看雪CTF2017第一题简单分析

原创

Youngs0xFF 于 2017-06-01 17:44:19 发布 2001 收藏
分类专栏： CrackMe

本文链接：https://blog.csdn.net/Youngs0xff/article/details/72833576

本题比较简单，无壳、无VM、无密码学，适合入门练习

中午一觉睡醒，逛逛论坛才发现有ctf比赛，就随便看了一下

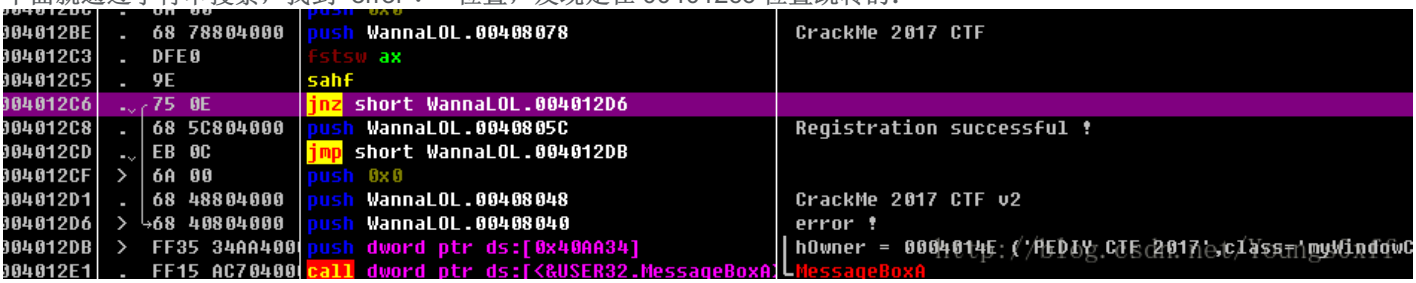
动态调一下就好了，发现确实是多解，下面我就简单说一下自己的分析流程吧

0x00 定位关键跳转

首先拿到CrackMe，就直接随便输入了一串字符串“111111”

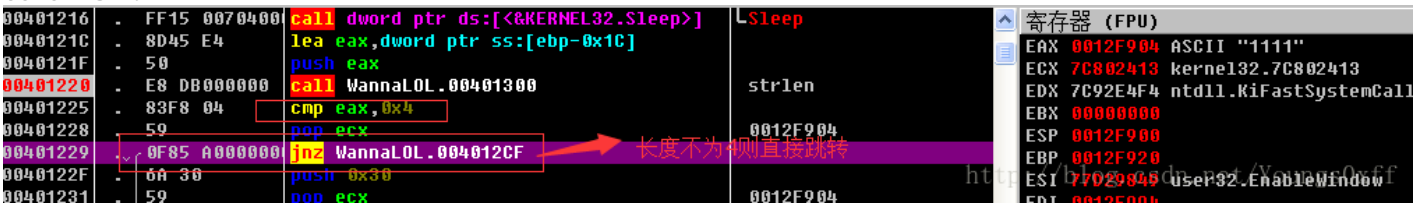
看到是有弹窗提示的，根据报出的“error！”字符串就能直接定位到关键跳转位置了。

下面就通过字符串搜索，找到 “error ！ ” 位置，发现是在 004012c6 位置跳转的：



0x01 判断输入值的长度

那么我们继续往上分析，可以查看到我们输入后传入值的部分，并通过 strlen 来获取其长度是否为 4 ，不为 4 则直接跳转到 004012CF ：



现在可以确定key的长度为4

0x02 验证key前2位的值

这里验证key的值是否都为字符串“0”，如果4位的key又一个为“0”，则直接跳转到“error”弹出框

然后接着就是继续验证key前2位字符串的值，即给出字符串的前2位为“15”

如果 key 的前 2 位的值不为 “15”，则直接跳转到 “error” 弹出框

```
00401229   . /0F85 A0000000 jnz WannaLOL.004012CF
0040122F   . |6A 30         push 0x30
00401231   . |59            pop ecx
00401232   . |384D E4       cmp byte ptr ss:[ebp-0x1C],cl //判断key第1个字符是否为“0”
00401235   . |0F84 94000000 je WannaLOL.004012CF
0040123B   . |384D E5       cmp byte ptr ss:[ebp-0x1B],cl //判断key第2个字符是否为“0”
0040123E   . |0F84 8B000000 je WannaLOL.004012CF
00401244   . |384D E6       cmp byte ptr ss:[ebp-0x1A],cl //判断key第3个字符是否为“0”
00401247   . |0F84 82000000 je WannaLOL.004012CF
0040124D   . |384D E7       cmp byte ptr ss:[ebp-0x19],cl //判断key第4个字符是否为“0”
00401250   . |74 7D         je short WannaLOL.004012CF
00401252   . |807D E4 31    cmp byte ptr ss:[ebp-0x1C],0x31 //判断key第1个字符是否等于“1”，不等于则直接跳转弹出“
00401256   . |75 77         jnz short WannaLOL.004012CF
00401258   . |807D E5 35    cmp byte ptr ss:[ebp-0x1B],0x35 //判断key第2个字符是否等于“5”，不等于则直接跳转弹出“
0040125C   . |75 71         jnz short WannaLOL.004012CF
0040125E   . |74 03         je short WannaLOL.00401263
00401260   . |75 01         jnz short WannaLOL.00401263
```

