Отчёт по лабораторной работе №5

Дисциплина: Информационная безопасность

Андрианова Марина Георгиевна

Содержание

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.

Выполнение лабораторной работы

Для лабораторной работы необходимо проверить, установлен ли компилятор gcc, делаем это с помощью команды `gcc -v`. Также осуществляется отключение системы запретом с помощью `setenforce 0` (рис. 1).

```
[mgandrianova@localhost ~]$ gcc -v
Используются внутренние спецификации.
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
Целевая архитектура: x86_64-redhat-linux
Параметры конфигурации: ../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/ --enable-shared --enable-threads=posix --enable-checking=release --with-sys
tem-zlib --enable-__cxa_atexit --disable-libunwind-exceptions --enable-gnu-unique-object --enable-linker-build-id --with-gcc-major-version-only --enable-plugi
n --enable-initfini-array --without-isl --enable-multilib --with-linker-hash-style=gnu --enable-offload-targets=nvptx-none --without-cuda-driver --enable-gnu-
indirect-function --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-lt
o --enable-link-serialization=1
Модель многопоточности: posix
Supported LTO compression algorithms: zlib zstd
gcc версия 11.4.1 20231218 (Red Hat 11.4.1-3) (GCC)
[mgandrianova@localhost ~]$ sudo setenforce 0
[sudo] пароль для mgandrianova:
[mgandrianova@localhost ~]$ getenforce
Permissive
[mgandrianova@localhost ~]$
```

Подготовка к лабораторной работе

Проверяем местоположение gcc и g++ (рис. 2).

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

Местоположение gcc и g++

Осуществляется вход от имени пользователя guest (рис. 3).

```
[mgandrianova@localhost ~]$ su guest
Пароль:
[guest@localhost mgandrianova]$ touch simpleid.c
```

Вход от имени пользователя guest

Создание файла simpled.c и запись в файл кода (рис. 4)

```
[guest@localhost ~]$ touch simpleid.c
[guest@localhost ~]$ nano simpleid.c
```

Создание файла simpled.c

C++ Листинг 1 #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; }

Содержимое файла выглядит следующим образом (рис. 5)

Содержимое файла

Компилирую файл, проверяю, что он скомпилировался (рис. 6)

```
[guest@localhost ~]$ gcc simpleid.c -o simpleid
[guest@localhost ~]$ ls
 simpleid     Видео       Загрузки      Музыка          'Рабочий стол'
 simpleid.c   Документы   Изображения   Общедоступные    Шаблоны
[guest@localhost ~]$
```

Компиляция файла

Запускаю исполняемый файл. В выводе файла выписаны номера пользователя и групп; от вывода при вводе if они отличаются только тем, что информации меньше (рис. 7)

```
[guest@localhost ~]$ ./simpleid
uid=1001, gid=1001
[guest@localhost ~]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest),10(wheel) контекст=unconfined
_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Сравнение команд

Создание файла simpled2.c и запись в файл кода (рис. 8-9)

```
[guest@localhost ~]$ touch simpleid2.c
[guest@localhost ~]$ nano simpleid2.c
```

Создание файла simpled2.c

```
  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;
}
```

Содержимое файла

```
C++ Листинг 2 #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; }
```

Компиляция файла simpled2.c. Запуск программы (рис. 10)

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

Компиляция файла simpled2.c.

С помощью chown изменяю владельца файла на суперпользователя, с помощью chmod изменяю права доступа (рис. 11)

```
[guest@localhost ~]$ sudo chown root:guest /home/guest/simpleid2
[sudo] пароль для guest:
[guest@localhost ~]$ sudo chmod u+s /home/guest/simpleid2
[guest@localhost ~]$ sudo ls -l /home/guest/simpleid2
-rwsr-xr-x. 1 root guest 17720 окт  4 20:52 /home/guest/simpleid2
```

Смена владельца файла и прав доступа к файлу

Создание файла readfile.c и запись кода в файл (рис. 12-13)

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

Создание и компиляция файла

Содержимое файла readfile.c

```
C++ Листинг 3 #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; }
```

Компиляция файла readfile.c. Запуск программы (рис. 14)

```
[guest@localhost ~]$ gcc readfile.c -o readfile
[guest@localhost ~]$ ls
 readfile     simpleid    simpleid2.c   Видео       Загрузки     Музыка          'Рабочий стол'
 readfile.c   simpleid2   simpleid.c    Документы   Изображения  Общедоступные   Шаблоны
```

Компиляция файла readfile.c

Снова от имени суперпользователи меняю владельца файла readfile. Далее меняю права доступа так, чтобы пользователь guest не смог прочесть содержимое файла (рис. 15)

