

# **Лабораторная работа №5**

**Основы информационной безопасности**

Серёгина Ирина Андреевна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>3</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>4</b>
<b>3</b>	<b>Выводы</b>	<b>10</b>
	<b>Список литературы</b>	<b>11</b>

# 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 Выполнение лабораторной работы

1. Проверяю, установлен ли у меня gcc (рис. 2.1).

```
iseregina@localhost ~]$ sudo -i
[sudo] пароль для iseregina:
[root@localhost ~]# yum install gcc
Последняя проверка окончания срока действия метаданных: 0:54:48 назад, Чт 28 мар 2024 19:13:30.
Зависимости разрешены.
=====
Пакет                Архитектура  Версия                Репозиторий  Размер
=====
Установка:
gcc                  x86_64       11.4.1-2.1.el9       appstream    32 M
Обновление:
glibc                x86_64       2.34-83.el9.12       baseos       1.9 M
glibc-all-langpacks x86_64       2.34-83.el9.12       baseos       18 M
glibc-common         x86_64       2.34-83.el9.12       baseos       303 k
glibc-gconv-extra    x86_64       2.34-83.el9.12       baseos       1.5 M
glibc-langpack-ru    x86_64       2.34-83.el9.12       baseos       532 k
Установка зависимостей:
glibc-devel          x86_64       2.34-83.el9.12       appstream    43 k
glibc-headers        x86_64       2.34-83.el9.12       appstream    444 k
kernel-headers       x86_64       5.14.0-362.24.1.el9_3 appstream    6.1 M
libxcrypt-devel      x86_64       4.4.18-3.el9         appstream    28 k
```

Рис. 2.1: gcc установлен

2. Отключаю систему запретов до следующей перезагрузки системы (рис. 2.2).

```
[root@localhost ~]# setenforce 0
[root@localhost ~]# getenforce
Permissive
[root@localhost ~]#
Выход
```

Рис. 2.2: отключаю систему запретов

3. Вхожу от имени пользователя guest (рис. 2.3).

```
[guest@localhost iseregina]$ su - guest
Пароль:
[guest@localhost ~]$ pwd
```

Рис. 2.3: пользователь guest

4. Создаю программу simpleid.c (рис. 2.4).

```
[guest@localhost ~]$ touch simpleid.c
[guest@localhost ~]$ ls
[git] Видео Загрузки Музыка 'Рабочий стол'
simpleid.c Документы Изображения Общедоступные Шаблоны
[guest@localhost ~]$ nano simpleid.c
```

Рис. 2.4: создание программы

5. Заполняю программу (рис. 2.5).

```
GNU nano 5.6.1 simpleid.c Изменён
#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. 2.5: simpleid.c

6. Компилирую файл и выполняю системную программу id (рис. 2.6).

```
[guest@localhost ~]$ gcc simpleid.c -o simpleid
[guest@localhost ~]$ ./simpleid
uid=1001, gid=1001
[guest@localhost ~]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Рис. 2.6: запускаю программу

7. Усложняю программу (рис. 2.7).

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

Имя файла для записи: simpleid2.c

Рис. 2.7: измененная программа

8. Компилирую и запускаю программу (рис. 2.8).

```
[guest@localhost ~]$ gcc simpleid2.c -o simpleid2
[guest@localhost ~]$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
```

Рис. 2.8: запускаю программу

9. Выполняю следующие команды от лица суперпользователя, после чего выполняю проверку правильности запуска (рис. 2.9).

```
[guest@localhost ~]$ su root
Пароль:
[root@localhost guest]# chown root:guest /home/guest/simpleid2
[root@localhost guest]# chmod u+s /home/guest/simpleid2
[root@localhost guest]#
exit
```

Рис. 2.9: выполняю команды от имени суперпользователя

10. Запускаю simpleid и id (рис. 2.10).

```
[guest@localhost ~]$ ls -l simpleid2
-rwsr-xr-x. 1 root guest 24488 мар 28 20:22 simpleid2
[guest@localhost ~]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfi
ned_r:unconfined_t:s0-s0:c0.c1023
```

Рис. 2.10: simpleid.c и id

11. Создаю программу readfile.c, компилирую её (рис. 2.11).

```
GNU nano 5.6.1 readfile.c Изменён
#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;
    }
}
```

Имя файла для записи: readfile.c

Рис. 2.11: readfile.c

12. Меняю владельца файла, чтобы прочитать можно было только от лица суперпользователя (рис. 2.12).

```
[guest@localhost ~]$ su root
Пароль:
[root@localhost guest]# chown root:guest /home/guest/readfile.c
[root@localhost guest]# chmod o-r /home/guest/readfile.c
[root@localhost guest]#
exit
```

Рис. 2.12: меняю владельца файла

13. Readfile может прочитать файл /etc/shadow (рис. 2.13).





17. Файл удалился (рис. 2.16).

```
[guest3@localhost ~]$ rm /tmp/file01.txt
[guest3@localhost ~]$ ls /tmp
systemd-private-c9eb40305adf4a2f9f829f5477fca7eb-chronyd.service-kcouFT
systemd-private-c9eb40305adf4a2f9f829f5477fca7eb-colord.service-lvOpJI
systemd-private-c9eb40305adf4a2f9f829f5477fca7eb-dbus-broker.service-PkFr8N
systemd-private-c9eb40305adf4a2f9f829f5477fca7eb-fwupd.service-GQEa3L
systemd-private-c9eb40305adf4a2f9f829f5477fca7eb-ModemManager.service-2lFdyM
systemd-private-c9eb40305adf4a2f9f829f5477fca7eb-power-profiles-daemon.service-
uiBS3
systemd-private-c9eb40305adf4a2f9f829f5477fca7eb-rtkit-daemon.service-GbAlyS
systemd-private-c9eb40305adf4a2f9f829f5477fca7eb-switcheroo-control.service-eZX
5L
systemd-private-c9eb40305adf4a2f9f829f5477fca7eb-systemd-logind.service-WnzJW4
systemd-private-c9eb40305adf4a2f9f829f5477fca7eb-upower.service-ITwXpr
```

Рис. 2.16: файл удален

18. Возвращаю атрибут t (рис. 2.17).

```
пароль.
[root@localhost ~]# chmod +t /tmp
[root@localhost ~]#
выход
```

Рис. 2.17: возвращаю атрибут

## 3 Выводы

Я изучила механизмы изменения идентификаторов, применения SetUID- и Sticky-битов. 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.

## **Список литературы**