这里得到前2为key的值为“15”

0x03 分析算法

这里运作流程：

1、取key中第3位的十六进制值，然后减去0x30，这里则假定值为a

2、取key中第1位值为“1”的十六进制值即0x31，然后减去0x30，0x31-0x30 =1

3、取key中第2位值为“5”的十六进制值即0x35，然后减去0x30，0x31-0x30 =5 ，接着就是1除以5得出浮点数0.2

4、取key中第4位的十六进制值，然后减去0x30，这里则假定值为b

5、接着就是（a-0.2）*b乘以16得出的结果为c

6、判断c与384是否相等，相等则Registration successful !，不相同则“error”

公式： (a-0.2)*b*16 = 384 求解 a 和 b

```
0040126B   .  0FBE45 E6     movsx eax,byte ptr ss:[ebp-0x1A] //取输入key的第3位值
0040126F   .  2BC1          sub eax,ecx                      //减去0x30,得到a
00401271   .  8945 FC       mov dword ptr ss:[ebp-0x4],eax   //把得到的值保存在[ebp-0x4]中
00401274   .  0FBE45 E4     movsx eax,byte ptr ss:[ebp-0x1C] //取输入key的第1位值“1”
00401278   .  DB45 FC       fild dword ptr ss:[ebp-0x4]      //把[ebp-0x4]中保存的值压入到ST(0)中
0040127B   .  2BC1          sub eax,ecx                      //0x31减去0x30，则为1
0040127D   .  8945 FC       mov dword ptr ss:[ebp-0x4],eax   //把得到的值1保存在[ebp-0x4]中
00401280   .  0FBE45 E5     movsx eax,byte ptr ss:[ebp-0x1B] //取输入key的第2位值“5”
00401284   .  DB45 FC       fild dword ptr ss:[ebp-0x4]      //把[ebp-0x4]中保存的值1压入到ST(0)中
00401287   .  2BC1          sub eax,ecx                      //0x35减去0x30，则为5
00401289   .  8945 FC       mov dword ptr ss:[ebp-0x4],eax   //把得到的值5保存在[ebp-0x4]中
0040128C   .  DA75 FC       fidiv dword ptr ss:[ebp-0x4]     //st(0)Z中的值1除以[ebp-0x4]中的值5，得到0.2保存到
0040128F   .  0FBE45 E7     movsx eax,byte ptr ss:[ebp-0x19] //取输入key的第4位值
00401293   .  2BC1          sub eax,ecx                      //减去0x30，得到b
00401295   .  8945 FC       mov dword ptr ss:[ebp-0x4],eax   //把得到的值b保存在[ebp-0x4]中
00401298   .  DEE9          fsubp st(1),st                   //st(1)-st并保存在st(0)中，即（a-0.2）
0040129A   .  DA4D FC       fimul dword ptr ss:[ebp-0x4]     //st(0)乘以[ebp-0x4] 中保存的值b，结果保存在st(0)
0040129D   .  D80D 1C714000 fmul dword ptr ds:[0x40711C]     //数据段ds:[0x40711C]值为16，这里则是st(0)乘以16
004012A3   .  D95D FC       fstp dword ptr ss:[ebp-0x4]      //把上面得到值保存在[ebp-0x4]中
004012A6   .  74 03         je short WannaLOL.004012AB
004012A8   .  75 01         jnz short WannaLOL.004012AB
004012AA      E8            db E8
004012AB   >  66:B8 0800    mov ax,0x8
004012AF   .  66:35 0700    xor ax,0x7
004012B3   .  D945 FC       fld dword ptr ss:[ebp-0x4]       //取出[ebp-0x4]中保存的值
004012B6   .  D81D 18714000 fcomp dword ptr ds:[0x407118]    //得出的值与384比较是否相等
```

0x04 编写算法脚本

这里根据上面分析，写了个简单的脚本，跑出了 2 个结果 “151N” 和 "1555"

```
def test(Max):
 for i in range(Max):
  for j in range(Max):
   if((i - 0.2)*j ==24.0)
    b = chr(int(str(hex(i+48)),16))
    c = chr(int(str(hex(j+48)),16))
    print "key--->15%s"%b+"%s"%c
if __name__ == '__main__':
 test(255)
```

跑出来的结果：