

# Презентация по лабораторной работе №5

Информационная безопасность

---

НгуенФыокДат

7 октября 2023

Российский университет дружбы народов, Москва, Россия

# Информация

---

- НгуенФыокДат
- студент 4 курса группы НФИбд-01-20
- ст. б. 1032195855
- Российский университет дружбы народов
- 1032195855@pfur.ru

# Вводная часть

---

- Обеспечение безопасности

- Приобретение практических навыков работы в консоли с дополнительными атрибутами файлов
- Изучение механизмов смены идентификатора процессов пользователей
- Изучение SetUID-, SetGID- и Sticky-битов

- Веб-сервис GitHub для работы с репозиториями
- Программа для виртуализации ОС VirtualBox
- Процессор pandoc для входного формата Markdown
- Результирующие форматы
  - pdf
  - docx
- Автоматизация процесса создания: Makefile

## Ход работы

---



# Написание программы 1



```
guest@phuocdat:~ — nano 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;
}
```

# Запуск программы 1

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

```
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
```

# Модификация программы



```
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;
}
```

## Запуск модифицированной программы

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

# Изменение владельца файла

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

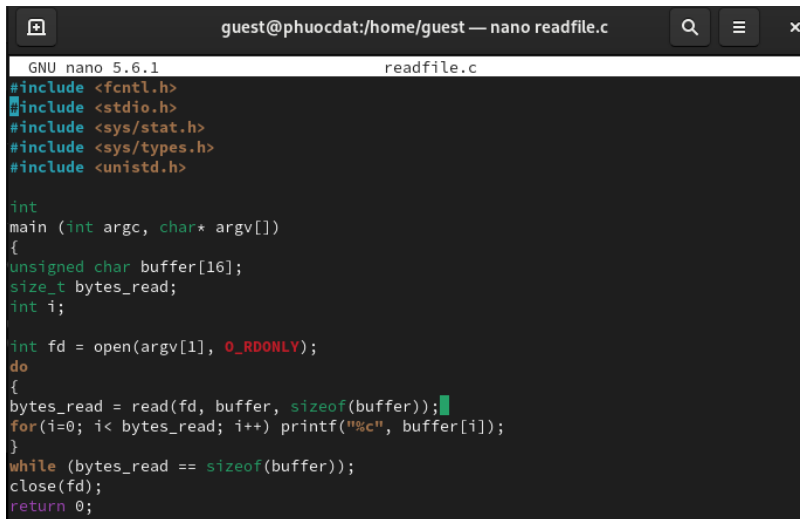
```
[root@phuocdat guest]# ls -l simpleid2
-rwsr-xr-x. 1 root guest 26064 Oct  7 20:31 simpleid2
[root@phuocdat guest]# ./simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
```

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

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

```
[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
```

## Написание программы 3



The screenshot shows a terminal window with the nano text editor open. The title bar indicates the user is 'guest' at 'phuocdat' in the directory '/home/guest', editing the file 'readfile.c'. The editor's status bar shows 'GNU nano 5.6.1' and the filename 'readfile.c'. The code is written in C and includes several header files, a main function, and logic to read a file into a buffer and print its contents character by character.

```
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;
}
```

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

## Проверка наличия Sticky-бита

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

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



## Попытка чтения, изменения и удаления файла

```
[guest@phuocdat ~]$ su guest2
Password:
[guest2@phuocdat guest]$ cat /tmp/file01.txt
test
-----
[guest2@phuocdat guest]$ echo "test2" >> /tmp/file01.txt
-----
[guest2@phuocdat guest]$ cat /tmp/file01.txt
test
test2
-----
```

```
[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
```

## Повторная попытка чтения, записи и удаления файла

```
[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-chrond.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
```

# Результаты

---

- Получены навыки работы с дополнительными атрибутами файлов
- Отточены навыки работы с механизмами смены владельца
- Рассмотрены принципы работы SetUID-, SetGID- и Sticky-битов