看雪ctf2017第一题详解

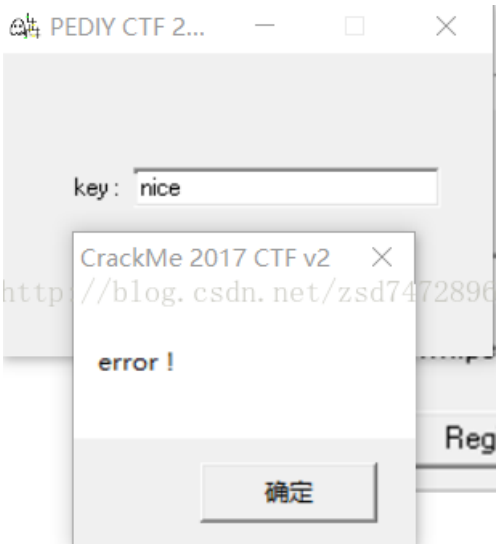
原创

zsd747289639 于 2017-06-10 17:05:42 发布 1491 收藏 1
分类专栏： PE逆向笔记 文章标签： 看雪ctf2017

本文链接：https://blog.csdn.net/zsd747289639/article/details/72993552

第一步、先瞅瞅

随便输入点啥，发现如下报错



第二步、Od加载程序，下断点

一般这种程序，第一感觉就是在下api断点，右键->查找->当前模块中的名称

找到getDlgItemTextA

右键->在输入函数上切换断点 设置断点

F9运行程序，程序断在了此处

```
748AD5C0   8BFF          mov edi,edi
748AD5C2   55            push ebp
748AD5C3   8BEC          mov ebp,esp
748AD5C5   FF75 0C       push dword ptr ss:[ebp+0xC]
748AD5C8   FF75 08       push dword ptr ss:[ebp+0x8]          WannaLOL.00401041
748AD5CB   E8 E0E5F7FF   call user32.GetDlgItem
748AD5D0   85C0          test eax,eax
748AD5D2   74 0E         je short user32.748AD5E2
748AD5D4   FF75 14       push dword ptr ss:[ebp+0x14]
748AD5D7   FF75 10       push dword ptr ss:[ebp+0x10]         user32.7483D2B3
748AD5DA   50            push eax
748AD5DB   E8 90E1F7FF   call user32.GetWindowTextA
748AD5E0   EB 0E         jmp short user32.748AD5F0
748AD5E2   837D 14 00    cmp dword ptr ss:[ebp+0x14],0x0
748AD5E6   74 06         je short user32.748AD5EE
748AD5E8   8B45 10       mov eax,dword ptr ss:[ebp+0x10]      user32.7483D2B3
748AD5EB   C600 00       mov byte ptr ds:[eax],0x0
748AD5EE   33C0          xor eax,eax
748AD5F0   5D            pop ebp                              WannaLOL.00401211
748AD5F1   C2 1000       retn 0x10
748AD5F4   CC            int3
748AD5F5   CC            int3
748AD5F6   CC            int3
748AD5F7   CC            int3
748AD5F8   CC            int3
748AD5F9   CC            int3
748AD5FA   CC            int3
```

第三步，算法分析

一路f8执行到retn返回，来到

```
00401211   .  68 F4010000    push 0x1F4                       Timeout = 500. ms
00401216   .  FF15 0070400(  call dword ptr ds:[<&KERNEL32.Sleep>]  Sleep
0040121C   .  8D45 E4        lea eax,dword ptr ss:[ebp-0x1C]
0040121F   .  50             push eax
00401220   .  E8 DB000000    call WannaLOL.00401300           判断输入的序列号是否为4位
00401225   .  83F8 04        cmp eax,0x4
00401228   .  59             pop ecx                          win32u.73BD26FC
00401229   .v 0F85 A000000(  jnz WannaLOL.004012CF
0040122F   .  6A 30          push 0x30
00401231   .  59             pop ecx                          win32u.73BD26FC
00401232   .  384D E4        cmp byte ptr ss:[ebp-0x1C],cl
00401235   .v 0F84 9400000(  je WannaLOL.004012CF
0040123B   .  384D E5        cmp byte ptr ss:[ebp-0x1B],cl
0040123E   .v 0F84 8B00000(  je WannaLOL.004012CF             判断这四个数是否为0
00401244   .  384D E6        cmp byte ptr ss:[ebp-0x1A],cl
00401247   .v 0F84 8200000(  je WannaLOL.004012CF
0040124D   .  384D E7        cmp byte ptr ss:[ebp-0x19],cl
00401250   .v 74 7D          je short WannaLOL.004012CF
00401252   .  807D E4 31     cmp byte ptr ss:[ebp-0x1C],0x31  判断第一个数是否为1，1的ascll码为31
00401256   .v 75 77          jnz short WannaLOL.004012CF
00401258   .  807D E5 35     cmp byte ptr ss:[ebp-0x1B],0x35  判断第二个数是否为5
0040125C   .v 75 71          jnz short WannaLOL.004012CF
0040125E   .v 74 03          je short WannaLOL.00401263
00401260   .v 75 01          jnz short WannaLOL.00401263
00401262      E8             db E8
00401263   .  66:B8 0800    mov ax,0x8
00401267   .  66:35 0700    xor ax,0x7
```

