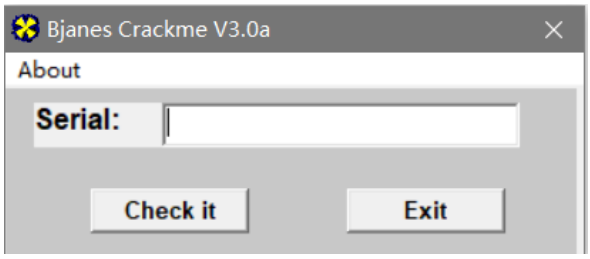
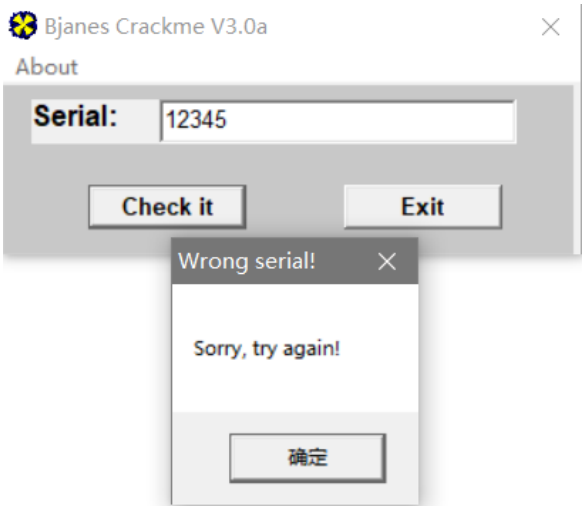
随风而逝的白色相簿

Crackme017

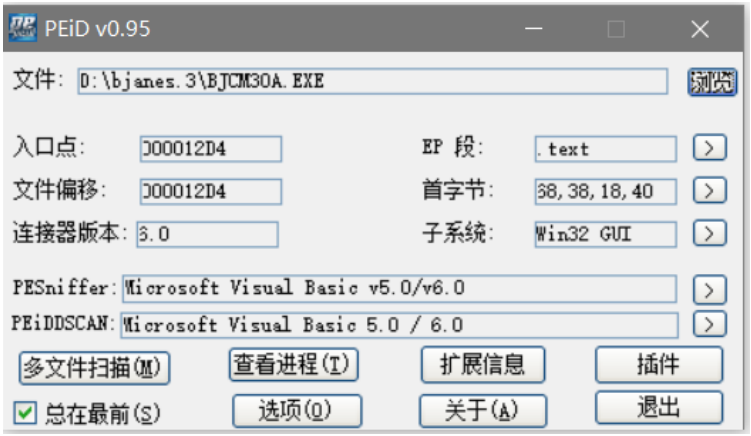
Crackme017 的逆向分析

1.程序观察





2.简单查壳



使用 VB5 编写的，没有壳。

3.程序分析

1. Crackme014(1)

使用 OD 载入程序，搜索字符串，点击跟进代码

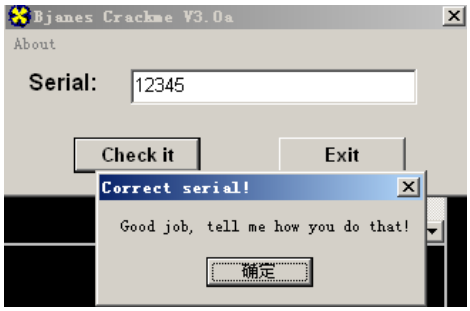
```
00404D5F  mov edx,BJCM30A.00402900                =
00404E11  mov dword ptr ss:[ebp-0xF0],BJCM30A.004  FFFF
00404E6A  mov dword ptr ss:[ebp-0x100],BJCM30A.00  Correct serial!
00404E88  mov dword ptr ss:[ebp-0xF0],BJCM30A.004  Good job, tell me how you do that!
00404F17  mov dword ptr ss:[ebp-0x100],BJCM30A.00  Wrong serial!
00404F35  mov dword ptr ss:[ebp-0xF0],BJCM30A.004  Sorry, try again!
```

```
00404E08  .  8D55 CC      lea edx,dword ptr ss:[ebp-0x34]
00404E0B  .  8D85 08FFFFFF lea eax,dword ptr ss:[ebp-0xF8]
00404E11  .  C785 10FFFFFF mov dword ptr ss:[ebp-0xF0],BJCM30A.00402B58   FFFF
00404E1B  .  52           push edx                                        var18 = 064D0220
00404E1C  .  50           push eax                                        var28 = 06CC0D70
00404E1D  .  C785 08FFFFFF mov dword ptr ss:[ebp-0xF8],0x8008             __vbaVarTstEq
00404E27  .  FF15 6C104000 call dword ptr ds:[<&MSVBVM60.__vbaVarTstEq>]
00404E2D  .  66:85C0      test ax,ax
00404E30  .  0F84 AD00000( je BJCM30A.00404EE3
00404E36  .  8B1D CC10400( mov ebx,dword ptr ds:[<&MSVBVM60.__vbaVarDup>]  msvbvm60.__vbaVarDup
00404E3C  .  B9 04000280  mov ecx,0x80020004
00404E41  .  898D 20FFFFFF mov dword ptr ss:[ebp-0xE0],ecx
00404E47  .  B8 0A000000  mov eax,0xA
00404E4C  .  898D 30FFFFFF mov dword ptr ss:[ebp-0xD0],ecx
00404E52  .  8D95 F8FEFFFF lea edx,dword ptr ss:[ebp-0x108]
00404E58  .  8D8D 38FFFFFF lea ecx,dword ptr ss:[ebp-0xC8]
00404E5E  .  8985 18FFFFFF mov dword ptr ss:[ebp-0xE8],eax
00404E64  .  8985 28FFFFFF mov dword ptr ss:[ebp-0xD8],eax
00404E6A  .  C785 00FFFFFF mov dword ptr ss:[ebp-0x100],BJCM30A.00402BB4  Correct serial!
00404E74  .  89B5 F8FEFFFF mov dword ptr ss:[ebp-0x108],esi
00404E7A  .  FFD3         call ebx                                        <&MSVBVM60.__vbaVarDup>
00404E7C  .  8D95 08FFFFFF lea edx,dword ptr ss:[ebp-0xF8]
00404E82  .  8D8D 48FFFFFF lea ecx,dword ptr ss:[ebp-0xB8]
00404E88  .  C785 10FFFFFF mov dword ptr ss:[ebp-0xF0],BJCM30A.00402B68  Good job, tell me how you do that!
00404E92  .  89B5 08FFFFFF mov dword ptr ss:[ebp-0xF8],esi
00404E98  .  FFD3         call ebx
```

