【reversing.kr逆向之旅】Twist1的writeup

原创

iqiqiya 于 2019-01-19 16:24:54 发布 386 收藏
分类专栏： 我的逆向之路 我的CTF之路 ------reversing.kr 我的CTF进阶之路 文章标签： 【reversing.kr逆向之旅】Twist1的write Twist1的writeup 逆向题

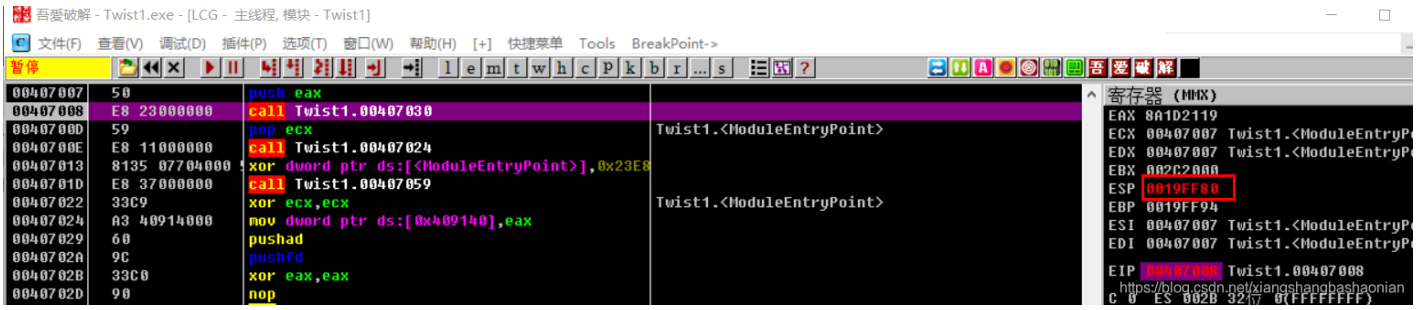
我的逆向之路 同时被 3 个专栏收录
108 篇文章 10 订阅
订阅专栏

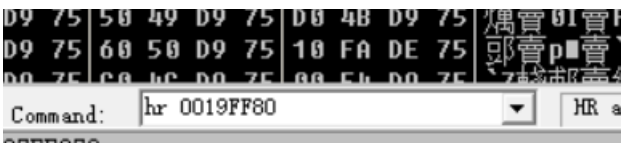
我的CTF之路
92 篇文章 5 订阅
订阅专栏

------reversing.kr
11 篇文章 0 订阅
订阅专栏

将程序载入OllyDbg

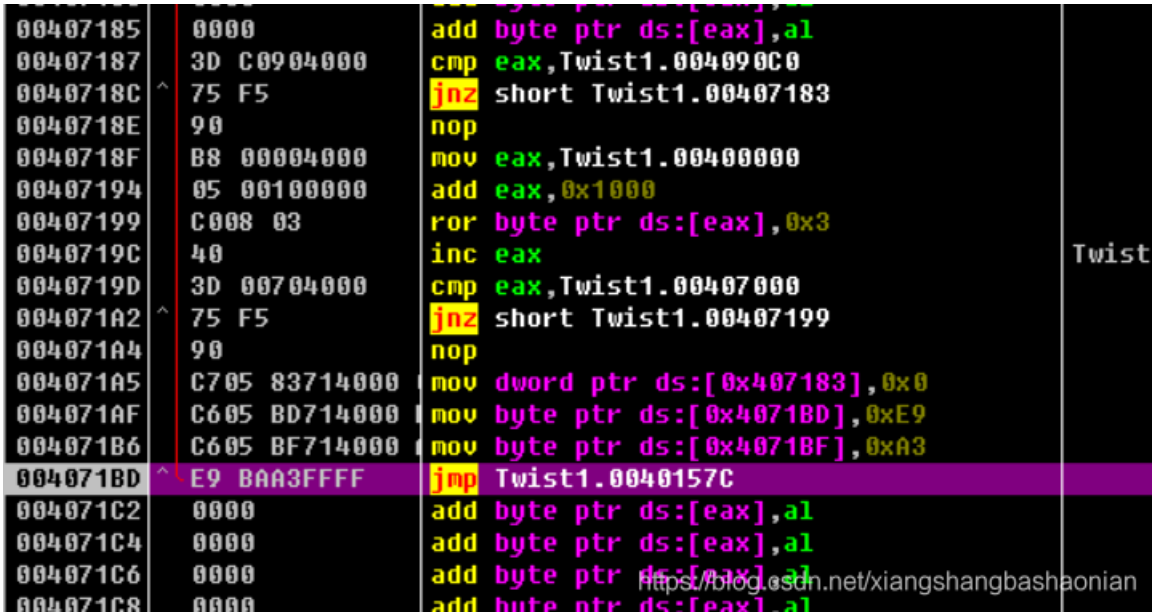
直接F8单步的话会直接停止运行　我们先可以单步一次使用ESP定律