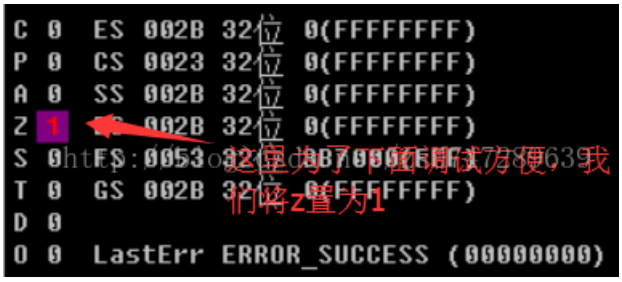
这里我们看到它调用了一个wannaLOL.00401300这么个函数，

执行返回后比较eax和4是否相等，这里我们不难想到，这里应该是判断输入的序列号的位数。

这里我们设[ebp-0x1c]=f(1),[ebp-0x1b]=f(2),ebp-0x1a=(3),ebp-0x19=f(4)

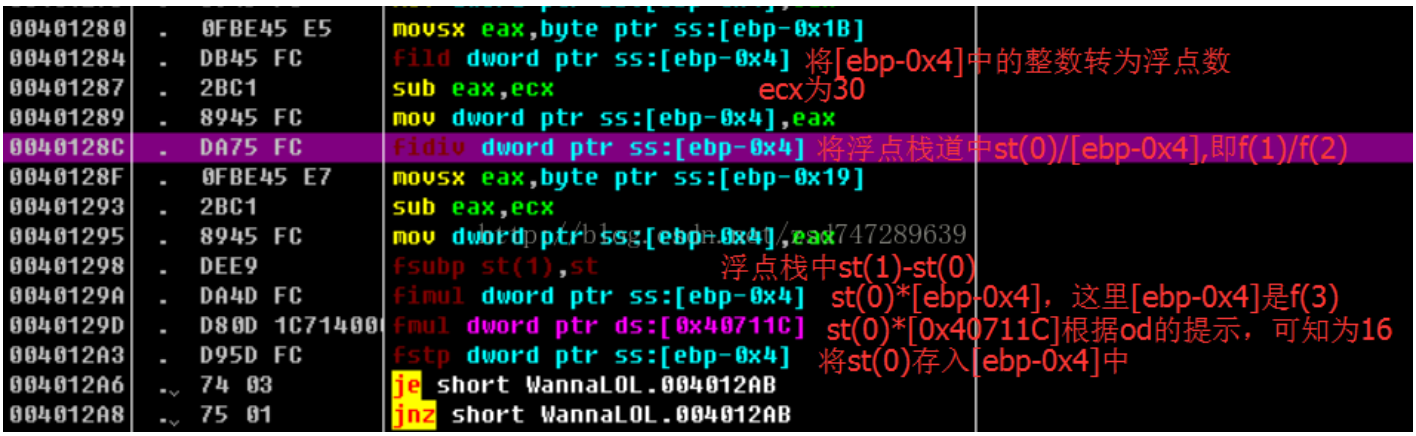
紧接着，下面判断这四个数是否为0,然后判断f(1)是否为1，f(2)是否为5。图上已经标注得很清楚了。

