

Презентация к лабораторной работе 5

Основы информационной безопасности

Хрусталеv Влад Николаевич

Российский университет дружбы народов им. Патриса Лумумбы, Москва, Россия

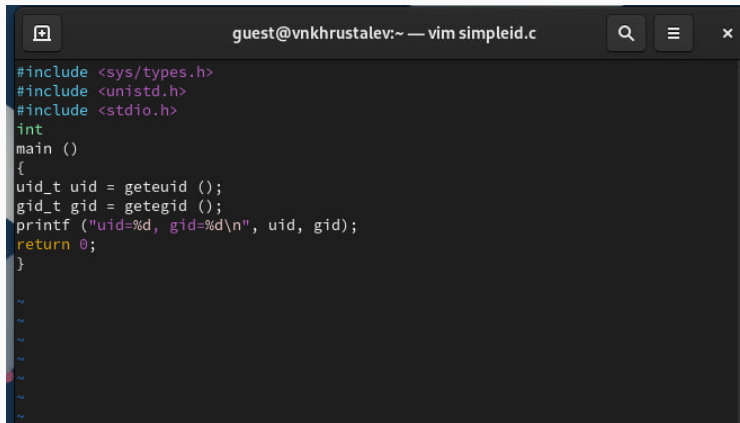
Изучение механизмов изменения идентификаторов, применения SetUID- и Sticky-битов. Получение практических навыков работы в консоли с дополнительными атрибутами. Рассмотрение работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов

Создание файла simpleid.c

```
[guest@vnkhrustalev ~]$ touch simpleid.c
[guest@vnkhrustalev ~]$ ls
dir1 Documents Pictures simpleid.c
[guest@vnkhrustalev ~]$ vim simpleid.c
```

Рис. 1: Создание файла simpleid.c

Создание(содержание) файла simpleid.c



```
guest@vnkhrustalev:~ — vim 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;
}

~
~
~
~
~
~
```

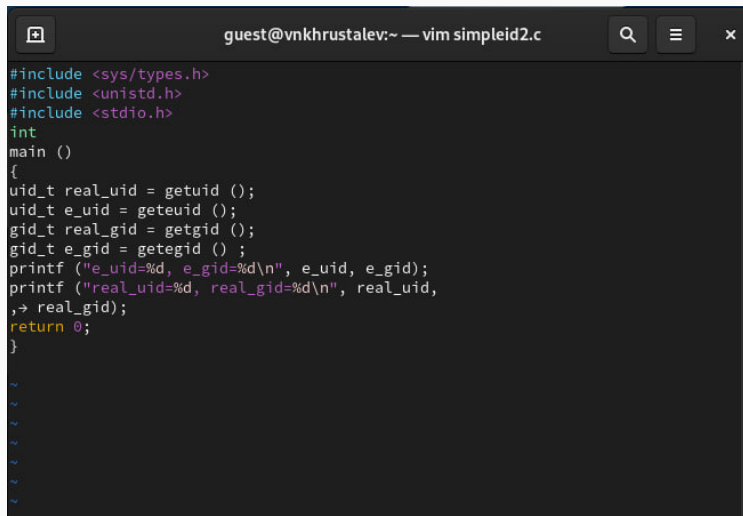
Рис. 2: Создание(содержание) файла simpleid.c

Компилирование программы simpleid.c и сравнение её работы с командой id

```
[guest@vnkhrustalev ~]$ gcc simpleid.c -o simpleid
[guest@vnkhrustalev ~]$ ls
dir1 Documents Pictures simpleid simpleid.c
[guest@vnkhrustalev ~]$ ./simpleid
uid=1007, gid=100
[guest@vnkhrustalev ~]$ id
uid=1007(guest) gid=100(users) группы=100(users) контекст=unconfined_u:unconfine
d_r:unconfined_t:s0-s0:c0.c1023
[guest@vnkhrustalev ~]$
```

Рис. 3: Компилирование программы simpleid.c и сравнение её работы с командой id

Создание(содержание) файла simpleid2.c



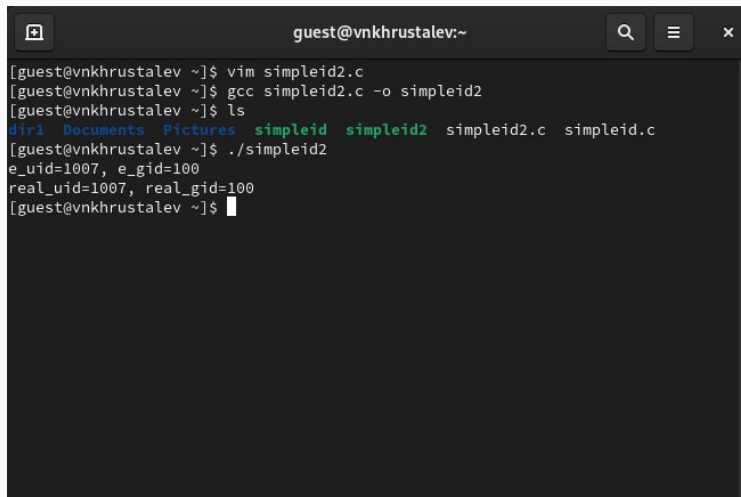
```
guest@vnkhrustalev:~ — vim simpleid2.c

#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: Создание(содержание) файла simpleid2.c

Компилирование программы simpleid2.c и запуск