可以看到不远处就有一个比较语句，下断点，运行程序，程序断在了断点处

```
00404E05  .  83C4 10      add esp,0x10
00404E08  .  8D55 CC      lea edx,dword ptr ss:[ebp-0x34]
00404E0B  .  8D85 08FFFFFF lea eax,dword ptr ss:[ebp-0xF8]
00404E11  .  C785 10FFFFFF mov dword ptr ss:[ebp-0xF0],BJCM30A.00402B58   FFFF
00404E1B  .  52           push edx                                        var18 = 00000040
00404E1C  .  50           push eax                                        var28 = NULL
00404E1D  .  C785 08FFFFFF mov dword ptr ss:[ebp-0xF8],0x8008             __vbaVarTstEq
00404E27  .  FF15 6C104000 call dword ptr ds:[<&MSVBVM60.__vbaVarTstEq>]
00404E2D  .  66:85C0      test ax,ax
00404E30  .  0F84 AD00000( je BJCM30A.00404EE3
00404E36  .  8B1D CC10400( mov ebx,dword ptr ds:[<&MSVBVM60.__vbaVarDup>]  msvbvm60.__vbaVarDup
```

修改 ZF 标志位，继续运行程序，程序提示成功



可以看出，刚才的比较处是程序的关键。

再次运行到断点处，查看函数的参数

```
0012F410  0012F518  var28 = 0012F518
0012F414  0012F5DC  var18 = 0012F5DC
```

```
00402B58  46 00 46 00  46 00 46 00  00 00 00 00  44 00 00 00  F.F.F.F.....D...
01A948A4  30 00 00 00  33 00 34 00  35 00 00 00  05 00 03 00  0...3.4.5...¥...
```

可以看到，程序有两个参数。一个参数是 0，还有一个参数是 FFFF。

继续观察代码，发现上面有一个赋值的代码

```
00404E11  .  C785 10FFFFFF mov dword ptr ss:[ebp-0xF0],BJCM30A.00402B58   FFFF
```

```
00402B58=BJCM30A.00402B58 (UNICODE "FFFF")
堆栈 ss:[0012F520]=00000001
```

正是这行代码，将参数2的值变为了 FFFF。

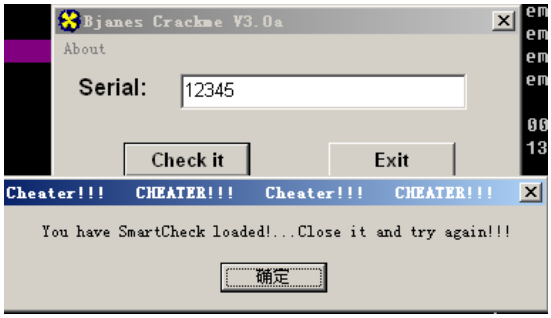
我们大胆猜测，FFFF 是正确的标志。

将参数1的值也修改为 FFFF，程序提示正确。

来到代码头部，下断点，点击 check 按钮，程序就断了下来

```
00404230  >  55           push ebp                                  检查按钮
00404231  .  8BEC         mov ebp,esp
00404233  .  83EC 0C      sub esp,0xC
00404236  .  68 66114000  push <jmp.&MSVBVM60.__vbaExceptHandler>   SE 处理程序安装
0040423B  .  64:A1 000000 mov eax,dword ptr fs:[0]
00404241  .  50           push eax                                  BJCM30A.00402007
00404242  .  64:8925 0000 mov dword ptr fs:[0],esp
00404249  .  81EC D801000 sub esp,0x1D8
0040424F  .  53           push ebx
00404250  .  56           push esi
00404251  .  57           push edi
```