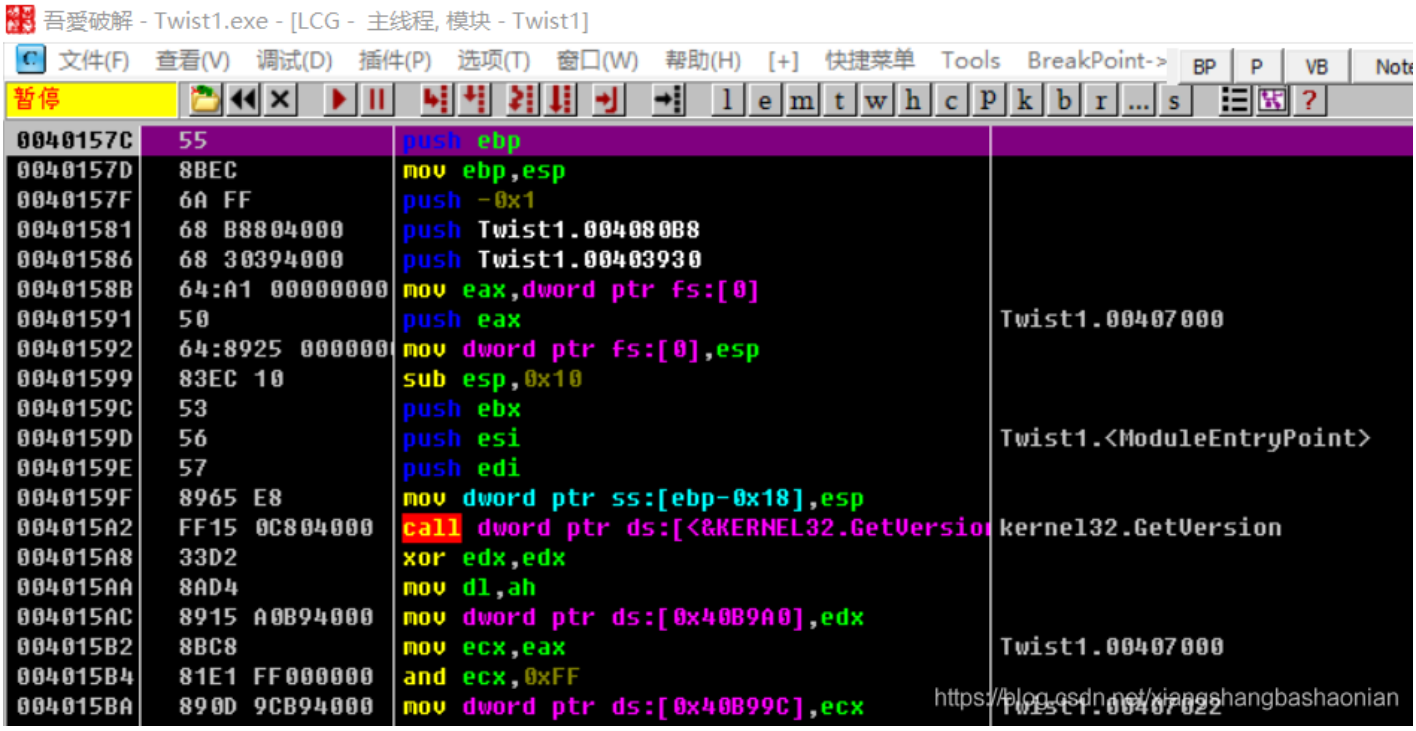




接着单步一步一步走就可以　遇到向上的跳转　在他的下一行按F4就可以走到这里

如下图所示　将要跳向OEP

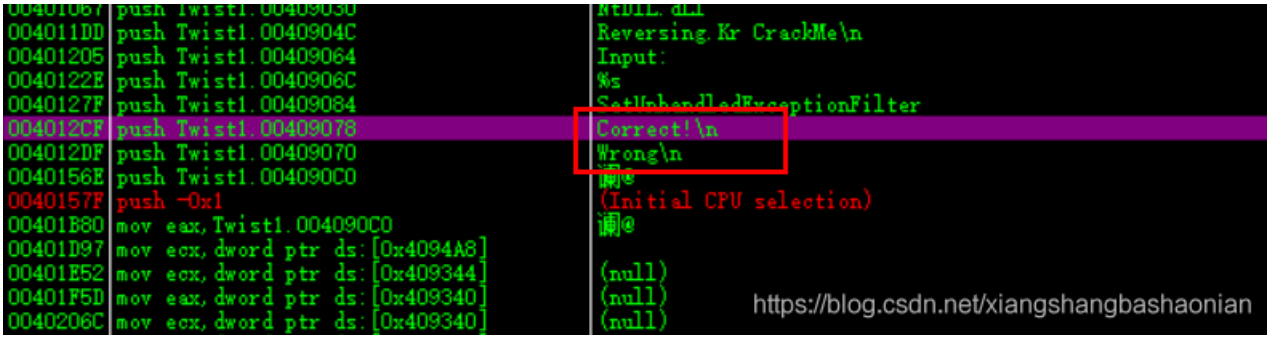




来到OEP后

直接右键OllyDump脱壳 两种方式都保存下　发现方式二可以成功运行

空白区域右键-->中文搜索引擎　我们可以看到关键字符串 那个je跳转貌似很不简单

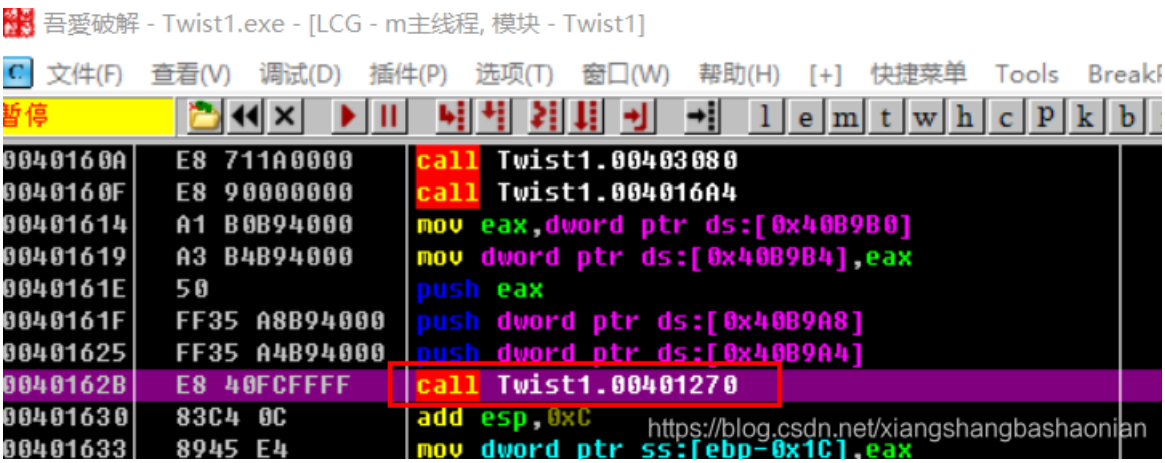


```
004012C5   90              nop
004012C6   E8 75FFFFFF     call Twist1.00401240
004012CB   85C0            test eax,eax                    Twist1.00407000
004012CD   74 10           je short Twist1.004012DF
004012CF   68 78904000     push Twist1.00409078            Correct!\n
004012D4   E8 5B020000     call Twist1.00401534
004012D9   83C4 04         add esp,0x4
004012DC   33C0            xor eax,eax                     Twist1.00407000
004012DE   C3              retn
004012DF   68 70904000     push Twist1.00409070            Wrong\n
004012E4   E8 4B020000     call Twist1.00401534
004012E9   83C4 04         add esp,0x4
```


