

之前的逆向（5）-看雪-hellore

原创

Startr4ck 于 2018-04-28 07:06:28 发布 265 收藏

分类专栏: [ctf](#) 文章标签: [writeup](#)

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/shakeyin1998/article/details/80115378>

版权



[ctf](#) 专栏收录该内容

17 篇文章 0 订阅

订阅专栏

【题目链接】

<http://www.shiyanbar.com/ctf/1884>

【解题流程和思路】下载程序之后拖入OD进行逆向分析, 在比较call中进行暂停, 发现是输入的字符串进行加密之后与加密后的字符进行对比。那么主要的是寻找加密方式, 使用OD并不能快速找到加密方式。换做IDA进行分析。记住之前od关键call位置, 在IDA当中进行寻找



IDA当中

```
__main();
v14 = 0;
memset(&v15, 0, 0x10u);
memset(&v4, 0, 0x14u);
v4 = -404624154;
v5 = -70;
v6 = -12;
v7 = -27;
v8 = -13;
v9 = -12;
v10 = -12;
v11 = -27;
v12 = -13;
v13 = -12;
v16 = 0;
puts("喵?");
scanf("%s", &v14);
LOBYTE(v16) = 0;
while ( *((_BYTE *)&v14 + v16) )
    *((_BYTE *)&v14 + v16++) |= 0x80u;
if ( !strcmp((const char *)&v14, (const char *)&v4) )
    printf("good");
else
    printf("wrong");
return 0;
```

分析加密方式编写程序进行反加密即可。

如何编写程序?

分析加密方法:

对输入的字符串挨个与0X80进行或运算, 或运算之后对其进行比较

下面是个人理解:

0x80在16进制是128, ascall能够输出的字符是0-127, 所以与128进行或运算相当于与0进行或运算, 与0进行逆运算只需要减去0, 那么同

理。

单位是16进制，所以现将OD当中的字符串进行转换为16进制，转换之后减去128，减去之后再转换成字符串。小细节，这里直接得到hex来转16进制。不用字符串来转

地址	HEX 数据	ASCII
0022FEF4	E6 EC E1 E7 BA F4 E5 F3 F4 F4 E5 F3 F4 00 00 00	裸微呼警越警?
0022FEF4	00 00 00 00 B1	北北北北北

编写的程序

```
1 v4 = [0xE6, 0xEC, 0xE1, 0xE7, 0xBA, 0xF4, 0xE5, 0xF3, 0xF4, 0xF4, 0xE5, 0xF3, 0xF4, 0x00, 0x00, 0x00]
2 v14=[]
3 for i in range(len(v4)):
4     v14.append(v4[i]-0x80)
5 flag=""
6 for i in range(len(v14)):
7     flag+=chr(v14[i])
8 print(flag)
9
```

for i in range(...

```
C:\Users\殷浩钦\AppData\Local\Programs\Python\Python36\python.exe C:/Users/殷浩钦/PycharmProjects/untitled2/a.py
flag: testtest
Process finished with exit code 0
```