向下调试代码，发现程序居然出现了一个新的提示



再查看上面的代码，程序先使用了循环浪费时间，然后使用循环结束的时间减去循环开始的时间，如果时间过大，就判断程序正在被调试，就会弹窗提示

```
00404320  .  FF15 9410400 call dword ptr ds:[<&MSVBVM60.#rtcGetTimer_535>]  msvbvm60.rtcGetTimer
00404326  .  FF15 D010400 call dword ptr ds:[<&MSVBVM60.__vbaFpI4>]         msvbvm60.__vbaFpI4
```

```
00404461  >  FF15 9410400 call dword ptr ds:[<&MSVBVM60.#rtcGetTimer_535>]  msvbvm60.rtcGetTimer
00404467  .  FF15 D010400 call dword ptr ds:[<&MSVBVM60.__vbaFpI4>]         msvbvm60.__vbaFpI4
0040446D  .  2B45 A4      sub eax,dword ptr ss:[ebp-0x5C]                   减去第一次的时间
00404470  .v 0F80 340C000 jo BJCM30A.004050AA
00404476  .  83F8 05      cmp eax,0x5                                       大于 5 就不跳转
00404479  .v 0F8E AD00000 jle BJCM30A.0040452C
```

然后程序进行长度检测，如果长度小于 5，就会直接提示错误

```
0040456D  >  8B95 7CFFFFF mov edx,dword ptr ss:[ebp-0x84]
00404573  .  52           push edx                                          String = 0000002F ???
00404574  .  FF15 1410400 call dword ptr ds:[<&MSVBVM60.__vbaLenBstr>]      __vbaLenBstr
0040457A  .  33DB         xor ebx,ebx
0040457C  .  83F8 05      cmp eax,0x5                                       比较长度是否小于5
0040457F  .  0F9CC3       setl bl
00404582  .  8D8D 7CFFFFF lea ecx,dword ptr ss:[ebp-0x84]
00404588  .  F7DB         neg ebx
0040458A  .  FF15 F010400 call dword ptr ds:[<&MSVBVM60.__vbaFreeStr>]      msvbvm60.__vbaFreeStr
00404590  .  8D8D 5CFFFFF lea ecx,dword ptr ss:[ebp-0xA4]
00404596  .  FF15 F410400 call dword ptr ds:[<&MSVBVM60.__vbaFreeObj>]      msvbvm60.__vbaFreeObj
0040459C  .  66:3BDF      cmp bx,di                                         长度小于5跳转
0040459F  .v 0F85 3909000 jnz BJCM30A.00404EDE
```

接下来程序会根据序列号的长度建立一个循环，循环的次数为序列号的长度

```
00404610  .  FF15 3010400 call dword ptr ds:[<&MSVBVM60.__vbaHresultCheckO  msvbvm60.__vbaHresultCheckObj
00404616  >  8B8D 7CFFFFF mov ecx,dword ptr ss:[ebp-0x84]
0040461C  .  51           push ecx                                          String = 00000020 ???
0040461D  .  FF15 1410400 call dword ptr ds:[<&MSVBVM60.__vbaLenBstr>]      __vbaLenBstr
00404623  .  8985 00FFFFF mov dword ptr ss:[ebp-0x100],eax
00404629  .  8D95 08FFFFF lea edx,dword ptr ss:[ebp-0xF8]
0040462F  .  8D85 F8FEFFF lea eax,dword ptr ss:[ebp-0x108]
00404635  .  52           push edx                                          Step8 = 0000002F
00404636  .  8D8D E8FEFFF lea ecx,dword ptr ss:[ebp-0x118]
0040463C  .  50           push eax                                          End8 = NULL
0040463D  .  8D95 64FEFFF lea edx,dword ptr ss:[ebp-0x19C]
00404643  .  51           push ecx                                          Start8 = 00000020
00404644  .  8D85 74FEFFF lea eax,dword ptr ss:[ebp-0x18C]
0040464A  .  52           push edx                                          TMPend8 = 0000002F
0040464B  .  8D4D 94      lea ecx,dword ptr ss:[ebp-0x6C]
0040464E  .  50           push eax                                          TMPstep8 = NULL
0040464F  .  51           push ecx                                          Counter8 = 00000020
00404650  .  C785 F8FEFFF mov dword ptr ss:[ebp-0x108],0x3
0040465A  .  C785 F0FEFFF mov dword ptr ss:[ebp-0x110],0x1
00404664  .  C785 E8FEFFF mov dword ptr ss:[ebp-0x118],0x2
0040466E  .  FF15 3810400 call dword ptr ds:[<&MSVBVM60.__vbaVarForInit>]   __vbaVarForInit
00404674  .  8D8D 7CFFFFF lea ecx,dword ptr ss:[ebp-0x84]
0040467A  .  8985 30FEFFF mov dword ptr ss:[ebp-0x1D0],eax
00404680  .  FF15 F010400 call dword ptr ds:[<&MSVBVM60.__vbaFreeStr>]     msvbvm60.__vbaFreeStr
00404686  .  8D8D 5CFFFFF lea ecx,dword ptr ss:[ebp-0xA4]
0040468C  .  FF15 F410400 call dword ptr ds:[<&MSVBVM60.__vbaFreeObj>]     msvbvm60.__vbaFreeObj
00404692  .  8B1D DC10400 mov ebx,dword ptr ds:[<&MSVBVM60.__vbaStrMove>]  msvbvm60.__vbaStrMove
00404698  >  39BD 30FEFFF cmp dword ptr ss:[ebp-0x1D0],edi
```

