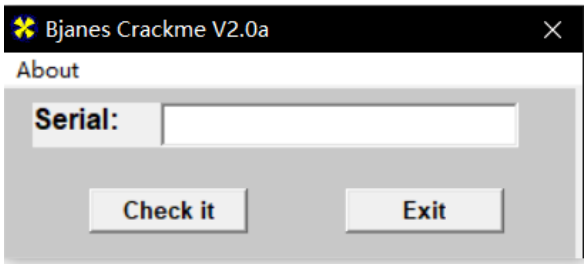
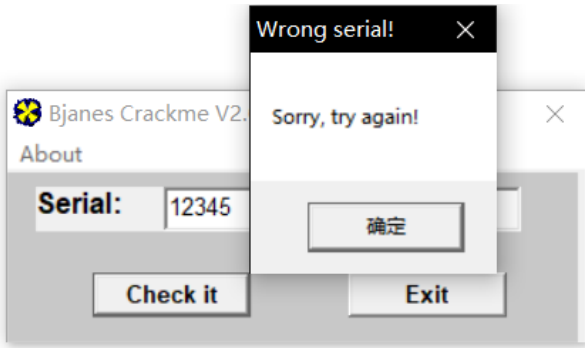
随风而逝的白色相簿

Crackme014

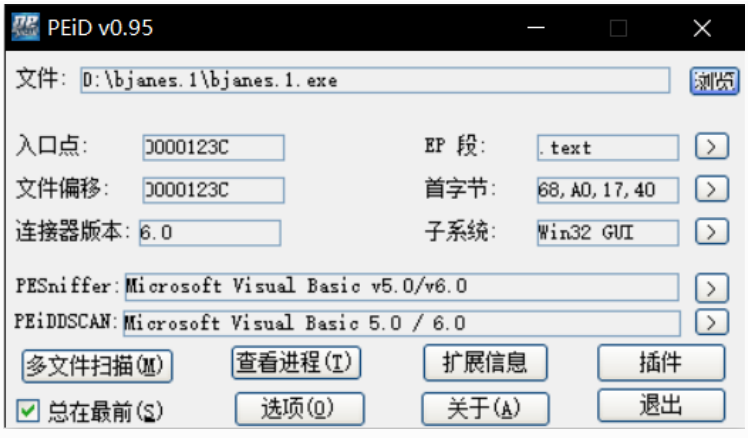
Crackme014 的逆向分析

1.程序观察





2.简单查壳



无壳，使用 VB 编写。

3.程序分析

使用 OD 载入程序，搜索字符串

```
0040329E  ascii "DD@D@DD@",0
004032F0  dd bjanes_1.00400074          ASCII ".\r\r\n$"
0040330F  ascii "7D",0
00403329  dd bjanes_1.00404074          UNICODE "2"
0040333E  ascii "D@D@",0
00403348  ascii "DD",0
0040334B  ascii "DDD",0
0040334F  ascii "D@",0
0040335B  ascii "DDDD"
00403725  mov ecx,dword ptr ds:[esi]    (初始 CPU 选择)
00403A4E  mov dword ptr ss:[ebp-0xA8],bjanes_1.00  UNICODE "Wrong serial!"
00403A69  mov dword ptr ss:[ebp-0x98],bjanes_1.00  UNICODE "Sorry, try again!"
00403ACE  mov dword ptr ss:[ebp-0xA8],bjanes_1.00  UNICODE "Correct serial!"
00403AE9  mov dword ptr ss:[ebp-0x98],bjanes_1.00  UNICODE "Good job, tell me how you do that!"
00403C90  push bjanes_1.0040533C        ASCII "T崔"
```

