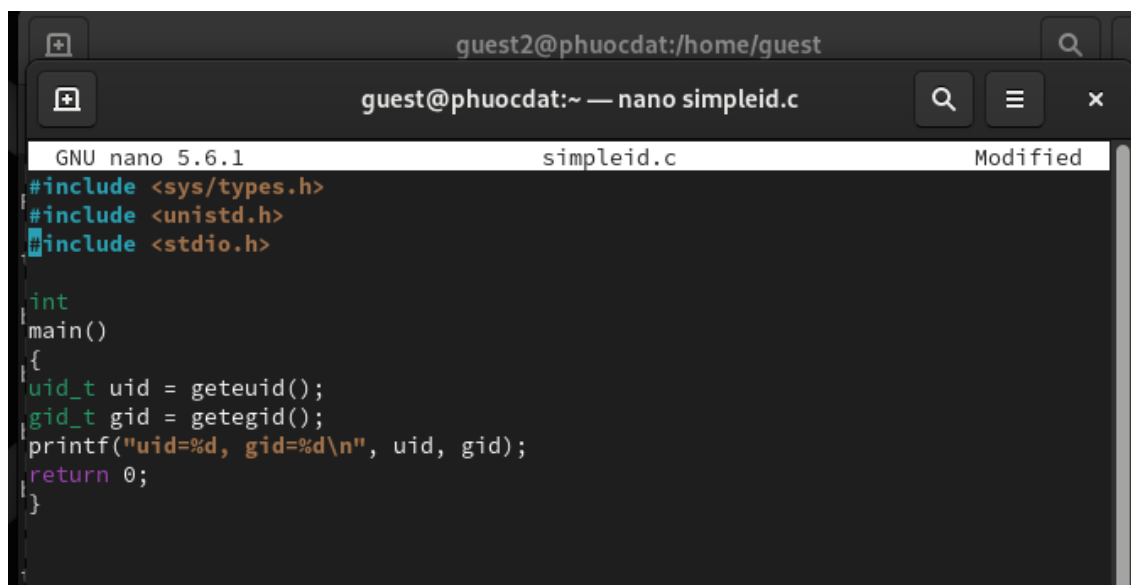
Выполнение работы Создание программы

1. Войду в систему от имени пользователя guest



```
guest@phuocdat:~  
[guest@phuocdat ~]$
```

2. Создам программу simpleid.c



```
GNU nano 5.6.1 simpleid.c Modified  
#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;  
}
```

3. Скомпилирую программу командой `gcc simpleid.c -o simpleid` и удостоверюсь, что файл программы создан

```
[guest@phuocdat ~]$ gcc simpleid.c -o simpleid
```

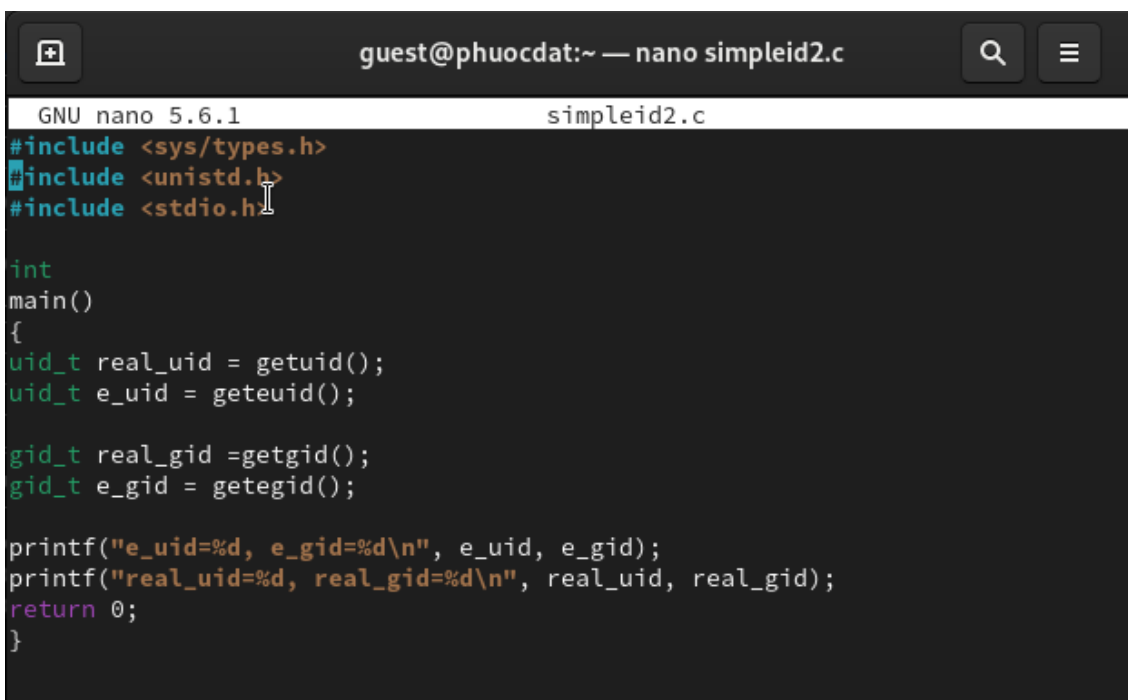
4. Выполню программу `simpleid` командой `./simpleid`

```
[guest@phuocdat ~]$ ./simpleid
uid=1002, gid=1002
```

