

Лабораторная работа №5

Основы информационной безопасности

Серёгина Ирина Андреевна

13 апреля 2024

Российский университет дружбы народов, 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.

Выполнение лабораторной работы

1. Проверяю, установлен ли у меня gcc (рис. 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 М
Обновление:
glibc                x86_64                2.34-83.el9.12    baseos       1.9 М
glibc-all-langpacks x86_64                2.34-83.el9.12    baseos       18 М
glibc-common         x86_64                2.34-83.el9.12    baseos       303 к
glibc-gconv-extra    x86_64                2.34-83.el9.12    baseos       1.5 М
glibc-langpack-ru    x86_64                2.34-83.el9.12    baseos       532 к
/установка зависимостей:
glibc-devel          x86_64                2.34-83.el9.12    appstream    43 к
glibc-headers        x86_64                2.34-83.el9.12    appstream    444 к
kernel-headers       x86_64                5.14.0-362.24.1.el9_3 appstream    6.1 М
libxcrypt-devel      x86_64                4.4.18-3.el9      appstream    28 к
```

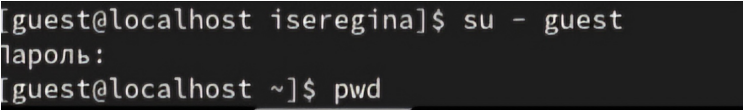
Рис. 1: gcc установлен

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

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

Рис. 2: отключаю систему запретов

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



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

Рис. 3: пользователь guest

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

```
[guest@localhost ~]$ touch simpleid.c
[guest@localhost ~]$ ls
dir1          Видео          Загрузки      Музыка        'Рабочий стол'
simpleid.c     Документы      Изображения   Общедоступные  Шаблоны
[guest@localhost ~]$ nano simpleid.c
```

Рис. 4: создание программы

5. Заполняю программу (рис. 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;
}
```

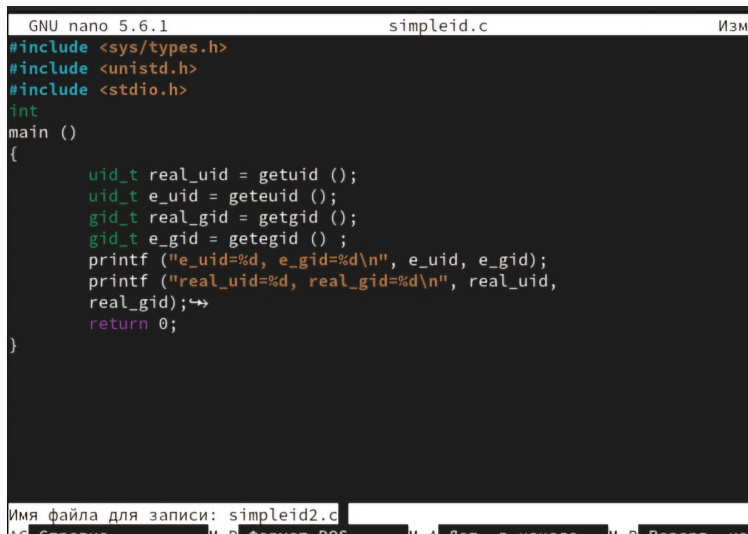
Рис. 5: simpleid.c

6. Компилирую файл и выполняю системную программу id (рис. 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. 6: запускаю программу

7. Усложняю программу (рис. 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. 7: измененная программа

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

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

Рис. 8: запускаю программу

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

10. Запускаю simpleid и id (рис. 10).

```
[guest@localhost ~]$ ls -l simpleid2
-rwsr-xr-x. 1 root guest 24488 map 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. 10: simpleid.c и id

11. Создаю программу readfile.c, компилирую её (рис. 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. 11: readfile.c

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

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

```
[guest@localhost ~]$ ./readfile /etc/shadow
XXXXXXXXXXXXXXXXXXXXFV@XXXXXXXXXXXX>XXXXXXXXXXXXhXXXXXXXXXXXXFV@XXXXp@
XXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXDXXXXXXXXXXXXDXXXXXXXXXXXX
XXXXXXXXXXXX@DXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXX@X8