在循环中，程序依次比较每个字符是否和后一个字符是否相同

```
00404779  .  8B8D 7CFFFFF mov ecx,dword ptr ss:[ebp-0x84]
0040477F  .  8B3D 5410400 mov edi,dword ptr ds:[<&MSVBVM60.#rtcMidCharBstr  msvbvm60.rtcMidCharBstr
00404785  .  50           push eax
00404786  .  51           push ecx
00404787  .  FFD7         call edi                                          msvbvm60.rtcMidCharBstr; <&MSVBVM60.#rtcMidCharBstr_63
00404789  .  8BD0         mov edx,eax                                       取 key[n], n 是循环的次数
0040478B  .  8D8D 74FFFFF lea ecx,dword ptr ss:[ebp-0x8C]
00404791  .  FFD3         call ebx                                          msvbvm60.__vbaStrMove
00404793  .  50           push eax
00404794  .  8D95 28FFFFF lea edx,dword ptr ss:[ebp-0xD8]
0040479A  .  8D45 94      lea eax,dword ptr ss:[ebp-0x6C]
0040479D  .  52           push edx
0040479E  .  8D8D F8FEFFF lea ecx,dword ptr ss:[ebp-0x108]
004047A4  .  50           push eax                                          var18 = 00182214
004047A5  .  8D95 38FFFFF lea edx,dword ptr ss:[ebp-0xC8]
004047AB  .  51           push ecx                                          var28 = 0012F580
004047AC  .  52           push edx                                          saveto8 = 00182214
004047AD  .  FF15 C810400 call dword ptr ds:[<&MSVBVM60.__vbaVarAdd>]       __vbaVarAdd
004047B3  .  50           push eax                                          n = n + 1
004047B4  .  FF15 C410400 call dword ptr ds:[<&MSVBVM60.__vbaI4Var>]        msvbvm60.__vbaI4Var
004047BA  .  50           push eax
004047BB  .  8B85 78FFFFF mov eax,dword ptr ss:[ebp-0x88]
004047C1  .  50           push eax
004047C2  .  FFD7         call edi                                          msvbvm60.rtcMidCharBstr
004047C4  .  8BD0         mov edx,eax                                       取下一个个字符,也就是 key[n+1]
004047C6  .  8D8D 70FFFFF lea ecx,dword ptr ss:[ebp-0x90]
004047CC  .  FFD3         call ebx                                          msvbvm60.__vbaStrMove
004047CE  .  50           push eax
004047CF  .  FF15 6810400 call dword ptr ds:[<&MSVBVM60.__vbaStrCmp>]       msvbvm60.__vbaStrCmp
004047D5  .  8BF8         mov edi,eax                                       比较 key[n]和key[n+1]是否相同
004047D7  .  8D8D 70FFFFF lea ecx,dword ptr ss:[ebp-0x90]
```