5. Выполню системную программу `id` командой `id`. Результат совпадает

```
uid=1002 gid=1002
[guest@phuocdat ~]$ id
uid=1002(guest) gid=1002(guest) groups=1002(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

6. Усложню программу, добавив вывод действительных идентификаторов. Создам новый файл `simpleid2.c`



```
guest@phuocdat:~ — nano simpleid2.c
GNU nano 5.6.1 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;
}
```

7. Скомпилирую и запущу `simpleid2.c`

```
[guest@phuocdat ~]$ gcc simpleid2.c -o simpleid2
[guest@phuocdat ~]$ ./simpleid2
e_uid=1002, e_gid=1002
real_uid=1002, real_gid=1002
```

8. Работа с e SetUID-битом

9. 1. От имени суперпользователя выполняю команды: `chown root:guest /home/guest/simpleid2`
`chmod u+s /home/guest/simpleid2`

```
[guest@phuocdat ~]$ su
Password:
[root@phuocdat guest]# chown root:guest /home/guest/simpleid2
[root@phuocdat guest]# chmod u+s /home/guest/simpleid2
```

8.2 Команда `chown root:guest /home/guest/simpleid2` меняет владельца файла. Команда `chmod u+s /home/guest/simpleid2` меняет права доступа к файлу.

8.3 Проверю правильность установки новых атрибутов и смены владельца файла `simpleid2` командой: `ls -l simpleid2`

```
[root@phuocdat guest]# ls -l simpleid2
-rwsr-xr-x. 1 root guest 26064 Oct  7 20:31 simpleid2
```

8.4 Запущу `simpleid2` и `id`, команды: `./simpleid2` и `id`

```
[root@phuocdat guest]# ls -l simpleid2
-rwsr-xr-x. 1 root guest 26064 Oct  7 20:31 simpleid2
[root@phuocdat guest]# ./simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
```

После выполнения команд изменился параметр `e_uid`.

9. Прделаю тоже самое относительно SetGID-бита
10. 1 От имени суперпользователя выполняю команды: `chmod u-s /home/guest/simpleid2` - чтобы отменить изменения на прошлом шаге `chmod g+s /home/guest/simpleid2`

```
[root@phuocdat guest]# chmod u-s /home/guest/simpleid2
[root@phuocdat guest]# chmod g+s /home/guest/simpleid2
```

9.2 Проверю правильность установки новых атрибутов и смены владельца файла `simpleid2` командой: `ls -l simpleid2`

```
[root@phuocdat guest]# ls -l simpleid2
-rwxr-sr-x. 1 root guest 26064 Oct  7 20:31 simpleid2
```

9.3 Запущу `simpleid2` и `id`, команды: `./simpleid2` и `id`. Ничего не изменилось.

```
[root@phuocdat guest]# ./simpleid2
e_uid=0, e_gid=1002
real_uid=0, real_gid=0
[root@phuocdat guest]# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

10. Создам программу readfile.c

```
guest@phuocdat:/home/guest — nano readfile.c
GNU nano 5.6.1 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;
}
```

11. Скомпилирую её командой: gcc readfile.c -o readfile

```
[root@phuocdat guest]# gcc readfile.c -o readfile
```

12. Сменю владельца у файла readfile.c и изменю права так, чтобы только суперпользователь (root) мог прочитать его, а guest не мог.

```
[root@phuocdat guest]# chown root:guest /home/guest/readfile.c
[root@phuocdat guest]# chmod 700 /home/guest/readfile.c
```

13. Проверю, что пользователь guest не может прочитать файл readfile.c.

```
~~~~~[guest@phuocdat ~]$ ls -l readfile.c
-rwx-----. 1 root guest 397 Oct  7 20:51 readfile.c
[guest@phuocdat ~]$ cat readfile.c
cat: readfile.c: Permission denied
```

14. Сменю у программы readfile владельца и установлю SetUID-бит.

```
[root@phuocdat guest]# chown root:guest /home/guest/readfile.c
[root@phuocdat guest]# chmod 700 /home/guest/readfile.c
```

15. Проверю, может ли программа readfile прочитать файл readfile.c


```
[guest@phuocdat ~]$ ls -l / | grep tmp
drwxrwxrwt. 16 root root 4096 Oct  7 20:59 tmp
```

2. От имени пользователя guest создам файл file01.txt в директории /tmp со словом test: `echo "test" > /tmp/file01.txt`

```
[guest@phuocdat ~]$ echo "test" > /tmp/file01.txt
[guest@phuocdat ~]$ cat /tmp/file01.txt
test
```

3. Просмотрю атрибуты у только что созданного файла и разрешу чтение и запись для категории пользователей «все остальные»: `ls -l /tmp/file01.txt chmod o+rw /tmp/file01.txt ls -l /tmp/file01.txt`

