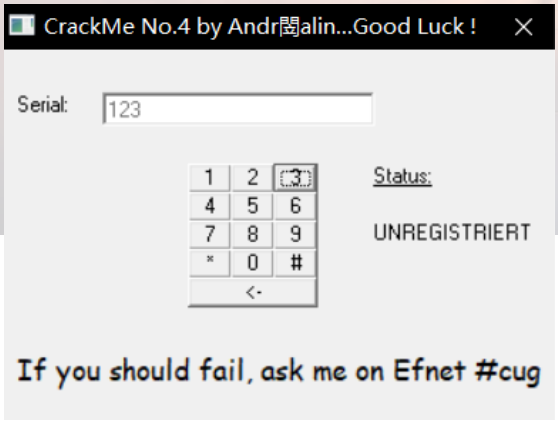
CrackMe011

Crackme011 的逆向分析

1.程序观察

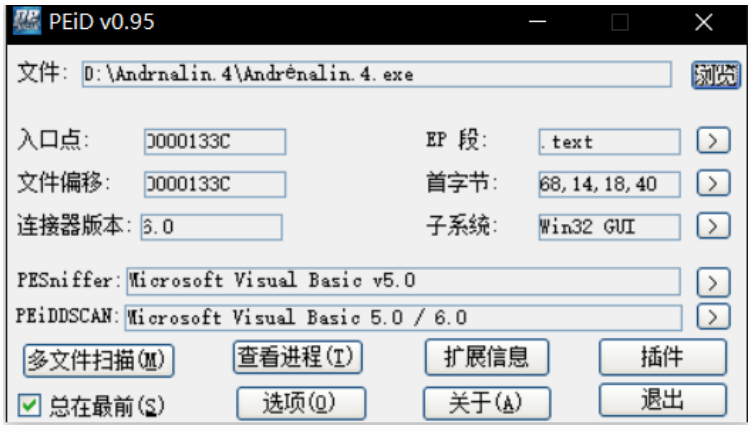


可以看到，程序只有让输入的地方，没有确认按钮什么的。
在程序左侧，写着 Status: UNREGISTRIERT。

猜想：

1. 程序会根据输入框的变化事件来判断是否正确（其实是不正确的，到后面就知道了）。
2. 输入正确的序列号程序旁边的 Status 会改变。

