

Отчёт по лабораторной работе №5

Дисциплина: Основы информационной безопасности

Зинченко Анастасия Романовна

Содержание

Цель работы	1
Выполнение лабораторной работы	1
Подготовка лабораторного стенда	1
Создание программы	2
Исследование Sticky-бита	8
Выводы	10
Список литературы	10

Цель работы

Целью данной работы является 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

Подготовка лабораторного стенда

Для лабораторной работы необходимо проверить, установлен ли компилятор gcc. Проверим это с помощью команды *gcc -v*. Также осуществим отключение системы запретов с помощью *sudo setenforce 0* и проверим командой *getenforce*, что выводится Permissive (рис. [-@fig:001])

```
[arzinchenko@arzinchenko ~]$ 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-languages=c,c++,fortran,lto --prefix=/usr --mandir=/usr/share
/man --infodir=/usr/share/info --with-bugurl=https://bugs.rockylinux.org/ --enab
le-shared --enable-threads=posix --enable-checking=release --with-system-zlib --
enable-__cxa_atexit --disable-libunwind-exceptions --enable-gnu-unique-object --
enable-linker-build-id --with-gcc-major-version-only --enable-plugin --enable-in
itfini-array --without-isl --enable-multilib --with-linker-hash-style=gnu --enab
le-offload-targets=nvptx-none --without-cuda-driver --enable-gnu-indirect-functi
on --enable-cet --with-tune=generic --with-arch_64=x86-64-v2 --with-arch_32=x86-
64 --build=x86_64-redhat-linux --with-build-config=bootstrap-lto --enable-link-s
erialization=1
Thread model: posix
Supported LTO compression algorithms: zlib zstd
gcc version 11.5.0 20240719 (Red Hat 11.5.0-2) (GCC)
```

Подготовка к лабораторной работе

Создание программы

Войдём в систему от имени пользователя guest. И создадим программу simpleid.c (рис. [-@fig:002]), (рис. [-@fig:003])

```
[arzinchenko@arzinchenko ~]$ su guest
Password:
[guest@arzinchenko arzinchenko]$ touch simpled.c
touch: cannot touch 'simpled.c': Permission denied
[guest@arzinchenko arzinchenko]$ cd
[guest@arzinchenko ~]$ touch simpled.c
```

Создание и открытие файла simpleid.c

Листинг программы simpleid.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;
}
```

```

GNU nano 5.6.1                               sim
#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;
}

```

Редактирование файла simpleid.c

Скомпилируем программу и убедимся, что файл программы создан: `gcc simpleid.c -o simpleid` (рис. [-@fig:004])

```

[guest@arzinchenko ~]$ gcc simpled.c -o simpled
[guest@arzinchenko ~]$ ls
Desktop  Documents  Music      Public    simpled.c  Videos
iri      Downloads  Pictures   simpled   Templates
[guest@arzinchenko ~]$

```

Компиляция программы simpleid.c

Выполним программу simpleid: `./simpleid`. В выводе файла выписаны номера пользователя и групп, от вывода при вводе `id`, они отличаются только тем, что информации меньше (рис. [-@fig:005])

```

[guest@arzinchenko ~]$ ./simpled
uid=1001, gid=1001

```

Выполнение программы simpleid

Выполните системную программу `id`: `id` (рис. [-@fig:006])

```

[guest@arzinchenko ~]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023

```

Выполнение системной программы id

Далее усложним программу, добавив вывод действительных идентификаторов. Создадим программу под названием `simpleid2.c` (рис. [-@fig:007])

Листинг программы `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;
}

```



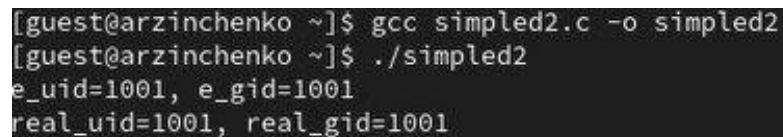
```

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;
}