```
[guest@localhost ~]$ sudo chown root:guest /home/guest/readfile.c
[guest@localhost ~]$ sudo chmod u+s /home/guest/readfile.c
[guest@localhost ~]$ sudo chmod 700 /home/guest/readfile.c
[guest@localhost ~]$ sudo chmod -r /home/guest/readfile.c
[guest@localhost ~]$ sudo chmod u+s /home/guest/readfile.c
[guest@localhost ~]$
```

Смена владельца файла и прав доступа к файлу

Проверка прочесть файл от имени пользователя guest. Прочесть файл не удается (рис. 16)

```
[guest@localhost ~]$ cat readfile.c
cat: readfile.c: Отказано в доступе
```

Попытка прочесть содержимое файла

Попытка прочесть тот же файл с помощью программы readfile, в ответ получаем непонятный текст (рис. 17)

```
[guest@localhost ~]$ ./readfile readfile.c
```

Попытка прочесть содержимое файла программой

Попытка прочесть файл `\etc\shadow` с помощью программы, все еще получаем непонятный текст (рис. 18)

```
[guest@localhost ~]$ ./readfile /etc/shadow
```

Попытка прочесть содержимое файла программой

Пробуем прочесть эти же файлы от имени суперпользователя и чтение файлов проходит успешно (рис. 19)

```
[guest@localhost ~]$ sudo /home/guest/readfile /etc/shadow
root:$6$CULeLnoUeBmXzEc.$uCzdaZzZdhgVq2g0pFCsMrNfZhL6cETw.gtPGDNpzlxhVH95HXCeeTDYlDZiRqQ8L.EqhGIwmClHGOf1dml2i.
::0:99999:7:::
bin:*:19820:0:99999:7:::
daemon:*:19820:0:99999:7:::
adm:*:19820:0:99999:7:::
lp:*:19820:0:99999:7:::
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:::
```

Чтение файла от имени суперпользователя

Проверяем папку tmp на наличие атрибута Sticky, т.к. в выводе есть буква t, то атрибут установлен (рис. 20)

```
[guest@localhost ~]$ ls -l / | grep tmp
drwxrwxrwt.  20 root root 4096 окт  4 21:05 tmp
```

Проверка атрибутов директории tmp

От имени пользователя guest создаю файл с текстом, добавляю права на чтение и запись для других пользователей (рис. 21)

```
[guest@localhost ~]$ echo "test" > /tmp/file01.txt
[guest@localhost ~]$ ls -l /tmp/file01.txt
-rw-r--r--. 1 guest guest 5 окт  4 21:06 /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 окт  4 21:06 /tmp/file01.txt
```

Создание файла, изменение прав доступа

Вхожу в систему от имени пользователя guest2, от его имени могу прочитать файл file01.txt (рис. 22), могу перезаписать в нём информацию (рис. 23)

```
[guest@localhost ~]$ su guest2
Пароль:
[guest2@localhost guest]$ cat /tmp/file01.txt
test
```

Попытка чтения файла

```
[guest2@localhost guest]$ echo "test2" >> /tmp/file01.txt
[guest2@localhost guest]$ cat /tmp/file01.txt
test
test2
[guest2@localhost guest]$ echo "test3" > /tmp/file01.txt
[guest2@localhost guest]$ cat /tmp/file01.txt
test3
```

Перезапись файла

Далее пробуем удалить файл, снова получаем отказ (рис. 24)

```
[guest2@localhost guest]$ rm /tmp/file01.txt
rm: невозможно удалить '/tmp/file01.txt': Операция не позволена
```

Попытка удалить файл

От имени суперпользователя снимаем с директории атрибут Sticky (рис. 25)

```
[guest2@localhost guest]$ su -
Пароль:
[root@localhost ~]# chmod -t /tmp
[root@localhost ~]# exit
выход
```

Смена атрибутов файла

Проверяем, что атрибут действительно снят (рис. 26)

```
[guest2@localhost guest]$ ls -l / | grep tmp
drwxrwxrwx.  20 root root 4096 окт  4 21:15 tmp
```

Проверка атрибутов директории

Далее был выполнен повтор предыдущих действий. По результатам без Sticky-бита запись в файл и дозапись в файл осталась невозможной, зато удаление файла прошло успешно (рис. 27)

```
[guest2@localhost guest]$ echo "test" > /tmp/file01.txt
[guest2@localhost guest]$ cat /tmp/file01.txt
test
[guest2@localhost guest]$ echo "test2" >> /tmp/file01.txt
[guest2@localhost guest]$ cat /tmp/file01.txt
test
test2
[guest2@localhost guest]$ echo "test3" > /tmp/file01.txt
[guest2@localhost guest]$ cat /tmp/file01.txt
test3
[guest2@localhost guest]$ rm /tmp/file01.txt
```

Повтор предыдущих действий

Возвращение директории tmp атрибута t от имени суперпользователя (рис. 28)

```
[guest2@localhost guest]$ su -
Пароль:
[root@localhost ~]# chmod +t /tmp
[root@localhost ~]# exit
выход
```

Изменение атрибутов

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.