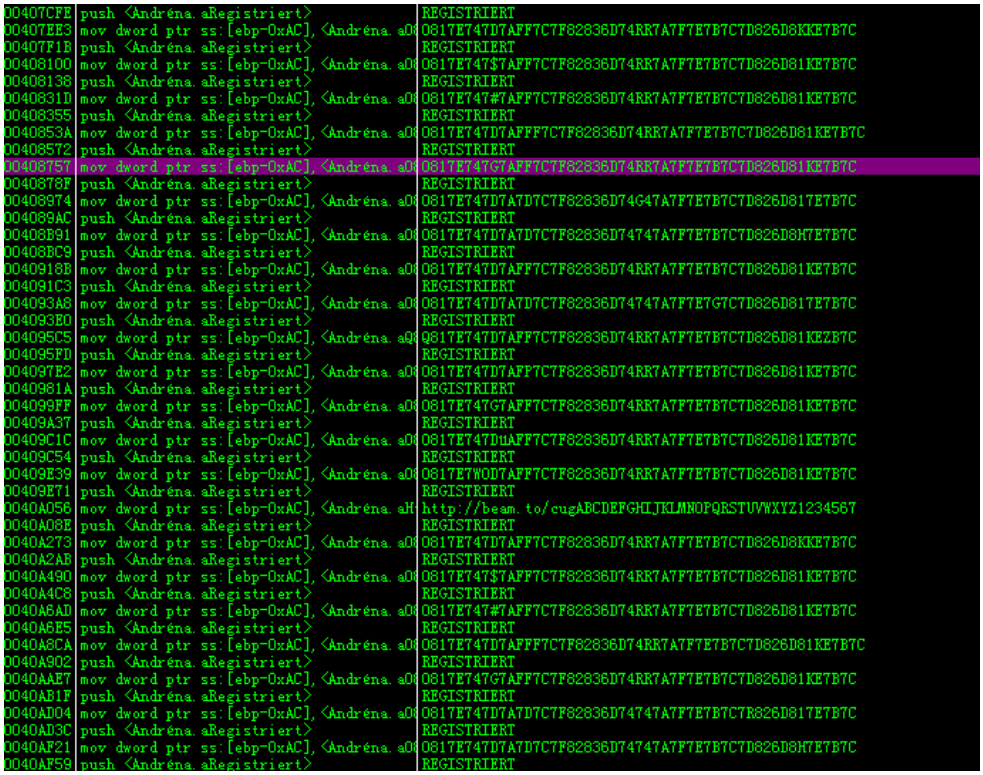
2.简单查壳



使用 VB 编写的程序，没有壳。

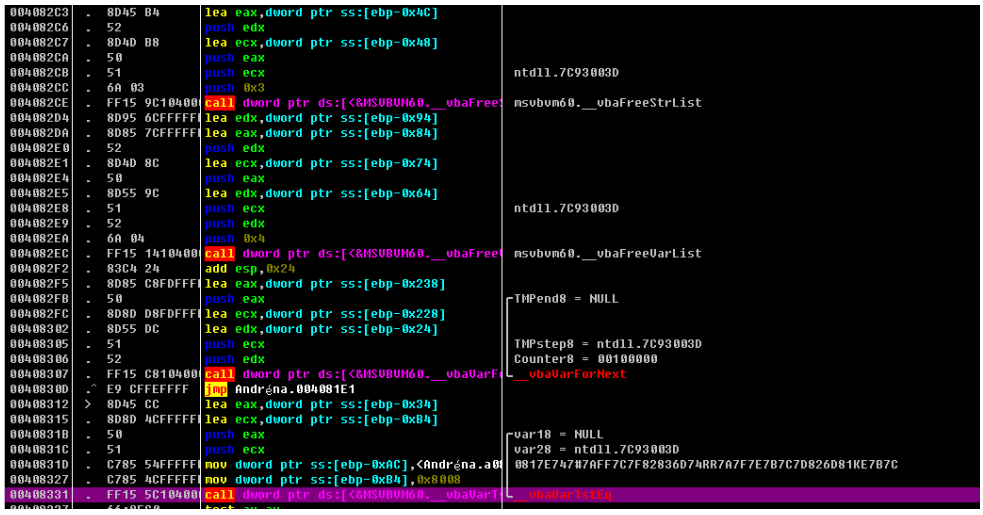
3.程序分析

使用 OD 载入程序，搜索字符串。



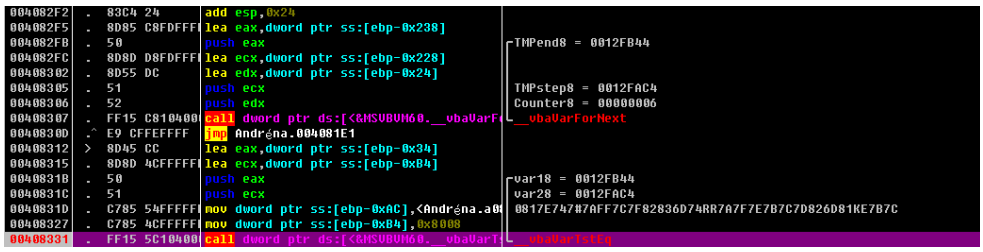
可以看到有很多的很相似的字符串，还有很多个 "REGISTRIERT"。

我们也不知道这是什么东西，随便双击一个进入代码。

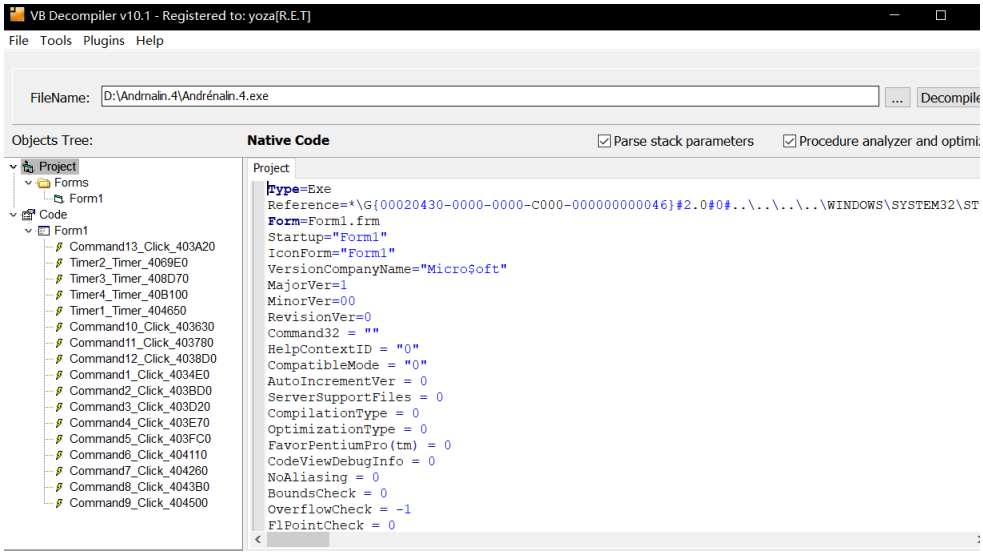