我们先不用管

接着单步

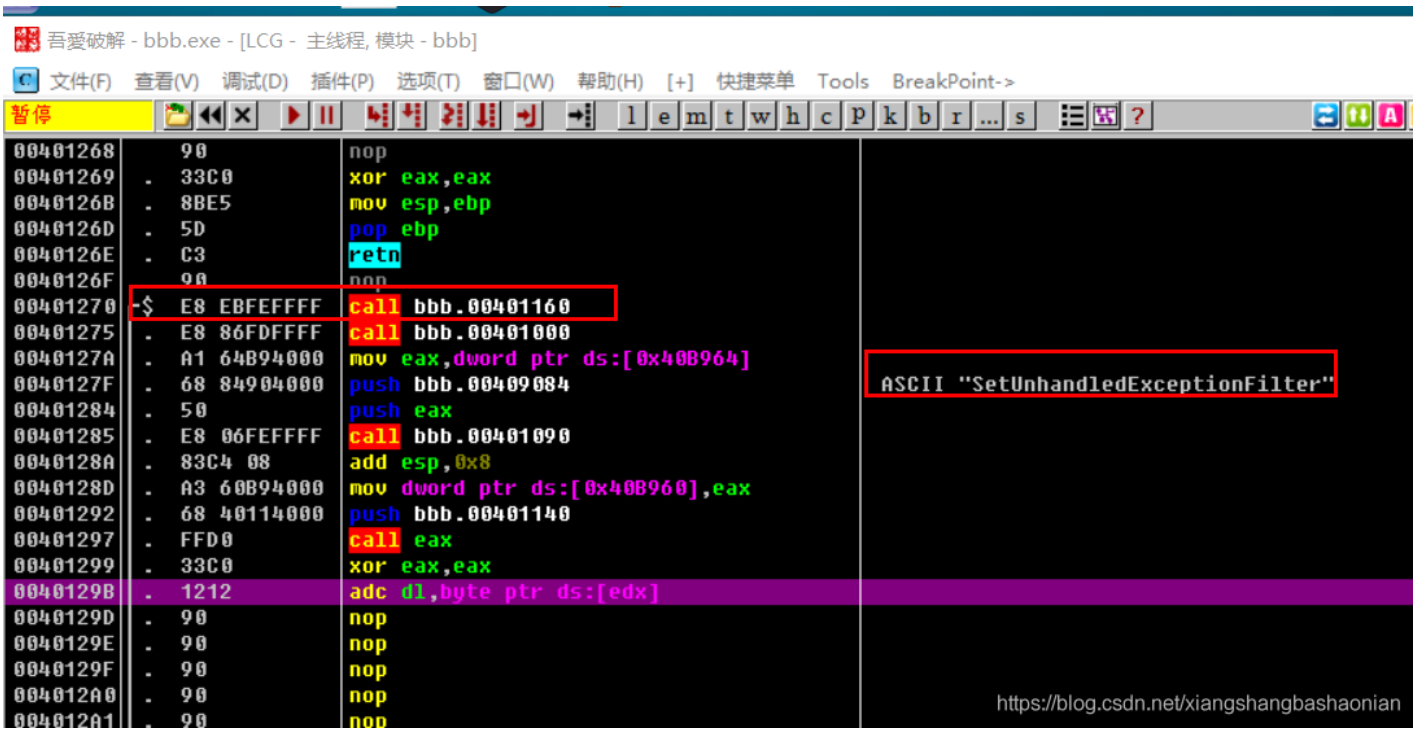
发现每次走到这里 F8的话 程序就会停止运行



重新载入 运行到这里的时候 我们右键跟随 将着只要一单步就崩溃

百度了一下SetUnhandledExceptionFilter

说是用来设置异常捕获函数 猜测可能就是这个的原因了



解决办法 我是直接在下边设置EIP来跳过这部分代码的执行 从而避免遇到异常

```
0040129B   .  1212          adc dl,byte ptr ds:[edx]
0040129D   .  90            nop
0040129E   .  90            nop
0040129F   .  90            nop
```

