

Отчёт по лабораторной работе 5

Основы информационной безопасности

Нджову Нелиа

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	15

Список иллюстраций

3.1	Подготовка к лабораторной работе	7
3.2	Подготовка к лабораторной работе	7
3.3	Создание файла	8
3.4	Содержимое файла	8
3.5	Компиляция файла	8
3.6	Сравнение команд	9
3.7	Создание файла	9
3.8	Содержимое файла	9
3.9	Компиляция файла	9
3.10	Смена владельца файла и прав доступа к файлу	10
3.11	Запуск файла	10
3.12	Создание файла	10
3.13	Содержимое файла	11
3.14	Компиляция файла	11
3.15	Смена владельца файла и прав доступа к файлу	11
3.16	Попытка прочесть содержимое файла	12
3.17	Попытка прочесть содержимое файла программой	12
3.18	Попытка прочесть содержимое файла программой	12
3.19	Чтение файла от имени пользователя guest	13
3.20	Проверка атрибутов директории tmp	13
3.21	Создание файла, изменение прав доступа	13
3.22	Попытка записать файла	13
3.23	Смена атрибутов файла	14
3.24	Проверка атрибутов директории	14
3.25	Повтор предыдущих действий	14
3.26	Изменение атрибутов	14

Список таблиц

1 Цель работы

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.

2 Задание

1. Создание программы
2. Исследование Sticky-бита

3 Выполнение лабораторной работы

1. Создание программы

Для этой лабораторной работы, вам необходимо проверить установлен ли компилятор(gcc), используя команду gcc -v. Также осуществляется отключение системы запретом с помощью setenforce 0(рис.1 и рис.2)

```
[Nelianjovu@Nelianjovu ~]$ 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
[Nelianjovu@Nelianjovu ~]$ whereis g++
g++: /usr/bin/g++ /usr/share/man/man1/g++.1.gz
[Nelianjovu@Nelianjovu ~]$ 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
```

Рис. 3.1: Подготовка к лабораторной работе

```
[Nelianjovu@Nelianjovu ~]$ sudo setenforce 0
[sudo] password for Nelianjovu:
[Nelianjovu@Nelianjovu ~]$ getenforce
Permissive
[Nelianjovu@Nelianjovu ~]$
```

Рис. 3.2: Подготовка к лабораторной работе

Я не мог использовать своего guest пользователя, потому что получила сообщение об ошибке когда попробовала установить файл. Из-за этого, я использовала суперпользователя, я создала файл simpled.c и записала в него код(рис.3 и рис.4)

```
[Nelianjovu@Nelianjovu ~]$ touch simplified.c
[Nelianjovu@Nelianjovu ~]$ nano simplified.c
[Nelianjovu@Nelianjovu ~]$
```

Рис. 3.3: Создание файла

```
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>
int
main ()
{
    uid_t uid = getuid ();
    gid_t gid = getegid ();
    printf ("uid=%d, gid=%d\n", uid, gid);
    return 0;
}
```

Рис. 3.4: Содержимое файла

Я скомпилировала файл, проверила что он скомпилирован(рис.5)

```
[Nelianjovu@Nelianjovu ~]$ gcc simplified.c -o simplified
[Nelianjovu@Nelianjovu ~]$ ls
Desktop  Downloads  Pictures  simplified  Templates
Documents Music      Public    simplified.c Videos
[Nelianjovu@Nelianjovu ~]$
```

Рис. 3.5: Компиляция файла

Я запустила исполняемый файл. Номера пользователя и группы указаны в выходных данных файла, они отличаются от выходных данных команды `id` тем, что в выходных данных для исполняемого файла отображается меньше информации, чем в команде `id`(рис.6)


```
[Nelianjovu@Nelianjovu ~]$ ./simplified
uid=1000, gid=1000
[Nelianjovu@Nelianjovu ~]$ id
uid=1000(Nelianjovu) gid=1000(Nelianjovu) groups=1000(Nelianjovu),10(wheel) cont
ext=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[Nelianjovu@Nelianjovu ~]$
```

Рис. 3.6: Сравнение команд

Я создала, записывала в файл в simplified2.c(рис.7 и рис.8)

```
[Nelianjovu@Nelianjovu ~]$ touch simplified2.c
[Nelianjovu@Nelianjovu ~]$ nano simplified2.c
[Nelianjovu@Nelianjovu ~]$
```