如果字符和后一个字符不相同，就进入下次循环；如果相同，就会在内存 0012F558 处加1

```
00404838  .  66:85FF      test di,di
0040483B  .  74 37        je short BJCM30A.00404874                         不相等跳转
0040483D  .  8D4D B8      lea ecx,dword ptr ss:[ebp-0x48]
00404840  .  8D95 08FFFFF lea edx,dword ptr ss:[ebp-0xF8]
00404846  .  51           push ecx                                          var18 = 00D1FC9C
00404847  .  8D85 48FFFFF lea eax,dword ptr ss:[ebp-0xB8]
0040484D  .  52           push edx                                          var28 = NULL
0040484E  .  50           push eax                                          saveto8 = NULL
0040484F  .  C785 10FFFFF mov dword ptr ss:[ebp-0xF0],0x1
00404859  .  C785 08FFFFF mov dword ptr ss:[ebp-0xF8],0x2
00404863  .  FF15 C810400 call dword ptr ds:[<&MSVBVM60.__vbaVarAdd>]       __vbaVarAdd
00404869  .  8BD0         mov edx,eax                                       次数加1
0040486B  .  8D4D B8      lea ecx,dword ptr ss:[ebp-0x48]
0040486E  .  FF15 0810400 call dword ptr ds:[<&MSVBVM60.__vbaVarMove>]      msvbvm60.__vbaVarMove
00404874  >  8D8D 64FEFFF lea ecx,dword ptr ss:[ebp-0x19C]
0040487A  .  8D95 74FEFFF lea edx,dword ptr ss:[ebp-0x18C]
00404880  .  51           push ecx                                          TMPend8 = 00D1FC9C
00404881  .  8D45 94      lea eax,dword ptr ss:[ebp-0x6C]
00404884  .  52           push edx                                          TMPstep8 = NULL
00404885  .  50           push eax                                          Counter8 = NULL
00404886  .  FF15 E810400 call dword ptr ds:[<&MSVBVM60.__vbaVarForNext>]   __vbaVarForNext
0040488C  .  8985 30FEFFF mov dword ptr ss:[ebp-0x1D0],eax
00404892  .  33FF         xor edi,edi
00404894  .^ E9 FFFDFFFF  jmp BJCM30A.00404698
```

然后在后面和序列号长度减去1作比较，如果相同，则跳转到错误提示处

```
004048E4  >  8B95 7CFFFFF mov edx,dword ptr ss:[ebp-0x84]
004048EA  .  52           push edx                                  ┌String = "12345"
004048EB  .  FF15 1410400 call dword ptr ds:[<&MSVBVM60.__vbaLenBstr>]  └__vbaLenBstr
004048F1  .  83E8 01      sub eax,0x1                               len - 1
004048F4  .  8D8D 08FFFFF lea ecx,dword ptr ss:[ebp-0xF8]
004048FA  .  0F80 AA07000 jo BJCM30A.00405 0AA
00404900  .  8985 10FFFFF mov dword ptr ss:[ebp-0xF0],eax
00404906  .  8D45 B8      lea eax,dword ptr ss:[ebp-0x48]
00404909  .  50           push eax                                  ┌var18 = 0012F5C8
0040490A  .  51           push ecx                                  │var28 = 0012F518
0040490B  .  C785 08FFFFF mov dword ptr ss:[ebp-0xF8],0x8003        │
00404915  .  FF15 6C10400 call dword ptr ds:[<&MSVBVM60.__vbaVarTstEq>]  └__vbaVarTstEq
0040491B  .  8D8D 7CFFFFF lea ecx,dword ptr ss:[ebp-0x84]
00404921  .  66:8985 CCFE mov word ptr ss:[ebp-0x134],ax
00404928  .  FF15 F010400 call dword ptr ds:[<&MSVBVM60.__vbaFreeStr>]  msvbvm60.__vbaFreeStr
0040492E  .  8D8D 5CFFFFF lea ecx,dword ptr ss:[ebp-0xA4]
00404934  .  FF15 F410400 call dword ptr ds:[<&MSVBVM60.__vbaFreeObj>]  msvbvm60.__vbaFreeObj
0040493A  .  66:39BD CCFE cmp word ptr ss:[ebp-0x134],di
00404941  .  0F85 9705000 jnz BJCM30A.00404EDE
```