```
[guest@phuocdat ~]$ ls -l /tmp/file01.txt
-rw-r--r--. 1 guest guest 5 Oct  7 21:01 /tmp/file01.txt
[guest@phuocdat ~]$ chmod o+rw /tmp/file01.txt
[guest@phuocdat ~]$ ls -l /tmp/file01.txt
-rw-r--rw-. 1 guest guest 5 Oct  7 21:01 /tmp/file01.txt
```

4. От пользователя guest2 (не являющегося владельцем) попробую прочитать файл /tmp/file01.txt: `cat /tmp/file01.txt`

```
[guest@phuocdat ~]$ su guest2
Password:
[guest2@phuocdat guest]$ cat /tmp/file01.txt
test
```

5. От пользователя guest2 попробую дозаписать в файл /tmp/file01.txt слово test2 командой `echo "test2" >> /tmp/file01.txt`

```
[guest2@phuocdat guest]$ echo "test2" >> /tmp/file01.txt
```

Мне удалось выполнить операцию. 6. Проверю содержимое файла командой `cat /tmp/file01.txt`

```
[guest2@phuocdat guest]$ cat /tmp/file01.txt
test
test2
```

7. От пользователя guest2 попробую записать в файл /tmp/file01.txt слово test3, стерев при этом всю имеющуюся в файле информацию командой `echo "test3" > /tmp/file01.txt`

```
[guest2@phuocdat guest]$ echo "test3" > /tmp/file01.txt
```

Мне удалось выполнить операцию.

8. Проверю содержимое файла командой `cat /tmp/file01.txt`

```
[guest2@phuocdat guest]$ cat /tmp/file01.txt  
test3
```

Мне не удалось удалить файл. 10. Повышу свои права до суперпользователя следующей командой `su` и выполню после этого команду, снимающую атрибут `t` (Sticky-бит) с директории `/tmp`: `chmod -t /tmp`

```
[guest2@phuocdat guest]$ su  
Password:  
[root@phuocdat guest]# chmod -t /tmp
```

11. Покину режим суперпользователя командой `exit`

```
[root@phuocdat guest]# exit  
exit
```

12. От пользователя `guest2` проверьте, что атрибута `t` у директории `/tmp` нет: `ls -l / | grep tmp`

```
[guest2@phuocdat guest]$ ls -l / | grep tmp  
drwxrwxrwx. 16 root root 4096 Oct  7 21:20 tmp
```

13. Повторю предыдущие шаги.

```
[guest2@phuocdat guest]$ cat /tmp/file01.txt
test3
[guest2@phuocdat guest]$ echo "test2" >> /tmp/file01.txt
[guest2@phuocdat guest]$ cat /tmp/file01.txt
test3
test2
[guest2@phuocdat guest]$ echo "test3" > /tmp/file01.txt
[guest2@phuocdat guest]$ cat /tmp/file01.txt
test3
[guest2@phuocdat guest]$ rm /tmp/file01.txt
[guest2@phuocdat guest]$ ls /tmp
systemd-private-00332518211c41888d5ee0ae0f6865bf-chronyd.service-zptRmp
systemd-private-00332518211c41888d5ee0ae0f6865bf-colord.service-tqErob
systemd-private-00332518211c41888d5ee0ae0f6865bf-dbus-broker.service-LfD8fD
systemd-private-00332518211c41888d5ee0ae0f6865bf-fwupd.service-9NU4Uq
systemd-private-00332518211c41888d5ee0ae0f6865bf-ModemManager.service-8ItbIj
systemd-private-00332518211c41888d5ee0ae0f6865bf-power-profiles-daemon.service-p2UnGR
systemd-private-00332518211c41888d5ee0ae0f6865bf-rtkit-daemon.service-9btTUQ
systemd-private-00332518211c41888d5ee0ae0f6865bf-switcheroo-control.service-oozOMk
systemd-private-00332518211c41888d5ee0ae0f6865bf-systemd-logind.service-zLLnz1
systemd-private-00332518211c41888d5ee0ae0f6865bf-upower.service-Z2xpJ6
```

14. Мне удалось удалить файл от имени пользователя, не являющегося его владельцем. Это связано с тем, что Sticky-bit позволяет защищать файлы от случайного удаления, когда несколько пользователей имеют права на запись в один и тот же каталог. Если у файла атрибут t стоит, значит пользователь может удалить файл, только если он является пользователем-владельцем файла или каталога, в котором содержится файл. Если же этот атрибут не установлен, то удалить файл могут все пользователи, которым позволено удалять файлы из каталога.
15. Повышу свои права до суперпользователя и верну атрибут t на директорию /tmp: su
chmod +t /tmp exit

```
[guest2@phuocdat guest]$ su
Password:
[root@phuocdat guest]# chmod +t /tmp
[root@phuocdat guest]# ls -l / | grep tmp
drwxrwxrwt. 17 root root 4096 Oct  7 21:22 tmp
[root@phuocdat guest]# exit
exit
```

Заключение В ходе данной лабораторной работы я изучила механизмы изменения идентификаторов, применения SetUID-, SetGID- и Sticky-битов. Рассмотрела работ механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.