```

Редактирование файла simpleid2.c

Скомпилируем и запустим simpleid2.c: `gcc simpleid2.c -o simpleid2` и `./simpleid2` (рис. [-@fig:008]), (рис. [-@fig:009])

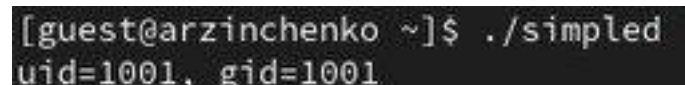


```

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

```

Компиляция программы simpleid2.c



```

[guest@arzinchenko ~]$ ./simpleid2
uid=1001, gid=1001

```

Выполнение программы simpleid2

От имени суперпользователя выполним команды: `chown root:guest /home/guest/simpleid2` и `chmod u+s /home/guest/simpleid2`. Также выполним проверку правильности установки новых атрибутов и смены владельца файла simpleid2: `ls -l simpleid2`. С помощью `chown` мы меняем владельца файла на суперпользователя, а с помощью `chmod` меняем права доступа (рис. [-@fig:010])

```
[arzinchenko@arzinchenko ~]$ sudo chown root:guest /home/guest/simplied2
[arzinchenko@arzinchenko ~]$ sudo chmod u+s /home/guest/simplied2
[arzinchenko@arzinchenko ~]$ sudo ls -l /home/guest/simplied2
-rwsr-xr-x. 1 root guest 17656 Apr 19 17:37 /home/guest/simplied2
```

Смена владельца файла и прав доступа к файлу `simpleid2`

Запустим `simpleid2` и `id`: `./simpleid2` и `id`. Наша команда снова вывела только ограниченное количество информации (рис. [-@fig:011])

```
[arzinchenko@arzinchenko ~]$ sudo /home/guest/simplied2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[arzinchenko@arzinchenko ~]$ id
uid=1000(arzinchenko) gid=1000(arzinchenko) groups=1000(arzinchenko),10(wheel) c
ontext=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[arzinchenko@arzinchenko ~]$ sudo id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfi
ned_t:s0-s0:c0.c1023
```

Запуск `simpleid2` и `id`

Создадим программу `readfile.c` (рис. [-@fig:012]), (рис. [-@fig:013])

```
[guest@arzinchenko ~]$ touch readfile.c
[guest@arzinchenko ~]$ nano readfile.c
```

Создание и открытие файла `readfile.c`

Листинг программы `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;
}
```

```

#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;
}

```

Редактирование файла readfile.c

Откомпилируем её: `gcc readfile.c -o readfile` (рис. [-@fig:014])

```

[guest@arzinchenko ~]$ gcc readfile.c -o readfile
[guest@arzinchenko ~]$ ls
Desktop  Downloads  Public      simplified  simplified.c
dir1     Music      readfile    simplified2  Templates
Documents Pictures  readfile.c  simplified2.c Videos

```

Компиляция программы readfile.c

Далее сменим владельца у файла readfile.c и изменим права так, чтобы только суперпользователь (root) мог прочитать его, а guest не мог (рис. [-@fig:015])

```

[arzinchenko@arzinchenko ~]$ sudo chmod u+s /home/guest/readfile.c
[arzinchenko@arzinchenko ~]$ sudo chmod 700 /home/guest/readfile.c
[arzinchenko@arzinchenko ~]$ sudo chmod -r /home/guest/readfile.c
[arzinchenko@arzinchenko ~]$ sudo chmod u+s /home/guest/readfile.c

```

Смена владельца файла и прав доступа к файлу readfile

Проверим, может ли программа readfile прочитать файл readfile.c (рис. [-@fig:016])