也就说序列号不可以全部为同一个值，比如说都是 1。

接下来，又是一个循环

```
004049A6  >  8B95 7CFFFFF mov edx,dword ptr ss:[ebp-0x84]
004049AC  .  52           push edx                                  ┌String = NULL
004049AD  .  FF15 1410400 call dword ptr ds:[<&MSVBVM60.__vbaLenBstr>]  └__vbaLenBstr
004049B3  .  8985 00FFFFF mov dword ptr ss:[ebp-0x100],eax
004049B9  .  8D85 08FFFFF lea eax,dword ptr ss:[ebp-0xF8]
004049BF  .  8D8D F8FEFFF lea ecx,dword ptr ss:[ebp-0x108]
004049C5  .  50           push eax                                  ┌Step8 = NULL
004049C6  .  8D95 E8FEFFF lea edx,dword ptr ss:[ebp-0x118]          │
004049CC  .  51           push ecx                                  │End8 = 00D1FC9C
004049CD  .  8D85 44FEFFF lea eax,dword ptr ss:[ebp-0x1BC]          │
004049D3  .  52           push edx                                  │Start8 = NULL
004049D4  .  8D8D 54FEFFF lea ecx,dword ptr ss:[ebp-0x1AC]          │
004049DA  .  50           push eax                                  │TMPend8 = NULL
004049DB  .  8D55 94      lea edx,dword ptr ss:[ebp-0x6C]           │
004049DE  .  51           push ecx                                  │TMPstep8 = 00D1FC9C
004049DF  .  52           push edx                                  │Counter8 = NULL
004049E0  .  C785 F8FEFFF mov dword ptr ss:[ebp-0x108],0x3          │
004049EA  .  C785 F0FEFFF mov dword ptr ss:[ebp-0x110],0x1          │
004049F4  .  C785 E8FEFFF mov dword ptr ss:[ebp-0x118],0x2          │
004049FE  .  FF15 3810400 call dword ptr ds:[<&MSVBVM60.__vbaVarForInit>]  └__vbaVarForInit
00404A04  .  8D8D 7CFFFFF lea ecx,dword ptr ss:[ebp-0x84]
00404A0A  .  8985 2CFEFFF mov dword ptr ss:[ebp-0x1D4],eax
00404A10  .  FF15 F010400 call dword ptr ds:[<&MSVBVM60.__vbaFreeStr>]  msvbvm60.__vbaFreeStr
00404A16  .  8D8D 5CFFFFF lea ecx,dword ptr ss:[ebp-0xA4]
00404A1C  .  FF15 F410400 call dword ptr ds:[<&MSVBVM60.__vbaFreeObj>]  msvbvm60.__vbaFreeObj
00404A22  >  39BD 2CFEFFF cmp dword ptr ss:[ebp-0x1D4],edi
00404A28  .  0F84 1D03000 je BJCM30A.00404D4B
```

程序求得序列号的长度，然后转化为字符串

```
00404A73  .  FF15 3010400 call dword ptr ds:[<&MSVBVM60.__vbaHresultCheckO  msvbvm60.__vbaHresultCheckObj
00404A79  >  8B85 7CFFFFF mov eax,dword ptr ss:[ebp-0x84]
00404A7F  .  50           push eax                                  ┌String = "5"
00404A80  .  FF15 1410400 call dword ptr ds:[<&MSVBVM60.__vbaLenBstr>]  └__vbaLenBstr
00404A86  .  8D8D 48FFFFF lea ecx,dword ptr ss:[ebp-0xB8]
00404A8C  .  8985 50FFFFF mov dword ptr ss:[ebp-0xB0],eax
00404A92  .  51           push ecx
00404A93  .  C785 48FFFFF mov dword ptr ss:[ebp-0xB8],0x3
00404A9D  .  FF15 A810400 call dword ptr ds:[<&MSVBVM60.#rtcHexBstrFromVar  msvbvm60.rtcHexBstrFromVar
00404AA3  .  8BD0         mov edx,eax                               转化为字符串
```

然后得到序列号最左边的字符，最后也转化为字符串

```
00404AD0  .  6A 01        push 0x1
00404AD2  .  8D95 28FFFFF lea edx,dword ptr ss:[ebp-0xD8]
00404AD8  .  51           push ecx
00404AD9  .  52           push edx
00404ADA  .  89BD 58FFFFF mov dword ptr ss:[ebp-0xA8],edi
00404AE0  .  8985 40FFFFF mov dword ptr ss:[ebp-0xC0],eax
00404AE6  .  C785 38FFFFF mov dword ptr ss:[ebp-0xC8],0x9
00404AF0  .  FF15 D410400 call dword ptr ds:[<&MSVBVM60.#rtcLeftCharVar_61  msvbvm60.rtcLeftCharVar
00404AF6  .  8D85 28FFFFF lea eax,dword ptr ss:[ebp-0xD8]
00404AFC  .  8D8D 78FFFFF lea ecx,dword ptr ss:[ebp-0x88]
00404B02  .  50           push eax                                  ┌String8 = 01A9AE5C
00404B03  .  51           push ecx                                  │ARG2 = 001A0000
00404B04  .  FF15 9010400 call dword ptr ds:[<&MSVBVM60.__vbaStrVarVal>]  │__vbaStrVarVal
00404B0A  .  50           push eax                                  │String = "3"
00404B0B  .  FF15 2810400 call dword ptr ds:[<&MSVBVM60.#rtcAnsiValueBstr_  └rtcAnsiValueBstr
00404B11  .  8D95 18FFFFF lea edx,dword ptr ss:[ebp-0xE8]
00404B17  .  66:8985 20FF mov word ptr ss:[ebp-0xE0],ax
00404B1E  .  52           push edx
00404B1F  .  C785 18FFFFF mov dword ptr ss:[ebp-0xE8],0x2
00404B29  .  FF15 A810400 call dword ptr ds:[<&MSVBVM60.#rtcHexBstrFromVar  msvbvm60.rtcHexBstrFromVar
00404B2F  .  8BD0         mov edx,eax                               转化为字符串
```