```
guest@vnkhrustalev:~  
[guest@vnkhrustalev ~]$ vim simpleid2.c  
[guest@vnkhrustalev ~]$ gcc simpleid2.c -o simpleid2  
[guest@vnkhrustalev ~]$ ls  
dir1 Documents Pictures simpleid simpleid2 simpleid2.c simpleid.c  
[guest@vnkhrustalev ~]$ ./simpleid2  
e_uid=1007, e_gid=100  
real_uid=1007, real_gid=100  
[guest@vnkhrustalev ~]$
```

Рис. 5: Компилирование программы simpleid2.c и запуск

Изменение прав и владельца simpleid2

```
[vnkhrustalev@vnkhrustalev guest]$ sudo chown root:guest /home/guest/simpleid2  
[vnkhrustalev@vnkhrustalev guest]$ sudo chmod u+s /home/guest/simpleid2  
[vnkhrustalev@vnkhrustalev guest]$
```

Рис. 6: Изменение прав и владельца simpleid2

Проверка прав у simpleid2 + сравнение вывода программы с командой id

```
[guest@vnkhrustalev ~]$ ls -l simpleid2
-rwsr-xr-x. 1 root guest 26048 anp 13 15:46 simpleid2
[guest@vnkhrustalev ~]$ ./simpleid2
e_uid=0, e_gid=100
real_uid=1007, real_gid=100
[guest@vnkhrustalev ~]$ id
uid=1007(guest) gid=100(users) группы=100(users) контекст=unconfined_u:unconfine
d_r:unconfined_t:s0-s0:c0.c1023
[guest@vnkhrustalev ~]$
```

Рис. 7: Проверка прав у simpleid2 + сравнение вывода программы с командой id

Создание(содержание) файла readfile.c



```
#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;
}

~
~
:wq
```

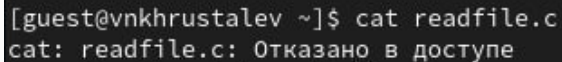
Рис. 8: Создание(содержание) файла readfile.c

Изменение владельца и изменение прав доступа к файлу readfile.c

```
[root@vnkhrustalev guest]# sudo chown root readfile.c
[root@vnkhrustalev guest]# sudo chmod 400 readfile.c
[root@vnkhrustalev guest]# ls
dir1      Pictures  readfile.c  simpleid2  simpleid.c
Documents readfile  simpleid    simpleid2.c
```

Рис. 9: Изменение владельца и изменение прав доступа к файлу readfile.c

Попытка чтения файла readfile.c от стороннего пользователя

A terminal window with a dark background. The prompt is [guest@vnkhrustalev ~]. The user enters the command cat readfile.c. The output is cat: readfile.c: Отказано в доступе.

```
[guest@vnkhrustalev ~]$ cat readfile.c  
cat: readfile.c: Отказано в доступе
```

Рис. 10: Попытка чтения файла readfile.c от стороннего пользователя

Изменение прав доступа к файлу readfile(u+s)

```
[guest@vnkhrustalev ~]$ su
Пароль:
[root@vnkhrustalev guest]# sudo chown root readfile
[root@vnkhrustalev guest]# sudo chmod u+s readfile
[root@vnkhrustalev guest]#
```

Рис. 11: Изменение прав доступа к файлу readfile(u+s)

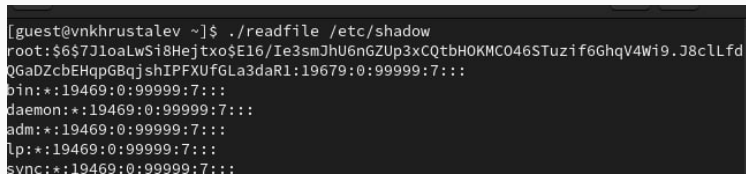
Попытка чтения файла readfile.c через программу readfile



```
[root@vnkhurstalev guest]# ./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;
}
```

Рис. 12: Попытка чтения файла readfile.c через программу readfile

Попытка чтения файла /etc/shadow через программу readfile



```
[guest@vnkhrustalev ~]$ ./readfile /etc/shadow
root:$6$7JloaLwSi8Hejtxo$E16/Ie3smJhU6nGZUp3xCQtbH0KMC046Stuzif6GhqV4Wi9.J8clLfd
QGadZcbEHqp6BqjshIPFXUfGLa3daR1:19679:0:99999:7:::
bin:!:19469:0:99999:7:::
daemon:!:19469:0:99999:7:::
adm:!:19469:0:99999:7:::
lp:!:19469:0:99999:7:::
sync:!:19469:0:99999:7:::
```

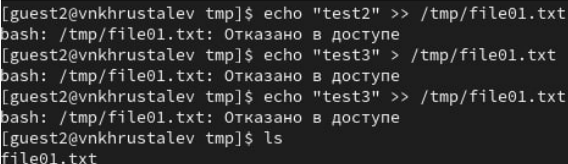
Рис. 13: Попытка чтения файла /etc/shadow через программу readfile

Создание файла /tmp/file.01 и изменение прав от имени пользователя guest