```

35:*.mjpeg=01;35:*.gif=01;35:*.bmp=01;35:*.pbm=01;35:*.pgm=01;35:*.ppm=01;35:*.t
ga=01;35:*.xbm=01;35:*.xpm=01;35:*.tif=01;35:*.tiff=01;35:*.png=01;35:*.svg=01;3
5:*.svgz=01;35:*.mng=01;35:*.pcx=01;35:*.mov=01;35:*.mpg=01;35:*.mpeg=01;35:*.m2
v=01;35:*.mkv=01;35:*.webm=01;35:*.webp=01;35:*.ogm=01;35:*.mp4=01;35:*.m4v=01;3
5:*.mp4v=01;35:*.vob=01;35:*.qt=01;35:*.nuv=01;35:*.wmv=01;35:*.asf=01;35:*.rm=0
1;35:*.rmvb=01;35:*.flc=01;35:*.avi=01;35:*.fli=01;35:*.flv=01;35:*.gl=01;35:*.d
l=01;35:*.xcf=01;35:*.xwd=01;35:*.yuv=01;35:*.cgm=01;35:*.emf=01;35:*.ogv=01;35:
*.ogx=01;35:*.aac=01;36:*.au=01;36:*.flac=01;36:*.m4a=01;36:*.mid=01;36:*.midi=0
1;36:*.mka=01;36:*.mp3=01;36:*.mpc=01;36:*.ogg=01;36:*.ra=01;36:*.wav=01;36:*.og
a=01;36:*.opus=01;36:*.spx=01;36:*.xspf=01;36:XdG_CURRENT_DESKTOP=GNOMEVTE_VERSI
ON=6402WAYLAND_DISPLAY=wayland-0GNOME_TERMINAL_SCREEN=/org/gnome/terminal/screen
/5525d985_0d78_4db4_aeea_85635c6c2348GNOME_SETUP_DISPLAY=:XdG_SESSION_CLASS=use
rTERM=xterm-256colorLESSOPEN=||/usr/bin/lesspipe.sh %sUSER=guestGNOME_TERMINAL_S
ERVICE=:1.79DISPLAY=:0SHLVL=2QT_IM_MODULE=ibusXdG_RUNTIME_DIR=/run/user/1000whic
h_declare=declare -fXdG_DATA_DIRS=/home/guest/.local/share/flatpak/exports/share
:/home/arzinchenko/.local/share/flatpak/exports/share:/var/lib/flatpak/exports/s
hare:/usr/local/share:/usr/share/PATH=/home/guest/.local/bin:/home/guest/bin:/h
ome/arzinchenko/.local/bin:/home/arzinchenko/bin:/usr/local/bin:/usr/local/sbin:
/usr/bin:/usr/sbinGDMSESSION=gnomeDBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1
000/buMAIL=/var/spool/mail/arzinchenkoBASH_FUNC_which%%=( ) { ( alias;
eval ${which_declare} ) | /usr/bin/which --tty-only --read-alias --read-functio
ns --show-tilde --show-dot $@
Segmentation fault (core dumped)

```

Попытка прочесть содержимое файла readfile.c программой readfile

Теперь попробуем прочесть эти же файлы от имени суперпользователя (рис. [-@fig:018])


```
sync:*:19820:0:99999:7:::
shutdown:*:19820:0:99999:7:::
halt:*:19820:0:99999:7:::
mail:*:19820:0:99999:7:::
operator:*:19820:0:99999:7:::
games:*:19820:0:99999:7:::
ftp:*:19820:0:99999:7:::
nobody:*:19820:0:99999:7:::
tss:!!:20138:::
systemd-coredump:!!:20138:::
dbus:!!:20138:::
polkitd:!!:20138:::
sssd:!!:20138:::
avahi:!!:20138:::
geoclue:!!:20138:::
rtkit:!!:20138:::
pipewire:!!:20138:::
libstoragemgmt:*:20138:::
cockpit-wsinstance:!!:20138:::
flatpak:!!:20138:::
colord:!!:20138:::
clevis:!!:20138:::
setroubleshoot:!!:20138:::
gdm:!!:20138:::
```

Попытка прочесть содержимое файлов от имени суперпользователя

Исследование Sticky-бита

Выясним, установлен ли атрибут Sticky на директории /tmp, для чего выполним команду `ls -l | grep tmp`. Так как в выводе есть буква t, это значит что атрибут установлен (рис. [-@fig:019])

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

Проверка атрибутов директории tmp

От имени пользователя guest создадим файл file01.txt в директории /tmp со словом test: `echo "test" > /tmp/file01.txt` (рис. [-@fig:020])