可以看到先前报错时，所提示的语句。

双击跟进程序。

在错误的字符串上方，有一个循环。

```
0040377C  > 66:8B8D 14FF  mov cx,word ptr ss:[ebp-0xEC]
00403783  . 66:394D E8    cmp word ptr ss:[ebp-0x18],cx
00403787  . 0F8F 1703000  jg bjanes_1.00403AA4                跳转到正确
0040378D  . 8B17          mov edx,dword ptr ds:[edi]          bjanes_1.00405A74
0040378F  . 57            push edi
00403790  . FF92 0803000  call dword ptr ds:[edx+0x308]
00403796  . 50            push eax
00403797  . 8D45 D4       lea eax,dword ptr ss:[ebp-0x2C]
0040379A  . 50            push eax
0040379B  . FF15 2C10400  call dword ptr ds:[<&MSVBVM60.__vbaObjS  msvbvm60.__vbaObjSet
004037A1  . 8BD8          mov ebx,eax
004037A3  . 8D55 E4       lea edx,dword ptr ss:[ebp-0x1C]
004037A6  . 52            push edx
004037A7  . 53            push ebx
004037A8  . 8B0B          mov ecx,dword ptr ds:[ebx]
004037AA  . FF91 A000000  call dword ptr ds:[ecx+0xA0]
004037B0  . 85C0          test eax,eax
004037B2  . DBE2          fclex
004037B4  . 7D 12         jge short bjanes_1.004037C8
004037B6  . 68 A0000000   push 0xA0
004037BB  . 68 44224000   push bjanes_1.00402244
004037C0  . 53            push ebx
004037C1  . 50            push eax
004037C2  . FF15 2410400  call dword ptr ds:[<&MSVBVM60.__vbaHres  msvbvm60.__vbaHresultCheckObj
004037C8  > 8B07          mov eax,dword ptr ds:[edi]          bjanes_1.00405A74
004037CA  . 57            push edi
004037CB  . FF90 0803000  call dword ptr ds:[eax+0x308]
004037D1  . 8D4D D0       lea ecx,dword ptr ss:[ebp-0x30]
004037D4  . 50            push eax
004037D5  . 51            push ecx                            ntdll.7C93003D
004037D6  . FF15 2C10400  call dword ptr ds:[<&MSVBVM60.__vbaObjS  msvbvm60.__vbaObjSet
004037DC  . 8BF8          mov edi,eax
004037DE  . 8D45 DC       lea eax,dword ptr ss:[ebp-0x24]
004037E1  . 50            push eax
```

循环的最开始，会进行一个比较。若是比较结果正确，会跳转到正确的提示代码处。

在循环的上方，还有一处关键的代码。

```
004036DC  . 50            push eax                            String = NULL
004036DD  . FF15 0810400  call dword ptr ds:[<&MSVBVM60.__vbaLenBs  __vbaLenBstr
004036E3  . 33C9          xor ecx,ecx                         清空ecx
004036E5  . 83F8 09       cmp eax,0x9                         比较长度是否为9
004036E8  . 0F95C1        setne cl                            cl 根据比较结果取值
004036EB  . F7D9          neg ecx                             相反数
004036ED  . 8BF1          mov esi,ecx                         ntdll.7C93003D
004036EF  . 8D4D E4       lea ecx,dword ptr ss:[ebp-0x1C]
004036F2  . FF15 C010400  call dword ptr ds:[<&MSVBVM60.__vbaFree  msvbvm60.__vbaFreeStr
004036F8  . 8D4D D4       lea ecx,dword ptr ss:[ebp-0x2C]
004036FB  . FF15 C410400  call dword ptr ds:[<&MSVBVM60.__vbaFree  msvbvm60.__vbaFreeObj
00403701  . 66:3BF3       cmp si,bx
00403704  . 0F85 1A03000  jnz bjanes_1.00403A24               跳转到错误提示处
```

程序先得到所输入序列号的长度，如果不为9，则跳转到错误提示。

所以序列号的程度应该是9位。

然后程序建立循环，就是上面那一张图上的大循环。
循环次数是 9 ，也就是序列号的位数。

然后程序将循环的次数，与 0x2 做异或运算

```
004038EA  .  50            push eax
004038EB  .  FF15 2410400  call dword ptr ds:[<&MSVBVM60.__vbaHres  msvbvm60.__vbaHresultCheckObj
004038F1  >  66:8B45 E8    mov ax,word ptr ss:[ebp-0x18]           ax = 循环的次数 = 序列号的位数
004038F5  .  8B1D 7410400  mov ebx,dword ptr ds:[<&MSVBVM60.#rtcStr  msvbvm60.rtcStrFromVar
004038FB  .  66:35 0200    xor ax,0x2                              位数 xor 2
004038FF  .  8D4D A8       lea ecx,dword ptr ss:[ebp-0x60]
00403902  .  0F80 A402000  jo bjanes_1.00403BAC
00403908  .  51            push ecx
00403909  .  66:8945 A8    mov word ptr ss:[ebp-0x58],ax
0040390D  .  C745 A0 0200  mov dword ptr ss:[ebp-0x60],0x2
00403914  .  FFD3          call ebx                                msvbvm60.rtcStrFromVar; <&MSVBVM60.#rtcStrFromVar_53
```

