

西普学院逆向writeup -----你会吗？？

原创

[tiaotiaolong99234](#) 于 2015-06-21 00:38:42 发布 1703 收藏

分类专栏: [ctf](#) 文章标签: [西普学院](#) [windows](#) [逆向](#) [writeup](#) [ctf](#)

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/tiaotiaolong99234/article/details/46577139>

版权



[ctf](#) 专栏收录该内容

4 篇文章 0 订阅

订阅专栏

西普学院逆向writeup -----你会吗？？

-----跳跳龙

明天我们学校就要ctf比赛, 深夜前10几分钟正好解了一道逆向题, 是西普学院的, 昨天, 在小强的提醒下, 也解了一道, 今天就写写自己的writeup, 写完睡觉。

首先拿到re1.exe。运行一下, 欢迎我来到DUTCTF。说实话, 我都没在意, 接下来, 随便输入了123456.提示错误。

```
C:\Users\tangtianlong\Desktop\re1.exe
欢迎来到DUTCTF呦
这是一道很可爱很简单的逆向题呦
输入flag吧:123456
flag不太对呦, 再试试呗, 加油呦
请按任意键继续...
http://blog.csdn.net/tiaotiaolong99234
```

接下来, 我就放在IDA里看了一下, (友情提示: 大家可以尝试一下只做自己的sig文件, 使逆向效果更加完美), 流程非常清晰, 感觉不是很难。

```
IDA View-A x Pseudocode-D x Pseudocode-C x Pseudocode-B x Pseudo
1 int sub_401000()
2 {
3   char v0; // ST10_1@1
4   char v1; // ST0C_1@1
5   int v2; // eax@1
6   __int128 v4; // [sp+0h] [bp-44h]@1
7   __int64 v5; // [sp+10h] [bp-34h]@1
8   int v6; // [sp+18h] [bp-2Ch]@1
9   __int16 v7; // [sp+1Ch] [bp-28h]@1
10  char v8; // [sp+20h] [bp-24h]@1
11
12  __mm_storeu_si128((__m128i *)&v4, __mm_loadu_si128((const __m128i *)&v5));
13  v6 = 0; http://blog.csdn.net/tiaotiaolong99234
14  __mm_storel_epi64((__m128i *)&v5, __mm_loadl_epi64((const __m128i *)&v7));
15  v7 = 0;
16  sub_40127B((int)"欢迎来到DUTCTF呦\n", v0);
17  sub_40127B((int)&kunk_413E60, v0);
18  sub_40127B((int)"输入flag吧:", v1);
19  sub_4010D1("%s", &v8);
20  v2 = strcmp((const char *)&v4, &v8);
21  if ( v2 )
22    v2 = -(v2 < 0) | 1;
23  if ( v2 )
24    sub_40127B((int)"flag不太对呦, 再试试呗, 加油呦\n", v0);
25  return v6;
}
```

接下来，发现，sub_40127B是输出函数，sub_4010D1是输入函数，并且自己的输入存在&v8中，与&v4进行比较，如果结果不相等，便提示不对。可见这个函数在执行时，也就是在同一个函数栈帧区间。内，v8在 [sp+20h]，v4在[sp+0h]。可见有思路了，接下来OD闪亮登场，（话说逆向题就是OD与IDA的完美结合，分析加密算法）

首先我看了一下字符串，这些中文字符是找不到了，但是我们会看见pause这个词，就是在函数的末尾执行system（“PAUSE”）的时候，压入栈的。所以很快定位。然后便是执行函数，等待程序执行输入我们的数据，与真实的flag进行比对。此时我们输入123456之后，便压入了flag。

```
寄存器 (CPU)
EAX 00000001
ECX 00FD115D re1.00FD115D
EDX 006A8768
EBX 00000000
ESP 0038FD34 ASCII "DUTCTF{Me1c0met0DUTCTF}"
EBP 0038FD78 http://blog.csdn.net/tiaotiaolong99234
ESI 00000000
EDI 00000000
EIP 00FD1062 re1.00FD1062
```

但是，思路并没有结束，v4和v8地址相差0x20，v8=bp-24h=0038fd78-24=38fd54,v4=0038fd34.正好相差20h，和我们预想的一样。明天考试，滚去睡。。。