```
[guest@arzinchenko ~]$ ls -l /tmp/file01.txt
-rw-r--r--. 1 guest guest 5 Apr 19 18:06 /tmp/file01.txt
[guest@arzinchenko ~]$ chmod o+rw /tmp/file01.txt
[guest@arzinchenko ~]$ ls -l /tmp/file01.txt
-rw-r--rw-. 1 guest guest 5 Apr 19 18:06 /tmp/file01.txt
```

Создание файла file01.txt со словом test

Просмотрим атрибуты у только что созданного файла `ls -l /tmp/file01.txt` и разрешим чтение и запись для категории пользователей «все остальные»: `chmod o+rw /tmp/file01.txt` (рис. [-@fig:021])

```
[arzinchenko@arzinchenko ~]$ su guest2
Password:
[guest2@arzinchenko arzinchenko]$ cat /tmp/file01.txt
test
```

Смена атрибутов файла file01.txt

От пользователя guest2 (не являющегося владельцем) попробуем прочитать файл `/tmp/file01.txt`: `cat /tmp/file01.txt` (рис. [-@fig:022])

```
[guest2@arzinchenko arzinchenko]$ echo 'text2' >> /tmp/file01.txt
bash: /tmp/file01.txt: Permission denied
[guest2@arzinchenko arzinchenko]$ cat /tmp/file01.txt
test
```

Попытка чтения файла

От пользователя guest2 попробуем дозаписать в файл `/tmp/file01.txt` слово test2 командой `echo "test2" > /tmp/file01.txt` (рис. [-@fig:023])

```
[guest2@arzinchenko arzinchenko]$ echo 'text3' > /tmp/file01.txt
bash: /tmp/file01.txt: Permission denied
[guest2@arzinchenko arzinchenko]$ cat /tmp/file01.txt
test
```

Попытка дозаписи в файл

От пользователя guest2 попробуем записать в файл `/tmp/file01.txt` слово test3, стерев при этом всю имеющуюся в файле информацию командой `echo "test3" > /tmp/file01.txt` (рис. [-@fig:024])

```
[guest2@arzinchenko arzinchenko]$ rm /tmp/file01.txt
rm: remove write-protected regular file '/tmp/file01.txt'?
```

Попытка записи в файл и удаление всей имеющейся в нём информации

От пользователя guest2 попробуем удалить файл `/tmp/file01.txt` командой `rm /tmp/file01.txt` (рис. [-@fig:025])

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

Попытка удалить файл

Далее от имени суперпользователя снимем с директории tmp атрибут Sticky командой `chmod -t /tmp` (рис. [-@fig:026]), (рис. [-@fig:027])

```
[guest2@arzinchenko arzinchenko]$ ls -s / | grep tmp  
4 tmp
```

Снятие с директории tmp атрибута Sticky

```
[root@arzinchenko ~]# chmod +t /tmp  
[root@arzinchenko ~]# exit  
logout
```

Проверка того что атрибут снялся

Выводы

В результате выполнения работы мы изучили механизмы изменения идентификаторов, применения SetUID- и Sticky-битов. Получили практические навыки работы в консоли с дополнительными атрибутами. Рассмотрели работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов

Список литературы

1. Лабораторная работа №5 [Электронный ресурс] URL: https://esystem.rudn.ru/pluginfile.php/2580984/mod_resource/content/2/005-lab_discret_sticky.pdf
2. Инструментарий программиста в Linux: Компилятор GCC [Электронный ресурс] URL: <http://parallel.imm.uran.ru/freesoft/make/instrum.html>