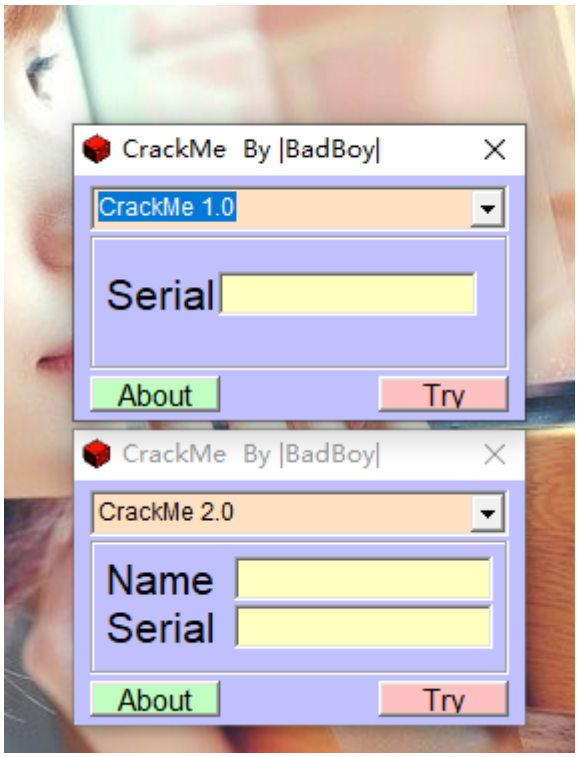
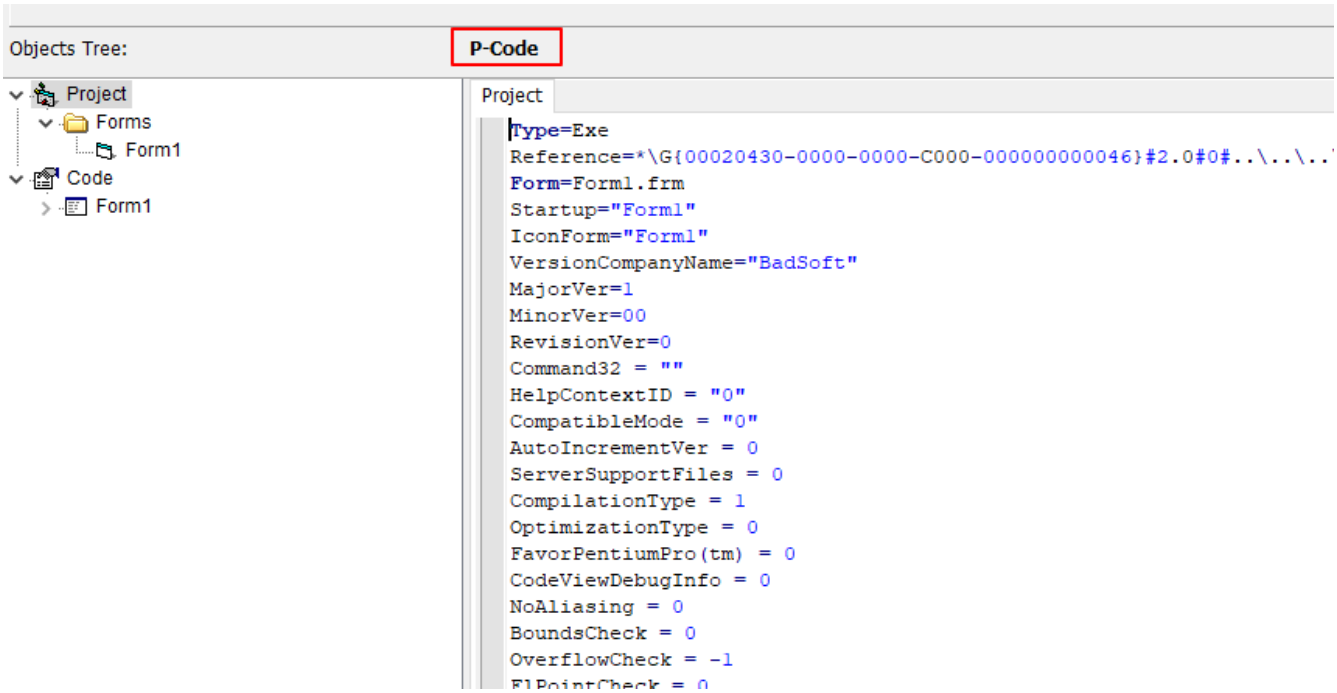
前言

之所以直接跳过Crackme012是因为那个是个16位的程序，放到现在来说就算逆出来了意义也不大，想要追求完美的同学可以去看下逆向驿站公众号号主发的文章。

分析程序



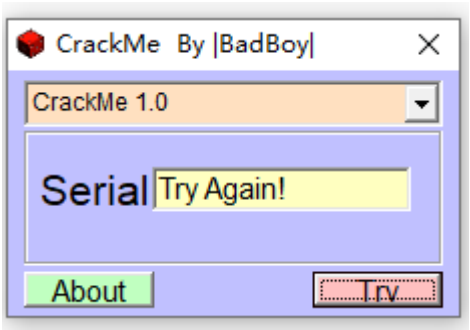
这个Crackme有两个，一个是单纯的序列号，另一个是用户名和序列号的保护方式。