接着F8走过好几个nop 还有三个call 第三个call是用来获取我们的输入

输入iqiqiya回车 在单步几个nop 来到这里

```
004012C5   .  90            nop
004012C6   .  E8 75FFFFFF   call bbb.00401240                关键call
004012CB   .  85C0          test eax,eax
004012CD   .  74 10         je short bbb.004012DF
004012CF   .  68 78904000   push bbb.00409078               ASCII "Correct!\n"
004012D4   .  E8 5B020000   call bbb.00401534
004012D9   .  83C4 04       add esp,0x4
004012DC   .  33C0          xor eax,eax
004012DE   .  C3            retn
004012DF   >  68 70904000   push bbb.00409070               ASCII "Wrong\n"
004012E4   .  E8 4B020000   call bbb.00401534
004012E9   .  83C4 04       add esp,0x4
004012EC   .  33C0          xor eax,eax
004012EE   .  C3            retn
004012EF      90            nop
```


可以看到关键就是401240这个call

我们还是右键跟随

```
文件(F) 查看(V) 调试(D) 插件(P) 选项(T) 窗口(W) 帮助(H) [+] 快捷菜单 Tools BreakPoint->
暂停
0040123C    90          nop                                                          寄
0040123D    90          nop                                                          EA
0040123E    90          nop                                                          EC
0040123F    90          nop                                                          ED
00401240  $ 55          push ebp                                                     EB
00401241  . 8BEC        mov ebp,esp                                                  ES
00401243  . 83EC 0C     sub esp,0xC                 esp的值减去0xC                   EB
00401246  . 33C0        xor eax,eax                 eax清0                           ES
00401248  > 8A88 70B940 mov cl,byte ptr ds:[eax+0x40B970]  将我们的输入iqiqiya 每次去一个字节赋值给cl  ED
0040124E  . 84C9        test cl,cl                                                   EI
00401250  . 74 0A       je short bbb.0040125C                                        C
00401252  . 884C05 F4   mov byte ptr ss:[ebp+eax-0xC],cl                             P
00401256  . 40          inc eax                                                      A
00401257  . 83F8 0A     cmp eax,0xA                                                  Z
0040125A  . 7C EC       jl short bbb.00401248                                        S
0040125C  > E9 AC5F0000 jmp bbb.0040720D                                             T
00401261    90          nop                                                          D
00401262    90          nop                                                          O
00401263    90          nop                                                          EF
00401264    90          nop                                                          ST
00401265    90          nop                                                          ST
00401266    90          nop                                                          ST
00401267    90          nop                                                          ST
00401268    90          nop                                                          ST
00401269  . 33C0        xor eax,eax                                                  ST
0040126B  . 8BE5        mov esp,ebp                                                  ST
0040126D  . 5D          pop ebp                     0019FF44                         ST
0040126E    C3          retn                                                         ST
ds:[0040B970]=69 ('i')                                                           FS
cl=C0
跳转来自 0040125A

地址      HEX 数据                                        ASCII
0040B970  69 71 69 71 69 79 61 00 00 00 00 00 00 00 00 00  iqiqiya.........
0040B980  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0040B990  00 00 00 00 F0 23 00 00 02 06 00 00 06 00 00 00  ....?..?..?...

0019FF34  0019FF44
0019FF38  00401238  返回到 bbb.
0019FF3C  0040906C  ASCII "%s"
```


接下来走到jmp

```
0040720B    00          db 00
0040720C    00          db 00
0040720D  > 0000        add byte ptr ds:[eax],al    将eax的值 也就是我们输入的字符串的长度作为DS段的偏移
0040720F    00          db 00
```

明天接着写