可以发现，这代码和 009 和 010 非常相像。
都是先求得输入的长度，然后建立循环，最后再进行比较。
但是这里，一个代码块有很多的这样的组合。

这是为什么呢?
我们先在比较函数处下断点。

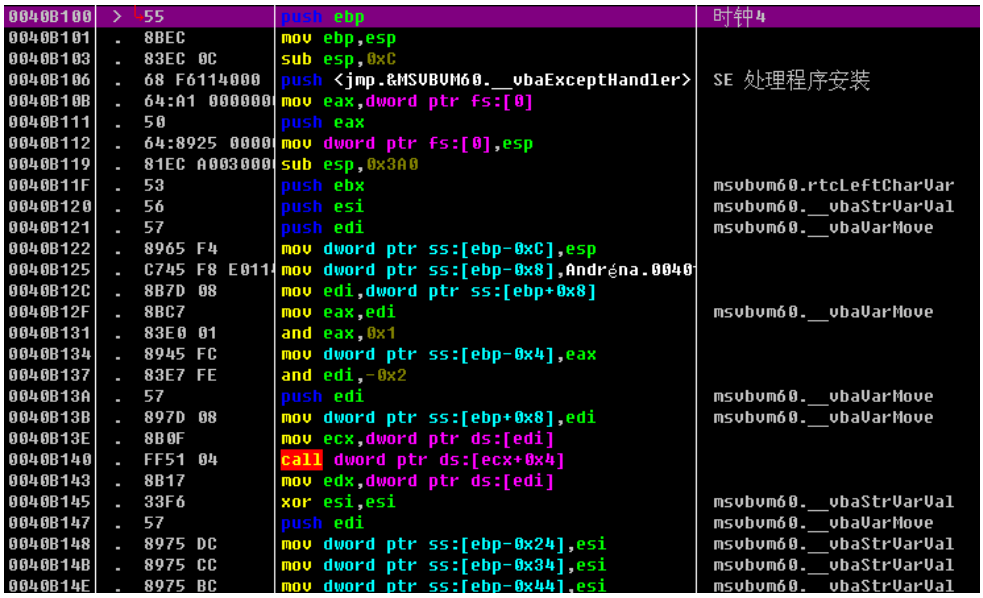


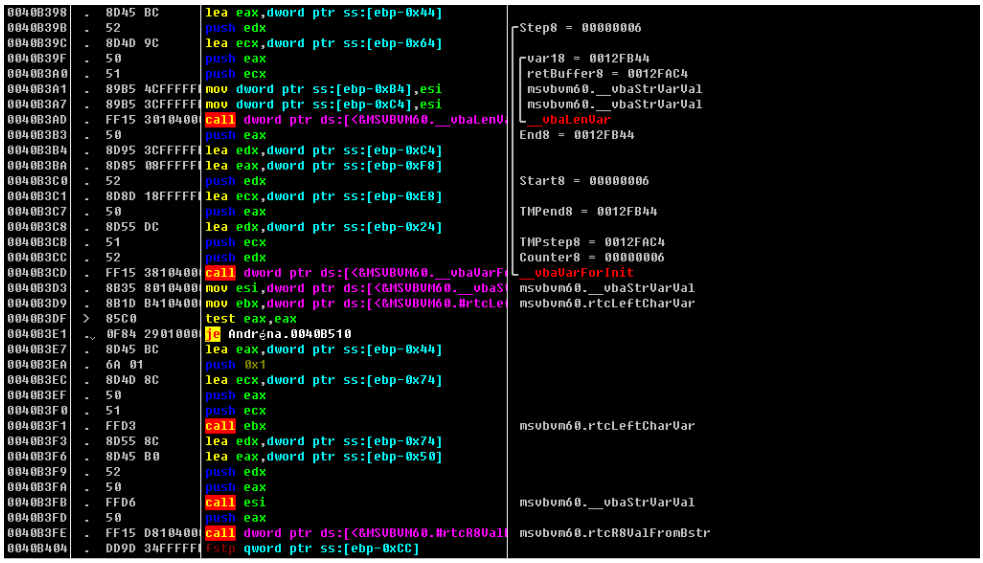
我们没有进行任何操作，程序就立刻中断了。
猜想程序可能使用了定时器。我们使用 VB Decompiler 加载程序