Рис. 3.7: Создание файла

```
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>
int
main ()
{
    uid_t real_uid = getuid ();
    uid_t e_uid = getuid ();
    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. 3.8: Содержимое файла

Я компилировала файл и запустила программы(рис.9)

```
[Nelianjovu@Nelianjovu ~]$ gcc simplified2.c -o simplified2
[Nelianjovu@Nelianjovu ~]$ ./simplified2
e_uid=1000, e_gid=1000
real_uid=1000, real_gid=1000
[Nelianjovu@Nelianjovu ~]$
```

Рис. 3.9: Компиляция файла

Я использовала `chown`, чтобы сменить владельца файла на суперпользователя, и `chmod`, чтобы изменить права доступа(рис.10)

```
[guest@Nelianjovu Nelianjovu]$ sudo chown root:Nelianjovu /home/Nelianjovu/simplified2
[sudo] password for guest:
[guest@Nelianjovu Nelianjovu]$ sudo chmod u+s /home/Nelianjovu/simplified2
[guest@Nelianjovu Nelianjovu]$ sudo ls -l /home/Nelianjovu/simplified2
-rwsr-xr-x. 1 root Nelianjovu 26008 Apr 18 12:03 /home/Nelianjovu/simplified2
[guest@Nelianjovu Nelianjovu]$
```

Рис. 3.10: Смена владельца файла и прав доступа к файлу

Я сравнила выходные данные программы и команды `id`, в очередной раз получила больше информации, используя команды `id`, чем наша программа(рис.11)

```
[guest@Nelianjovu Nelianjovu]$ sudo /home/Nelianjovu/simplified2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[guest@Nelianjovu Nelianjovu]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@Nelianjovu Nelianjovu]$ sudo id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@Nelianjovu Nelianjovu]$
```

Рис. 3.11: Запуск файла

Я создала, записывала в файл в `readfile.c`(рис.12 и рис.13)

```
[Nelianjovu@Nelianjovu ~]$ touch readfile.c
[Nelianjovu@Nelianjovu ~]$ nano readfile.c
[Nelianjovu@Nelianjovu ~]$
```

Рис. 3.12: Создание файла

```

#include <fcntl.h>
#include <stdio.h>
#include <sys/stat.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. 3.13: Содержимое файла

Я компилировала файл(рис.14)

```

[Nelianjovu@Nelianjovu ~]$ gcc readfile.c -o readfile
[Nelianjovu@Nelianjovu ~]$ ls
Desktop  Downloads  Pictures  readfile  simplified  simplified2.c  Templates
Documents Music      Public    readfile.c  simplified2  simplified.c   Videos
[Nelianjovu@Nelianjovu ~]$

```

Рис. 3.14: Компиляция файла

И снова, от имени пользователя guest, я меняла владельца файла для чтения. Затем я меняла права доступа, чтобы пользователи Nelianjovu не мог прочитать содержимое файла(рис.15)

```

[guest@Nelianjovu~Nelianjovu]$ sudo chown root:Nelianjovu /home/Nelianjovu/readfile.c
[sudo] password for guest:
[guest@Nelianjovu Nelianjovu]$ sudo chmod u+s /home/guest/readfile.c
chmod: cannot access '/home/guest/readfile.c': No such file or directory
[guest@Nelianjovu Nelianjovu]$ sudo chmod u+s /home/Nelianjovu/readfile.c
[guest@Nelianjovu Nelianjovu]$ sudo chmod 700 /home/Nelianjovu/readfile.c
[guest@Nelianjovu Nelianjovu]$ sudo chmod -r /home/Nelianjovu/readfile.c
[guest@Nelianjovu Nelianjovu]$ sudo chmod u+s /home/Nelianjovu/readfile.c
[guest@Nelianjovu Nelianjovu]$

```

Рис. 3.15: Смена владельца файла и прав доступа к файлу

Я попыталась прочитать содержимое файла readfile.c у пользователя Nelianjovu. Я не могу прочитать файл(рис.16)

```
[Nelianjovu@Nelianjovu ~]$ cat readfile.c
cat: readfile.c: Permission denied
[Nelianjovu@Nelianjovu ~]$
```

Рис. 3.16: Попытка прочесть содержимое файла

При попытке прочитать тот же файл с помощью программы readfile выдается сообщение “размытая информация”(рис.17)

```
[Nelianjovu@Nelianjovu ~]$ ./readfile readfile.c
86_64./readfilereadfile.cSHELL=/bin/bashSESSION_
MANAGER=local/unix:@/tmp/.ICE-unix/1683,unix/unix:/tmp/.ICE-unix/1683COLORTERM=t
ruecolorHISTCONTROL=ignoredupsXDG_MENU_PREFIX=gnome-HISTSIZE=1000HOSTNAME=Nelian
```