最后调用函数，将两个数相乘，也就是将 序列号的长度和序列号第一个字符的 ASCII 值相乘

```
00404B72  .  8B06          mov eax,dword ptr ds:[esi]                  BJCM30A.00406A74
00404B74  .  8D8D 68FFFFFF lea ecx,dword ptr ss:[ebp-0x98]
00404B7A  .  8D95 6CFFFFFF lea edx,dword ptr ss:[ebp-0x94]
00404B80  .  51            push ecx
00404B81  .  52            push edx
00404B82  .  8D8D 70FFFFFF lea ecx,dword ptr ss:[ebp-0x90]
00404B88  .  8D95 74FFFFFF lea edx,dword ptr ss:[ebp-0x8C]
00404B8E  .  51            push ecx
00404B8F  .  52            push edx
00404B90  .  56            push esi
00404B91  .  FF90 F8060000 call dword ptr ds:[eax+0x6F8]               相乘
```

接下里，程序依次将序列号相加

```
00404C9E  .  FF15 C4104000 call dword ptr ds:[<&MSVBVM60.__vbaI4Var>]      msvbvm60.__vbaI4Var
00404CA4  .  8B8D 7CFFFFFF mov ecx,dword ptr ss:[ebp-0x84]
00404CAA  .  50            push eax
00404CAB  .  51            push ecx
00404CAC  .  FF15 54104000 call dword ptr ds:[<&MSVBVM60.#rtcMidCharBstr_63  msvbvm60.rtcMidCharBstr
00404CB2  .  8BD0          mov edx,eax
00404CB4  .  8D8D 78FFFFFF lea ecx,dword ptr ss:[ebp-0x88]
00404CBA  .  FFD3          call ebx                                        msvbvm60.__vbaStrMove
00404CBC  .  50            push eax                                        String = "一"
00404CBD  .  FF15 28104000 call dword ptr ds:[<&MSVBVM60.#rtcAnsiValueBstr_ rtcAnsiValueBstr
00404CC3  .  66:8985 00FFF mov word ptr ss:[ebp-0x100],ax
00404CCA  .  8D55 CC       lea edx,dword ptr ss:[ebp-0x34]
00404CCD  .  8D85 F8FEFFFF lea eax,dword ptr ss:[ebp-0x108]
00404CD3  .  52            push edx                                        var18 = 0012F5DC
00404CD4  .  8D8D 38FFFFFF lea ecx,dword ptr ss:[ebp-0xC8]
00404CDA  .  50            push eax                                        var28 = 0012F508
00404CDB  .  51            push ecx                                        saveto8 = 0012F548
00404CDC  .  C785 F8FEFFFF mov dword ptr ss:[ebp-0x108],0x2
00404CE6  .  FF15 C8104000 call dword ptr ds:[<&MSVBVM60.__vbaVarAdd>]     __vbaVarAdd
```