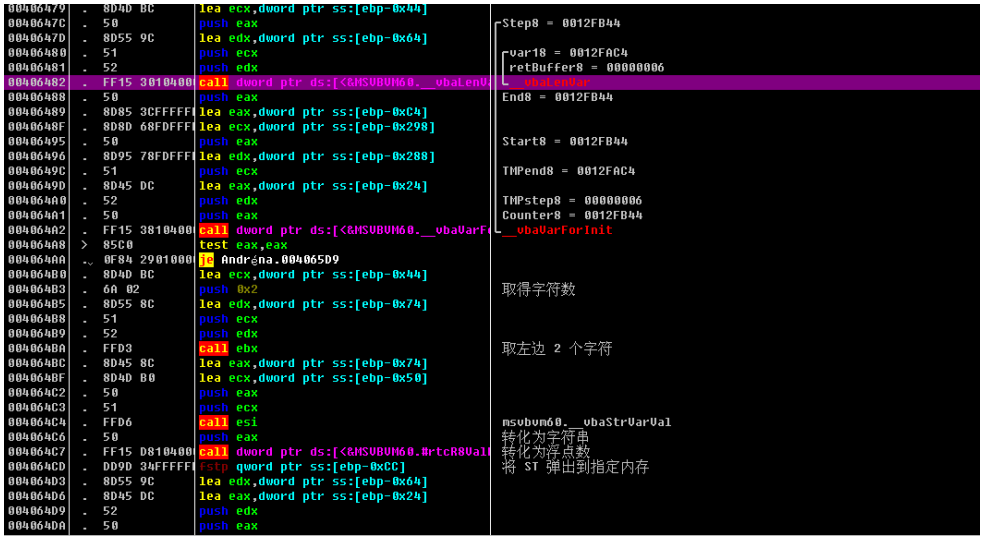
可以看到有4个定时器。

每个定时器代码块，里面都是很多个上面那样的代码。

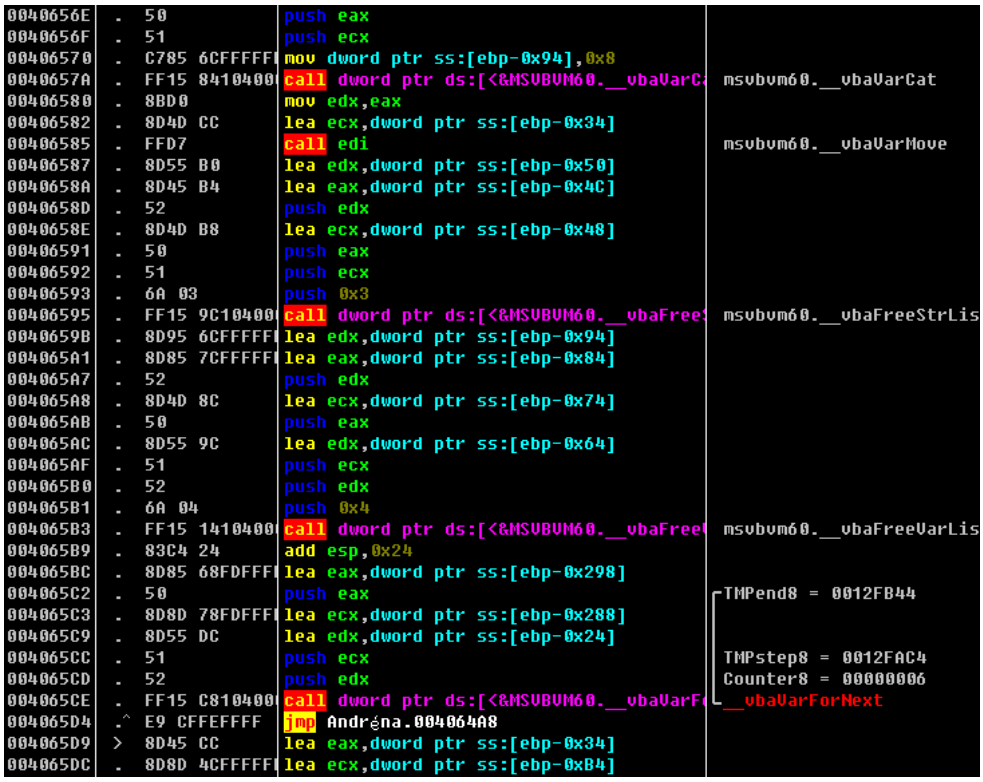


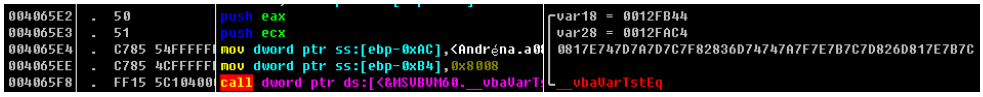


先分析程序的算法


Step8 = 0012FB44
var18 = 0012FAC4
retBuffer8 = 00000006
End8 = 0012FB44
Start8 = 0012FB44
TMPend8 = 0012FAC4
TMPstep8 = 00000006
Counter8 = 0012FB44
__vbaVarForInit
取得字符数
取左边 2 个字符
msvbvm60.__vbaStrVarVal
转化为字符串
转化为浮点数
将 ST 弹出到指定内存


msvbvm60.__vbaI4Var
转化为字符串
取一个字符
edx = 取得字符
msvbvm60.__vbaStrMove
String = "→"
转化为十六进制
将hex转化为10进制弹出到ST0
将 ST0 弹回该地址
加上前两个字符的浮点数
相加的值保存在该地址
将相加的值转化为hex





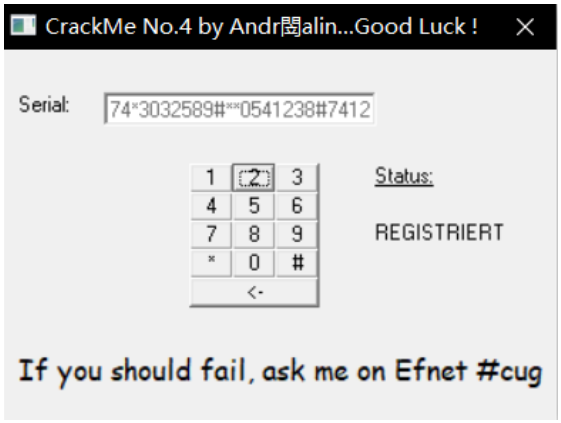
1. 程序先计算出我们输入的序列号的长度
2. 建立循环，循环次数是序列号的长度
