

# presentation

## Презентация по лабораторной работе №5

Подъярова Ксения Витальевна

### Цель работы

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

### Выполнение лабораторной работы

1. Для лабораторной работы необходимо проверить, установлен ли компилятор gcc, команда gcc -v позволяет это сделать. Также осуществляется отключение системы запретов с помощью setenforce 0 (рис. 1).

```
[kvpodjhyarova@kvpodjhyarova ~]$ whereis gcc
gcc: /usr/bin/gcc /usr/lib/gcc /usr/libexec/gcc /usr/share/man/man1/gcc.1.gz /usr/share/info/gcc.info.gz
[kvpodjhyarova@kvpodjhyarova ~]$ whereis g++
g++: /usr/bin/g++ /usr/share/man/man1/g++.1.gz
[kvpodjhyarova@kvpodjhyarova ~]$ 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-language-c, lto --prefix=/usr --mandir=/usr/share/man --infodir=/usr/share/info --with-bugurl=https://bugs.rockylinux.
hared --enable-threads=posix --enable-checking=release --with-system-zlib --enable-__cxa_atexit --disable-l
ions --enable-gnu-unique-object --enable-linker-build-id --with-gcc-major-version-only --enable-plugin --en
rray --without-isl --enable-multilib --with-linker-hash-style=gnu --enable-offload-targets=nvptx-none --wit
r --enable-gnu-indirect-function --enable-cet --with-tune=generic --with-arch_64=x86-64-v2 --with-arch_32=x
86_64-redhat-linux --with-build-config=bootstrap-lto --enable-link-serialization=1
Thread model: posix
Supported LTO compression algorithms: zlib zstd
gcc version 11.4.1 20231218 (Red Hat 11.4.1-3) (GCC)
[kvpodjhyarova@kvpodjhyarova ~]$ setenforce 0
setenforce: security_setenforce() failed: Permission denied
[kvpodjhyarova@kvpodjhyarova ~]$ sudo setenforce 0
[sudo] password for kvpodjhyarova:
[kvpodjhyarova@kvpodjhyarova ~]$ getenforce
Permissive
[kvpodjhyarova@kvpodjhyarova ~]$
```

2. Осуществляется вход от имени пользователя guest

3. Создание файла `simplified.c` и запись в файл кода. C++ Листинг 1 `#include` `#include` `#include` `int main () { uid_t uid = geteuid (); gid_t gid = getegid (); printf ("uid=%d, gid=%d\n", uid, gid); return 0; }` Содержимое файла выглядит следующим образом (рис. 3).

```
GNU nano 5.6.1 simplified.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;
}
```

4. Запускаю исполняемый файл. В выводе файла выписаны номера пользователя и групп, от вывода при вводе `if`, они отличаются только тем, что информации меньше

```
[evdvorkina@evdvorkina ~]$ sudo chattr +a /home/guest/dir1/file1
[sudo] пароль для evdvorkina:
[evdvorkina@evdvorkina ~]$
```

5. Создание, запись в файл и компиляция файла `simplified2.c`. Запуск программы

```
[guest@evdvorkina ~]$ lsattr dir1/file1
-----a----- dir1/file1
[guest@evdvorkina ~]$
```

6. С помощью `chown` изменяю владельца файла на суперпользователя, с помощью `chmod` изменяю права доступа

```
$ sudo chown root:guest /home/guest/simplified2
$ sudo chmod u+s /home/guest/simplified2
$ sudo ls -l /home/guest/simplified2
-rwsr-xr-x 1 root:guest 13 03:57 /home/guest/simplified2
$
```

7. Сравнение вывода программы и команды `id`, наша команда снова вывела только ограниченное количество информации

```
[evdvorkina@evdvorkina ~]$ sudo /home/guest/simplified2
uid=0, e_gid=0
real_uid=0, real_gid=0
[evdvorkina@evdvorkina ~]$ id
uid=1000(ev dvorkina) gid=1000(ev dvorkina) rpyнны=1000(ev dvorkina),10(wheel) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[evdvorkina@evdvorkina ~]$ sudo id
uid=0(root) gid=0(root) rpyнны=0(root) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[evdvorkina@evdvorkina ~]$
```

## Создание и компиляция файла readfile.c

```
touch readfile.c
nano readfile.c
nano readfile.c
gcc readfile.c -o readfile
ls
```

```
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;
}
```

8. Снова от имени суперпользователя меняю владельца файла readfile. Далее меняю права доступа так, чтобы пользователь guest не смог прочесть содержимое файла

```
sudo chown root:guest /home/guest/readfile.c
sudo chmod u+s /home/guest/readfile.c
sudo chmod 700 /home/guest/readfile.c
sudo chmod -r /home/guest/readfile.c
sudo chmod u+s /home/guest/readfile.c
```

9. Проверка прочесть файл от имени пользователя guest. Прочесть файл не удастся

```
cat: readfile.c: Отказано в доступе
```

Попытка прочесть тот же файл с помощью программы readfile, в ответ получаем "отказано в доступе". Попытка прочесть файл /etc/shadow с помощью программы, все еще получаем отказ в доступе.

```
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
s-
t-
-
p8
```

10. Пробуем прочесть эти же файлы от имени суперпользователя и чтение файлов проходит успешно

```
root:$6$3reywnb0G.0EfHL7$1td/ZD0qRQQEdBaZehnNr0Kq7lhY9Ns4Ip0CdU6M/hMk8vHf5qs02
gd3/YkGPNmw5AD2t0THlFZYuxi4eD/rU0::0:99999:7:::
bin:!:19469:0:99999:7:::
daemon:!:19469:0:99999:7:::
adm:!:19469:0:99999:7:::
lp:!:19469:0:99999:7:::
```

Добавляю расширенный атрибут `i` от имени суперпользователя, теперь все выполнено верно (рис. 13).

```
~]$ sudo chattr +i /home/guest/dir1/file1
~]$ sudo lsattr /home/guest/dir1/file1
/home/guest/dir1/file1
~]$
```

Пытаюсь записать в файл, дозаписать, переименовать или удалить, ничего из этого сделать нельзя (рис. 14).

```
[guest@evdvorkina ~]$ echo "test" > dir1/file1
bash: dir1/file1: Операция не позволена
[guest@evdvorkina ~]$ echo "test" >> dir1/file1
bash: dir1/file1: Операция не позволена
[guest@evdvorkina ~]$ cat dir1/file1
abcd
[guest@evdvorkina ~]$ mv dir1/file1 dir1/aaa
mv: невозможно переместить 'dir1/file1' в 'dir1/aaa': Операция не позволена
[guest@evdvorkina ~]$ rm dir1/file1
rm: невозможно удалить 'dir1/file1': Операция не позволена
![[guest@evdvorkina ~]$
```

## Выводы

Изучила механизм изменения идентификаторов, применила SetUID- и Sticky-биты. Получила практические навыки работы в консоли с дополнительными атрибутами. Рассмотрела работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.