```
[root@vnkhrustalev tmp]# su guest
[guest@vnkhrustalev tmp]$ ls -l / | grep tmp
drwxrwxrwt. 15 root root 4096 anp 13 16:01 tmp
[guest@vnkhrustalev tmp]$ echo "test" > /tmp/file01.txt
[guest@vnkhrustalev tmp]$ ls -l /tmp/file01.txt
-rw-r--r--. 1 guest users 5 anp 13 16:01 /tmp/file01.txt
[guest@vnkhrustalev tmp]$ chmod o+rw /tmp/file01.txt
[guest@vnkhrustalev tmp]$ ls -l /tmp/file01.txt
-rw-r--rw-. 1 guest users 5 anp 13 16:01 /tmp/file01.txt
[guest@vnkhrustalev tmp]$
```

Рис. 14: Создание файла /tmp/file.01 и изменение прав от имени пользователя guest

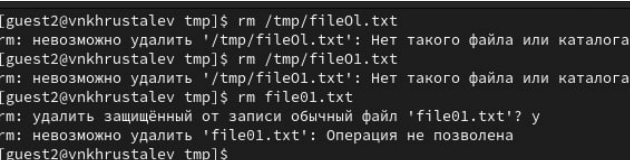
Попытка дописать(изменить) содержание файла /tmp/file.01 от другого пользователя

A terminal window with a dark background and light-colored text. It shows a series of commands and their outputs. The user is 'guest2@vnkhrustalev' in the 'tmp' directory. They attempt to append 'test2' to '/tmp/file01.txt', which fails with 'bash: /tmp/file01.txt: Отказано в доступе'. They then attempt to overwrite 'test3' to the same file, which also fails with the same message. Finally, they attempt to append 'test3' again, which also fails. The last command is 'ls', which lists 'file01.txt'.

```
[guest2@vnkhrustalev tmp]$ echo "test2" >> /tmp/file01.txt
bash: /tmp/file01.txt: Отказано в доступе
[guest2@vnkhrustalev tmp]$ echo "test3" > /tmp/file01.txt
bash: /tmp/file01.txt: Отказано в доступе
[guest2@vnkhrustalev tmp]$ echo "test3" >> /tmp/file01.txt
bash: /tmp/file01.txt: Отказано в доступе
[guest2@vnkhrustalev tmp]$ ls
file01.txt
```

Рис. 15: Попытка дописать(изменить) содержание файла /tmp/file.01 от другого пользователя

Попытка удалить файл /tmp/file.01 от другого пользователя



```
[guest2@vnkhrustalev tmp]$ rm /tmp/file01.txt
rm: невозможно удалить '/tmp/file01.txt': Нет такого файла или каталога
[guest2@vnkhrustalev tmp]$ rm /tmp/file01.txt
rm: невозможно удалить '/tmp/file01.txt': Нет такого файла или каталога
[guest2@vnkhrustalev tmp]$ rm file01.txt
rm: удалить защищённый от записи обычный файл 'file01.txt'? y
rm: невозможно удалить 'file01.txt': Операция не позволена
[guest2@vnkhrustalev tmp]$
```

Рис. 16: Попытка удалить файл /tmp/file.01 от другого пользователя

Изменение(удаление) Sticky бита папки /tmp и повторные попытки предыдущих манипуляций

```
[root@vnkhrustalev ~]# chmod -t /tmp
[root@vnkhrustalev ~]# exit
выход
[guest2@vnkhrustalev tmp]$ ls -l / | grep tmp
drwxrwxrwx. 18 root root 4096 anp 13 16:12 tmp
[guest2@vnkhrustalev tmp]$ cat /tmp/file01.txt
test3
[guest2@vnkhrustalev tmp]$ echo "test2" > /tmp/file01.txt
bash: /tmp/file01.txt: Отказано в доступе
[guest2@vnkhrustalev tmp]$ echo "test2" >> /tmp/file01.txt
bash: /tmp/file01.txt: Отказано в доступе
[guest2@vnkhrustalev tmp]$ id
uid=1008(guest2) gid=100(users) rгруппы=100(users),1005(guest) контекст=unconfine
d_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest2@vnkhrustalev tmp]$
```

Рис. 17: Изменение(удаление) Sticky бита папки /tmp и повторные попытки предыдущих манипуляций

Возврат Sticky бита папки /tmp на место

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

Рис. 18: Возврат Sticky бита папки /tmp на место

Выводы

Изучение механизмов изменения идентификаторов, применения SetUID- и Sticky-битов, а также получение практических навыков работы в консоли с дополнительными атрибутами позволяют глубже понять принципы безопасности и управления доступом в Unix-подобных системах. Рассмотрение работы механизма смены идентификатора процессов пользователей и влияние бита Sticky на запись и удаление файлов демонстрирует важность этих аспектов для обеспечения безопасности и контроля доступа в многопользовательских средах.