3. 在循环中，程序取得我们输入序列号的前两个字符，将其转化为浮点数，保存在内存中
4. 然后程序依次取得序列号的单个字符，将其转化为使用10进制表示的 ASCII 码
5. 将上述两个值相加，再转化为十六进制
6. 将每个字符和前两个字符相加的值连接起来

最后程序将这个字符串和一串字符串相比较，相同就注册成功。

由于我们输入的序列号最后转化的字符串是十六进制表示的，所以相比较的字符串也应该是在 0-9，A-F 范围内的，这样我们很容易就确定了真正的用来比较的字符串："081 7E 74 7D 7A 7D 7C 7F 82 83 6D 74 74 7A 7F 7E 7B 7C 7D 82 6D 8H 7E 7B 7C"。

由于最后生成的字符串，是和我们输入的序列号的前两位息息相关的。
前两位越大，最后每个字符相对应的也就越大。
前两位不一样，最后每个数字相对应的也就不一样。

根据正确的比较码，我们可以反推出正确的序列号：
74*3032589#**0541238#7412



相关文件在我的 Github：https://github.com/UnreachableLove/160-Crackme/tree/master/Crackme011

分类: 160 Crackme

随风而逝的白色相簿
关注 - 4
粉丝 - 1

« 上一篇： Crackme010

posted @ 2019-09-12 11:02 随风而逝的白色相簿 阅读(5) 评论(0) 编辑 收藏