Рис. 3.17: Попытка прочесть содержимое файла программой

При попытке прочитать содержимое файла shadow с помощью программы readfile выдается тоже сообщение “размытая информация”(рис.18)

```
[Nelianjovu@Nelianjovu ~]$ ./readfile /etc/shadow
PWN
```

Рис. 3.18: Попытка прочесть содержимое файла программой

При попытке прочитать содержимое файла shadow с помощью программы readfile от имени пользователя guest получилось(рис.19)

```
[guest@Nelianjovu Nelianjovu]$ sudo /home/Nelianjovu/readfile /etc/shadow
root:$6$gmhwqmJcf1h16Jsy$ZJK0kQSMGANY7PT8F5YZ5Fn1qEFes2qWONLQ8x5/.0tRheklhgL5Txp
JKyWwC96mFwP7yuRT0JFeHs0lfpsjx/::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. 3.19: Чтение файла от имени пользователя guest

2. Исследование Sticky-бита

Я проверила папку tmp на наличие атрибута Sticky, потому что в выходных данных есть буква t, значит, атрибут установлен(рис.20)

```
[guest@Nelianjovu Nelianjovu]$ ls -l / | grep tmp
drwxrwxrwt. 15 root root 4096 Apr 18 12:33 tmp
```

Рис. 3.20: Проверка атрибутов директории tmp

От имени пользователя Nelianjovu создала файл с текстом, добавляла права на чтение и запись для других пользователей(рис.21)

```
[Nelianjovu@Nelianjovu ~]$ echo "test" > /tmp/file01.txt
[Nelianjovu@Nelianjovu ~]$ ls -l /tmp/file01.txt
-rw-r--r--. 1 Nelianjovu Nelianjovu 5 Apr 18 12:36 /tmp/file01.txt
[Nelianjovu@Nelianjovu ~]$ chmod o+r /tmp/file01.txt
[Nelianjovu@Nelianjovu ~]$ ls -l /tmp/file01.txt
-rw-r--rw-. 1 Nelianjovu Nelianjovu 5 Apr 18 12:36 /tmp/file01.txt
[Nelianjovu@Nelianjovu ~]$
```

Рис. 3.21: Создание файла, изменение прав доступа

Вхожу в систему от имени пользователя guest2, от его имени перезаписать информацию в файле file 01.txt не могу(рис.22)

```
[guest2@Nelianjovu Nelianjovu]$ echo "test2" >> /tmp/file01.txt
bash: /tmp/file01.txt: Permission denied
[guest2@Nelianjovu Nelianjovu]$
```

Рис. 3.22: Попытка запись файла

От имени суперпользователя снимала с директории атрибут Sticky(рис.23)

```
[guest2@Nelianjovu Nelianjovu]$ su -
Password:
[root@Nelianjovu ~]# chmod -t /tmp
[root@Nelianjovu ~]# exit
logout
[guest2@Nelianjovu Nelianjovu]$
```

Рис. 3.23: Смена атрибутов файла

Я проверила, что атрибут действительно снят(рис.24)

```
[guest2@Nelianjovu Nelianjovu]$ ls -l / | grep tmp
drwxrwxrwx. 15 root root 4096 Apr 18 13:01 tmp
[guest2@Nelianjovu Nelianjovu]$
```

Рис. 3.24: Проверка атрибутов директории

Далее я повторила предыдущие действия. Согласно результатам, запись в файл и повторная запись в файл оставались невозможными без Sticky-бита(рис.25)

```
[guest2@Nelianjovu Nelianjovu]$ ls -l / | grep tmp
drwxrwxrwx. 15 root root 4096 Apr 18 13:01 tmp
[guest2@Nelianjovu Nelianjovu]$
```

Рис. 3.25: Повтор предыдущих действий

Затем я вернула каталог tmp с атрибутом t от имени суперпользователя(рис.26)

```
[guest2@Nelianjovu Nelianjovu]$ ls -l / | grep tmp
drwxrwxrwx. 15 root root 4096 Apr 18 13:01 tmp
[guest2@Nelianjovu Nelianjovu]$
```

Рис. 3.26: Изменение атрибутов

4 Выводы

Выполнив эту работу, я изучила механизмы изменения идентификаторов, применения SetUID- и Sticky-битов и 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.