循环的最后，程序将计算出来的值和我们输入的序列号作比较

```
00403990  .  C745 D8 0000  mov dword ptr ss:[ebp-0x28],0x0
00403997  .  8945 98       mov dword ptr ss:[ebp-0x68],eax
0040399A  .  8D45 80       lea eax,dword ptr ss:[ebp-0x80]
0040399D  .  50            push eax
0040399E  .  C745 90 0800  mov dword ptr ss:[ebp-0x70],0x8
004039A5  .  FF15 B010400  call dword ptr ds:[<&MSVBVM60.#rtcRightC  msvbvm60.rtcRightCharVar
004039AB  .  8D8D 30FFFFF  lea ecx,dword ptr ss:[ebp-0xD0]
004039B1  .  8D55 80       lea edx,dword ptr ss:[ebp-0x80]
004039B4  .  51            push ecx                                var18 = 0012F540
004039B5  .  52            push edx                                var28 = 0012F590
004039B6  .  FF15 A010400  call dword ptr ds:[<&MSVBVM60.__vbaVarT  __vbaVarTstNe
```

```
01A79EC4  33 00 00 00  00 00 00 00  22 03 00 00  03 00 03 00  3.......
```

因为程序是单个单个进行比较的，所以循环要进行 9 次，一次不符合就会跳转到错误提示处。

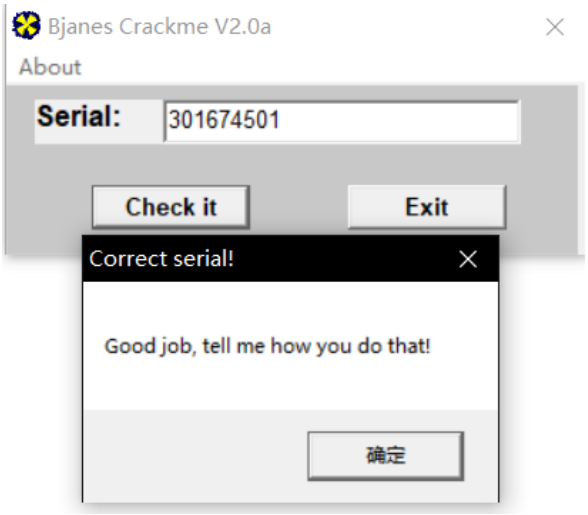
这样我们就可以通过计算得到正确的序列号

第一位：1 xor 2 = 3
第二位：2 xor 2 = 0
第三位：3 xor 2 = 1
第四位：4 xor 2 = 6
第五位：5 xor 2 = 7
第六位：6 xor 2 = 4
第七位：7 xor 2 = 5
第八位：8 xor 2 = 10
第九位：9 xor 2 = 11

这时候或许就有疑问了，最后两个计算出来的是两位数，怎么办呢？

```
0040398D  .  8B45 D8       mov eax,dword ptr ss:[ebp-0x28]
00403990  .  C745 D8 0000  mov dword ptr ss:[ebp-0x28],0x0
00403997  .  8945 98       mov dword ptr ss:[ebp-0x68],eax
0040399A  .  8D45 80       lea eax,dword ptr ss:[ebp-0x80]
0040399D  .  50            push eax
0040399E  .  C745 90 0800  mov dword ptr ss:[ebp-0x70],0x8
004039A5  .  FF15 B010400  call dword ptr ds:[<&MSVBVM60.#rtcRightC  msvbvm60.rtcRightCharVar
```

可以看到，程序计算出来值之后，是取所得值右边的一位，所以最后的序列号是：301674501。



相关文件在我的 Github：https://github.com/UnreachableLove/160-Crackme/tree/master/Crackme014

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: CrackMe011

posted @ 2019-09-16 21:26 随风而逝的白色相簿 阅读(0) 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]