攻防世界——debug

原创

chan3301 于 2019-08-23 11:57:13 发布 888 收藏 3

分类专栏： 逆向题目练习 文章标签： 逆向 攻防世界

本文链接：https://blog.csdn.net/sjt670994562/article/details/100034691

逆向题目练习 专栏收录该内容

16 篇文章 1 订阅

订阅专栏

emmmmmmm算是一个.net编译文件分析开始的一个里程碑吧，放到ida\OD里面发现都没法分析，用exeinfope查看是.net编译，开始以为是.net加壳，搜了很多，也没成功，ida照样没法分析，后来才明白应该用其他的反汇编工具，之前我用的.net refector,但经过了这一次感觉这个软件没有dnspy好用，所以强烈安利dnspy

查看源代码，全都在02000003里面


```
74          A_2 = .(A_2);
75      }
76
77      // Token: 0x06000008 RID: 8 RVA: 0x000021F0 File Offset: 0x000003F0
78      private static void (string[] A_0)
79      {
80          string b = null;
81          string value = string.Format("{0}", DateTime.Now.Hour + 1);
82          string a_ = "CreateByTenshine";
83          .(a_, Convert.ToInt32(value), ref b);
84          string a = Console.ReadLine();
85          if (a == b)
86          {
87              Console.WriteLine("u got it!");
88              Console.ReadKey(true);
89          }
90          else
91          {
92              Console.Write("wrong");
93          }
94          Console.ReadKey(true);
95      }
96  }
97
```

发现了不得了的东西

```
    }

    // Token: 0x06000006 RID: 6 RVA: 0x00002144 File Offset: 0x00000344
    private static string (string A_0)
    {
        byte[] bytes = Encoding.ASCII.GetBytes(A_0);
        return "flag{" + BitConverter.ToString(new MD5CryptoServiceProvider().ComputeHash(bytes)).Replace("-", "") + "}";
    }
```

在这边，他并没有检测就直接打出了flag，所以直接运行，中间中断在入口，单步往后面走，查看变量b就能知道flag了，这难度也是6666666

```
94          Console.ReadKey(true);
95      }
```

100 %

局部变量

名称	值	类型
System.Convert.ToInt32 返回	0x0000000C	int
A_0	{string[0x00000000]}	string[]
a	null	string
b	"flag{967DDDFBCD32C1F53527C221D9E40A0B}"	string
value	"12"	string
a_	"CreateByTenshine"	string

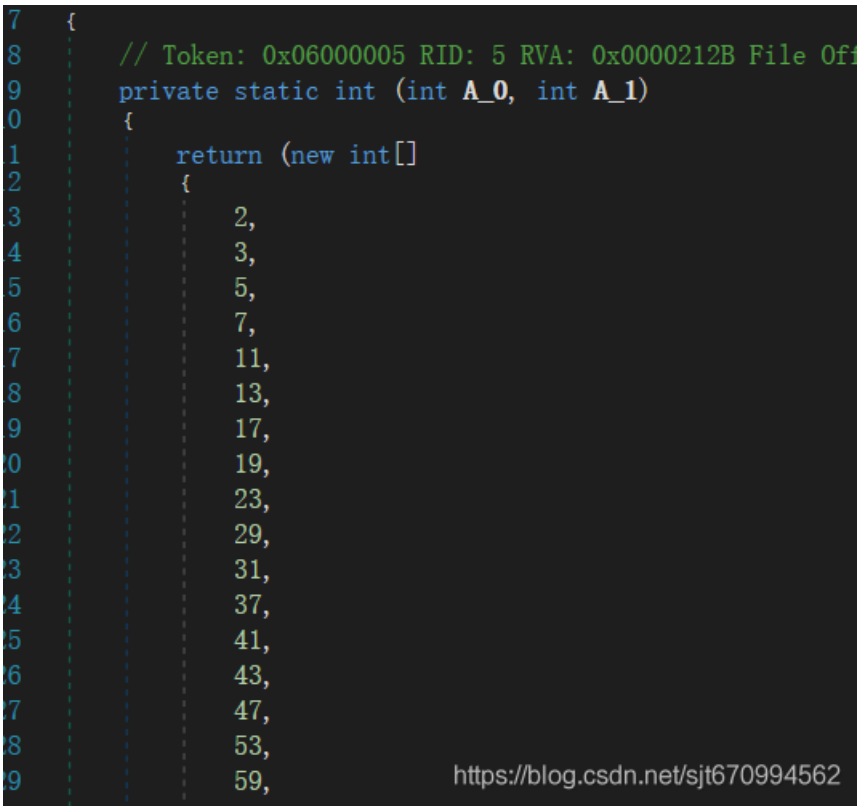
但是为了学习嘛，所以还是看了一下算法，并且学习一下.net语言的书写方法
这里是关键转变地点

```
string a_ = "CreateByTenshine";
.(a_, Convert.ToInt32(value), ref b);
```

跟进去调试，发现主要是在这里进行的操作

```
// Token: 0x06000007 RID: 7 RVA: 0x0000218C File Offset: 0x0000038C
private static void (string A_0, int A_1, ref string A_2)
{
    int num = 0;
    if (0 < A_0.Length)
    {
        do
        {
            char c = A_0[num];
            int num2 = 1;
            do
            {
                c = Convert.ToChar(.(Convert.ToInt32(c), num2));
                num2++;
            }
            while (num2 < 15);
            A_2 += c;
            num++;
        }
        while (num < A_0.Length);
    }
    A_2 = .(A_2);
}
```


大致的操作就是根据字符串CreateByTenshine让其每个字符都进行一次15循环的异或操作，异或数值在下面的这个表里面（这里的15是个坑，开头是1不是0也是个坑，一定要看好）



然后贴出python代码

```
import hashlib
a=[2,3,5,7,0xb,0xd,0x11,0x13,0x17,0x1d,0x1f,0x25,0x29,0x2b,0x2f,0x35,0x3b,0x3d,0x43,0x47,0x49,0x4f,0x53,0x59,0x6
1,0x65,0x67,0x6b,0x6d,0x71]
b="CreateByTenshine"
c=""
for i in range(len(b)):
    d=ord(b[i])
    for j in range(1,15):
        d=a[j]^d
    c+=chr(d)
print(c)
m=hashlib.md5(c.encode("utf-8")).hexdigest()
print("flag{",m,"}")
```

（这里flag转一下大写，小写好像不行）