

# 一道简单的逆向题目(取于2019成信大四叶草安全杯)

原创

ascar\_奥斯卡  于 2019-08-07 14:23:57 发布  311  收藏 2

分类专栏: [逆向](#) 文章标签: [逆向](#)

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/qq\\_42874910/article/details/98745411](https://blog.csdn.net/qq_42874910/article/details/98745411)

版权



[逆向](#) 专栏收录该内容

1 篇文章 0 订阅

订阅专栏

## 代码分析的逆向题目

### 基础工具

[ida](#)

[ollydbg](#)

[peid](#)

### 开始分析

[查壳](#)

[ida静态分析](#)

[杂谈](#)

## 基础工具

工欲善其事必先利其器, 做逆向题目时好的工具必不可少, peid,od和ida为本题练习需要的工具 //其实od在本题也非必须工具

### ida

静态分析神器

### ollydbg

逆向必备神器

### peid

查壳工具

## 开始分析

首先运行一下

```
(n_n)o  
Now look for what I want to say (n_n)o
```

随便输入一段字符后，程序就退出了

## 查壳

拖入peid发现无壳



## ida静态分析

拖入ida，使用快捷键f5反编译主函数

The screenshot displays the IDA Pro interface with the main function decompiled into C-like pseudocode. The interface includes a menu bar (File, Edit, Jump, Search, View, Debugger, Options, Windows, Help), a toolbar, and several windows: Functions window, IDA View-A, Pseudocode-A, Hex View-1, Structures, and Enums. The main window shows the following code:

```
1 int __cdecl main(int argc, const char **argv, const char **envp)
2 {
3     int v3; // ecx@1
4     int v4; // eax@1
5     unsigned int v5; // kr08_4@1
6     int result; // eax@2
7     int i; // ecx@3
8     char v8; // al@4
9     char v9; // dl@6
10    char v10[1024]; // [sp+8h] [bp-800h]@1
11    char v11; // [sp+408h] [bp-400h]@1
12
13    sub_404A50(aB_b0);
14    sub_404A50(&unk_414128);
15    v3 = *(DWORD *)(&word_417DB8 + 4);
16    v4 = sub_401180(10);
17    sub_401330(v10, 1023, v4);
18    strcpy(&v11, v10);
19    v5 = strlen(v10) + 1;
20    if ( v5 - 1 != strlen(aQht_2019_gsqqz) )
21    {
22        sub_404A50(aYou_can_doit);
23        return -1;
24    }
25    for ( i = 0; i <= (signed int)(v5 - 2); ++i )
26    {
27        v8 = v10[i];
28        if ( v8 > 90 || v8 < 65 )
29        {
30            if ( v8 > 122 || v8 < 97 )
31                continue;
32            v9 = (v8 - 83) % 26 + 97;
33        }
34        else
35        {
36            v9 = (v8 - 51) % 26 + 65;
37        }
38        v10[i] = v9;
39    }
40    if ( !strcmp(aQht_2019_gsqqz, v10) )
41    {
42        sub_404A50(aCongratulation);
43        system(aPause);
44        result = 0;
45    }
46 }
```

现在来分析代码

```

// 有用的代码
if ( v5 - 1 != strlen(aQht_2019_gsqzc) ) //判断输入字符串是否长度等于"Qht_2019_Gsqzcjsf_Vszzc"这个字符串长度
{
    sub_404A50(aYou_can_doit);
    return -1;
}
for ( i = 0; i <= (signed int)(v5 - 2); ++i ) //
{
    v8 = v10[i]; //将输入字符串转为一个个字符进行运算
    if ( v8 > 90 || v8 < 65 ) //判断字符是否不在Z~A之间
    {
        if ( v8 > 122 || v8 < 97 )
            continue; //若字符不在a~z之间不运算
        v9 = (v8 - 83) % 26 + 97; //字符在a~z之间, 英文字母向后移14位
    }
    else
    {
        v9 = (v8 - 51) % 26 + 65; //字符在A~Z之间, 英文字母向后移14位
    }
    v10[i] = v9; //一个个字符转换为字符串
}
if ( !strcmp(aQht_2019_gsqzc, v10) ) //判断运算后字符串是否等于"Qht_2019_Gsqzcjsf_Vszzc"
{
    sub_404A50(aCongratulation);
    system(aPause);
    result = 0;
}
else
{
    sub_404A50(aVeryclose_succ);
    result = -1;
}
return result;

```

这里我们就能知道对“Qht\_2019\_Gsqzcjsf\_Vszzc”这个字符串进行逆运算应该就可得到flag。

逆运算结果为 `Ctf_2019_Seclover_Hello`

## 杂谈

之所以我用到了od，是因为我直接通过修改判断，到达了成功的程序部分，然而我发现修改后的程序无论我输入什么，flag就是我输入的字符串，比赛的时候和小组成员讨论了一下，然后再次审计代码，才发现了问题所在，总的来说还是需要更多学习吧。