

# 2实验吧逆向

原创

卦星 于 2016-06-09 00:01:10 发布 1545 收藏

分类专栏: [逆向学习记录](#) 文章标签: [实验吧](#) [babycrack](#)

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/zhyl025907/article/details/51615911>

版权



[逆向学习记录](#) 专栏收录该内容

17 篇文章 6 订阅

订阅专栏

## 1.1 概述

随着对安全的逐渐深入的学习, 在接触部分linux知识的过程中, 认识了不少CTF的大神, 这些大神有不少是顶级赛棍, 奈何非科班出身的计算机基础差的要命, 只能一点一点积累和记录了

这一系列文章仅仅是自己的学习记录, 写博客督促自己不断学习的, 其实如果查资料, 并没有啥有价值的东西

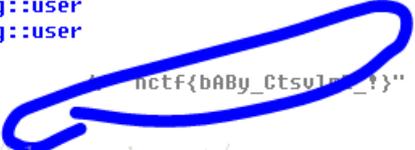
本次记录的是我的第二次逆向题目, 来自

## 1.2 实验吧的“babycrack”? 开始逆向

### 1.2.1 使用PEID查看是否存在壳, (略)

1.2.2 导入IDA一看, 是一个.net程序, 比较坑爹 (这个真不懂, 以前写过一点MFC, 不过, 后来都忘了, 硬着头皮点下去)

```
sts+10 string Easy_CM.Config::user
ldsflld string Easy_CM.Config::user
stloc.1
ldstr aHctfBaby_ctsul "nctf{bABy_Ctsulm_?}"
stloc.2
ldloc.2
ldloc.1
callvirt instance int32 [mscorlib]System.String::CompareTo(string)
stloc.5 4
ldloc.5 4
ldc.i4.0
```



投机取巧了, 不好意思, 这个真没分析出来逻辑, 但是字符串就在那放着!  
我也明白为啥是babycrack了

## 1.3实验吧的“阿拉神灯”? 开始逆向

```

aForm1:          // DATA XREF: WindowsApplication1.Form1__InitializeComponent+29B'
  unicode <Form1>,0
asc_1134:        // DATA XREF: WindowsApplication1.Form1__InitializeComponent+29B'
  unicode <通关密码>,0
aZhimakaimen@20: // DATA XREF: WindowsApplication1.Form1__Button1_Click+12f0
  unicode <zhimakaimen@2011>,0
asc_1170:        // DATA XREF: WindowsApplication1.Form1__Button1_Click+20f0
  unicode <通关密码正确! >,0
asc_1170:        // DATA XREF: WindowsApplication1.Form1__Button1_Click:loc_7B3f0
  unicode <通关密码错误! >,0
aWindowsapplica: // DATA XREF: WindowsApplication1.My.Resources.Resources__get_Re
  unicode <WindowsApplication1.Resources>,0
aM:              // DATA XREF: WindowsApplication1.My.Resources.Resources__get_m+!
  unicode <m>,0

```

字符串中有明文!

1.4实验吧的“just click”? 开始逆向  
使用IDA折腾一上午, 后来才明白, 其实c#可以更简单的ILSpy工具

```

// rev4.MainWindow
private void btn_Checker(int para)
{
  int[] array = new int[]
  {
    0,
    1,
    3,
    4,
    2,
    1,
    2,
    3,
    4
  };
  bool flag = this.hit < 8;
  if (flag)
  {
    int num = this.hit;
    this.hit = num + 1;
  }
  else
  {
    base.Close();
  }
  bool flag2 = this.first;
  if (flag2)
  {
    this.tb1.Text = "";
    this.first = false;
  }
  bool flag3 = array[this.hit] == para;
  if (flag3)
  {
    int num = this.correct;
    this.correct = num + 1;
  }
}

```

1.4 100000

拖入ida 逆向出算法, 结果如下

```

18 int v19, // [sp+00h] [up-00h]
19
20 __main();
21 *(_DWORD *)Str1 = 0;
22 memset(&v18, 0, 0x10u);
23 memset(&v17, 0, 0x14u);
24 v4 = -26;
25 v5 = -20;
26 v6 = -31;
27 v7 = -25;
28 v8 = -70;
29 v9 = -12;
30 v10 = -27;
31 v11 = -13;
32 v12 = -12;
33 v13 = -12;
34 v14 = -27;
35 v15 = -13;
36 v16 = -12;
37 v19 = 0;
38 puts("#");
39 scanf("%s", Str1);
40 LOBYTE(v19) = 0;
41 while ( Str1[v19] )
42 {
43     Str1[v19] |= 0x80u;
44     ++v19;

```

正向是和0x80u求|.反向应该是求& 0x7f  
得到结果，