这里我们为了进一步调试，需要在判断跳转时候将z标志位置1。在判断f(1)和f(2)时。



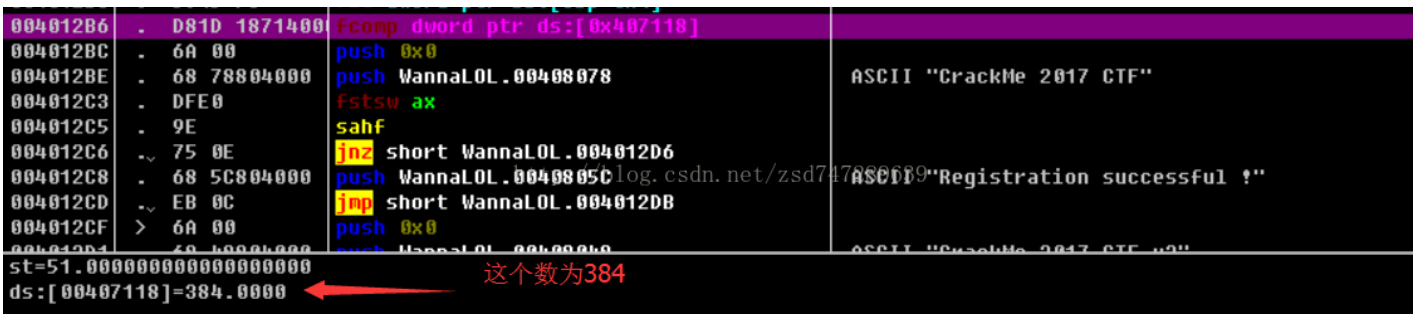
接着我们便来到了这里，需要进行一些简单的浮点计算,最后和[0x407118]处数值比较

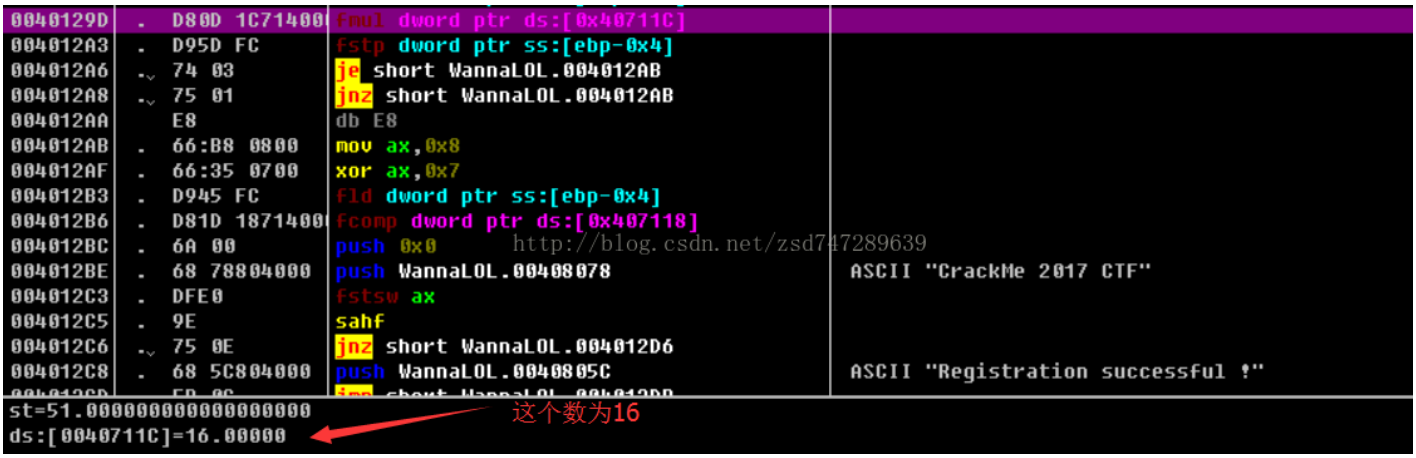
如图，我已经分析得很清楚了



我们可以整理得这个浮点计算的公式为 (f(3)-0.2)*f(4)*[0x40711c]=[0x407118]

其中根据od的提示，我们可以知道[0x407118]=384，[0x40711c]=16





所以公式整理得(f(3)-0.2)*f(4)=24，易估得，f(3)=5,f(4)=5

最后得序列号为：1555

输入验证下：

破解成功！