循环结束，程序将相加的结果和相乘的结果进行比较。如果相等，则返回 FFFF；不想等在返回 0。

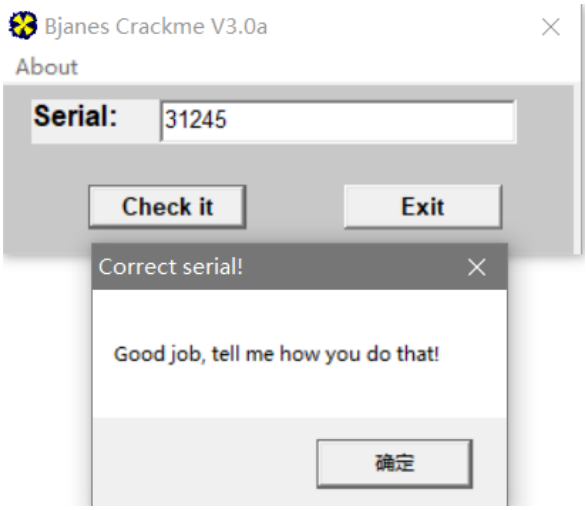
在最后再和 FFFF 进行比较，相同说明正确。

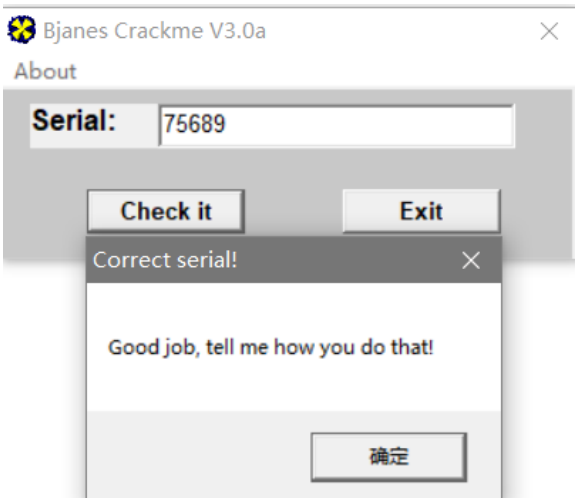
```
00404DA0  .  FF92 F8060000 call dword ptr ds:[edx+0x6F8]                进行比较
00404DA6  .  3BC7          cmp eax,edi
00404DA8  .v 7D 12         jge short BJCM30A.00404DBC
00404DAA  .  68 F8060000   push 0x6F8
00404DAF  .  68 B4274000   push BJCM30A.004027B4
00404DB4  .  56            push esi
00404DB5  .  50            push eax
00404DB6  .  FF15 30104000 call dword ptr ds:[<&MSVBVM60.__vbaHresultCheckO msvbvm60.__vbaHresultCheckObj
00404DBC  > 8B85 74FFFFFF mov eax,dword ptr ss:[ebp-0x8C]
00404DC2  .  BE 08000000   mov esi,0x8
00404DC7  .  8D95 48FFFFFF lea edx,dword ptr ss:[ebp-0xB8]
00404DCD  .  8D4D CC       lea ecx,dword ptr ss:[ebp-0x34]
00404DD0  .  89BD 74FFFFFF mov dword ptr ss:[ebp-0x8C],edi
00404DD6  .  8985 50FFFFFF mov dword ptr ss:[ebp-0xB0],eax
00404DDC  .  89B5 48FFFFFF mov dword ptr ss:[ebp-0xB8],esi
00404DE2  .  FF15 08104000 call dword ptr ds:[<&MSVBVM60.__vbaVarMove>]    msvbvm60.__vbaVarMove
00404DE8  .  8D95 70FFFFFF lea edx,dword ptr ss:[ebp-0x90]
00404DEE  .  8D85 78FFFFFF lea eax,dword ptr ss:[ebp-0x88]
00404DF4  .  52            push edx
00404DF5  .  8D8D 7CFFFFFF lea ecx,dword ptr ss:[ebp-0x84]
00404DFB  .  50            push eax
00404DFC  .  51            push ecx
00404DFD  .  6A 03         push 0x3
00404DFF  .  FF15 B4104000 call dword ptr ds:[<&MSVBVM60.__vbaFreeStrList>] msvbvm60.__vbaFreeStrList
00404E05  .  83C4 10       add esp,0x10
00404E08  .  8D55 CC       lea edx,dword ptr ss:[ebp-0x34]
00404E0B  .  8D85 08FFFFFF lea eax,dword ptr ss:[ebp-0xF8]
00404E11  .  C785 10FFFFFF mov dword ptr ss:[ebp-0xF0],BJCM30A.00402B58     FFFF
00404E1B  .  52            push edx                                        var18 = 0012F5DC
00404E1C  .  50            push eax                                        var28 = 0012F518
00404E1D  .  C785 08FFFFFF mov dword ptr ss:[ebp-0xF8],0x8008
00404E27  .  FF15 6C104000 call dword ptr ds:[<&MSVBVM60.__vbaVarTstEq>]   __vbaVarTstEq
```

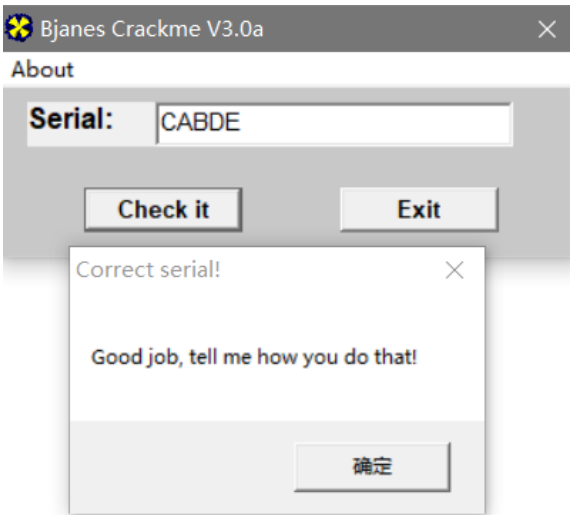
4.注册机

1. 不能少于5位
2. 不能全部相同
3. 序列号首位和长度的乘积要等于序列号各位相加的和

本来是想写注册机的，但是我发现了一个有趣的规律，所以就不用写注册机了







当序列号是五位数的时候。序列号首位随便填一个值，然后在 ASCII 码表上找到该值相邻的上面2个值和下面2个值，就是一个可以使用的序列号。

相关文件在我的 Github：https://github.com/UnreachableLove/160-Crackme/tree/master/Crackme017

2019-09-18 21:00:21

分类: 160 Crackme

好文要顶 关注我 收藏该文

随风而逝的白色相簿
关注 - 4
粉丝 - 1

« 上一篇： Crackme015

posted @ 2019-09-18 21:00 随风而逝的白色相簿 阅读(2) 评论(0) 编辑 收藏

刷新评论 刷新页面 返回顶部

发表评论

昵称： 随风而逝的白色相簿

评论内容：

提交评论 退出

[Ctrl+Enter快捷键提交]

0
推荐