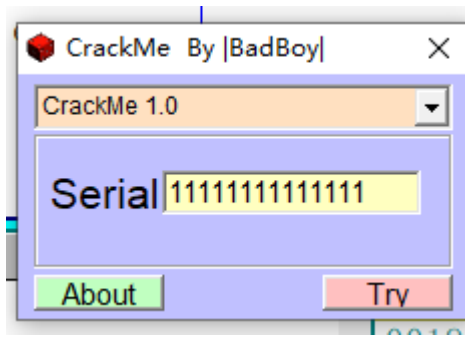
用VB反汇编工具来看下，是P-Code编译的，唉，头疼，直接用OD看吧

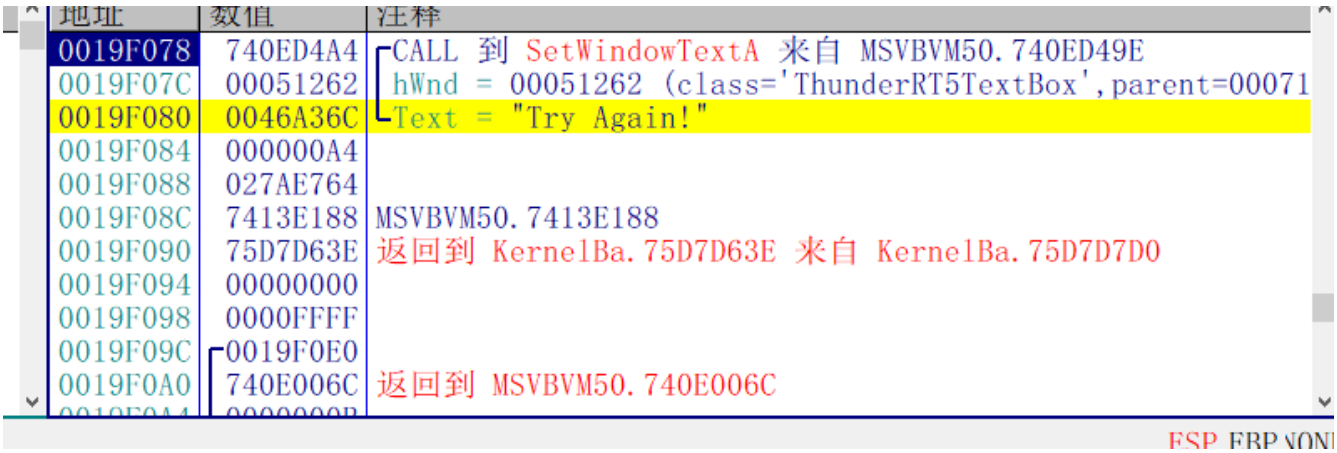
OD调试程序

首先来看1.0



根据这个错误的提示，直接在SetWindowTextA/W上下断点，输入一个假序列号

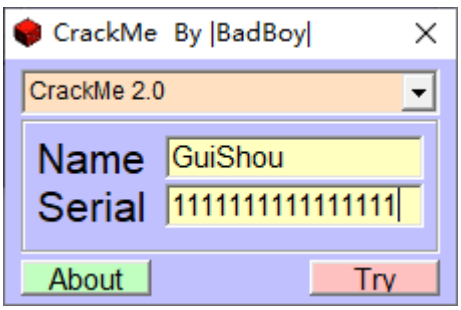




程序断下之后观察堆栈，往上拉，找到了一串字符串，

试试是不是这个，输入7723012

好 成功了，来看第二个，

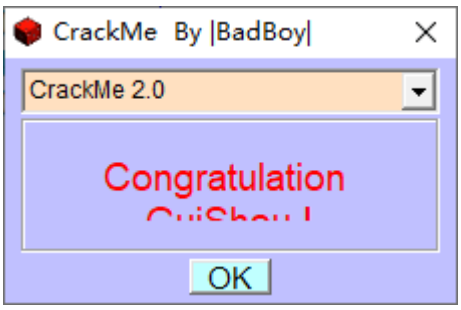


同样，输入用户名和密码

```
地址      数值      注释
0019EF20  740ED4A4  CALL 到 SetWindowTextA 来自 MSVBVM50.740ED49E
0019EF24  00031266  hWnd = 00031266 (class='ThunderRT5TextBox',parent=00041
0019EF28  004AC834  Text = "Try Again!"
0019EF2C  000000A4
0019EF30  027AE9EC
0019EF34  7413E188  MSVBVM50.7413E188
0019EF38  75D7D63E  返回到 KernelBa.75D7D63E 来自 KernelBa.75D7D7D0
0019EF3C  00000000
0019EF40  0000FFFF
0019EF44  0019EF88
0019EF48  740E006C  返回到 MSVBVM50.740E006C
```

程序断下后，一直往下拉，找到了这么一串字符串，输入进去试试

```
地址      数值      注释
0019EFB0  000000A4
0019EFB4  004058E3  badboy.004058E3
0019EFB8  FFFFFFFF
0019EFBC  004AC834  ASCII "Try Again!"
0019EFC0  0019F1D0
0019EFC4  74110D43  返回到 MSVBVM50.74110D43 来自 MSVBVM50.740E4417
0019EFC8  027AE9EC
0019EFCC  0000000B
0019EFD0  00000004
0019EFD4  00403C70  UNICODE "Try Again!"
0019EFD8  741BE836  返回到 MSVBVM50.741BE836
0019EFDC  027AE9EC
0019EFE0  00403C70  UNICODE "Try Again!"
0019EFE4  00000000
0019EFE8  004ACB54  UNICODE "66494-499749"
0019EFEC  00000000
0019EFF0  00000000
0019EFF4  00000000
0019EFF8  00000000
0019EFFC  00000000
0019F000  411E8094
0019F004  00000005
```



居然又成功了 哈哈。

到这里如果你的目的只是想破解这个程序，那么目的已经达成了，如果你是想分析算法或者了解P-Code的原理，那就需要深入分析了。由于微软对这一块的文档是保密的，到目前为止并没有系统的教程，只有在论坛上的几篇文章，我也就放弃了。

需要相关文件的可以到我的Github下载：https://github.com/TonyChen56/160-Crackme