XXXXXXXXXXXXX XXXXXXXX XXXXXXXX Sx86_64./readfile/etc/shadowSHELL=/bin/bashHISTCONTR
OL=1ignoredupsHISTSIZ=1000HOSTNAME=localhostPWD=/home/guestLOGNAME=guestXAUTHORI
TY=/home/guest/.xauthXSZYLBSHOME=/home/guestLANG=ru_RU.UTF-8LS_COLORS=rs=0:di=01;
34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33;01:cd=40;33;01:or=40;31;01
:mi=01;37;41:su=37;41:sg=30;43:ca=30;41:tw=30;42:ow=34;42:st=37;44:ex=01;32:*.ta
r=01;31:*.tgz=01;31:*.arc=01;31:*.arj=01;31:*.taz=01;31:*.lha=01;31:*.lz4=01;31:
*.lzh=01;31:*.lzma=01;31:*.tlz=01;31:*.txz=01;31:*.tzo=01;31:*.t7z=01;31:*.zip=0
1;31:*.z=01;31:*.dz=01;31:*.gz=01;31:*.lrz=01;31:*.lz=01;31:*.lzo=01;31:*.xz=01;
31:*.zst=01;31:*.tztst=01;31:*.bz2=01;31:*.bz=01;31:*.tbz=01;31:*.tbz2=01;31:*.tz
=01;31:*.deb=01;31:*.rpm=01;31:*.jar=01;31:*.war=01;31:*.ear=01;31:*.sar=01;31:*.
rar=01;31:*.alz=01;31:*.ace=01;31:*.zoo=01;31:*.cpio=01;31:*.7z=01;31:*.rz=01;3
1:*.cab=01;31:*.wim=01;31:*.swm=01;31:*.dwm=01;31:*.esd=01;31:*.jpg=01;35:*.jpeg
=01;35:*.mjpg=01;35:*.mjpeg=01;35:*.gif=01;35:*.bmp=01;35:*.pbm=01;35:*.pgm=01;3
5:*.ppm=01;35:*.tga=01;35:*.xbm=01;35:*.xpm=01;35:*.tif=01;35:*.tiff=01;35:*.png
=01;35:*.svg=01;35:*.svgz=01;35:*.mng=01;35:*.pcx=01;35:*.mov=01;35:*.mpg=01;35:
*.mpeg=01;35:*.m2v=01;35:*.mkv=01;35:*.webm=01;35:*.webp=01;35:*.ogm=01;35:*.mp4
=01;35:*.m4v=01;35:*.mp4v=01;35:*.vob=01;35:*.qt=01;35:*.nuv=01;35:*.wmv=01;35:*.
asf=01;35:*.rm=01;35:*.rmvb=01;35:*.flc=01;35:*.avi=01;35:*.fli=01;35:*.flv=01;
```

Рис. 13: /etc/shadow

14. Выясняю, установлен ли атрибут sticky на /tmp, создаю файл file01.txt, просматриваю его атрибуты (рис. 14).

```
[iseregina@localhost ~]$ su - guest
Пароль:
[guest@localhost ~]$ echo "test" > /tmp/file01.txt
[guest@localhost ~]$ ls -l /tmp/file01.txt
-rw-r--r--. 1 guest guest 5 map 28 20:34 /tmp/file01.txt
[guest@localhost ~]$ chmod o+rw /tmp/file01.txt
[guest@localhost ~]$ ls -l /tmp/file01.txt
-rw-r--rw-. 1 guest guest 5 map 28 20:34 /tmp/file01.txt
[guest@localhost ~]$
```

Рис. 14: проверяю наличие атрибута

15. От имени пользователя, не являющегося владельцем пробую прочитать и изменить файл. Не могу также удалить его (рис. 15).

пробую выполнить действия с файлом

16. От имени суперпользователя убираю атрибут `t` с директории (рис. 16).

```
пароль.  
[root@localhost ~]# chmod -t /tmp  
[root@localhost ~]#  
ВЫХОД
```

Рис. 15: убираю атрибут с директории

17. Файл удален (рис. 17).

```
[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. 16: файл удален

18. Возвращаю атрибут t (рис. 18).

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

Рис. 17: возвращаю атрибут

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.

Спасибо за внимание!