第七届山东省大学生网络安全技能大赛(科来杯)逆向WP

原创

北风~ 于 2021-03-14 21:42:39 发布 628 收藏

分类专栏： CTF 逆向与保护

本文链接：https://blog.csdn.net/weixin_45055269/article/details/114576865

版权

CTF 同时被 2 个专栏收录

31 篇文章 3 订阅

订阅专栏

逆向与保护

65 篇文章 4 订阅

订阅专栏

file

出题人有个不好的习惯，总喜欢清除回收站，这次在出题的过程中又把文件给删了，你能帮他还原回来吗？flag无标准格式,提交答案请加上flag{}

elf 静态分析

```
12
13  v12 = __readfsqword(0x28u);
14  v8 = fopen(argv[1], "r");
15  if ( !v8 )
16    __assert_fail("file!=NULL", "file.c", 0x8Fu, "main");
17  for ( i = 0; (unsigned int)__isoc99_fscanf(v8, "%c", (char *)&v11[64] + i) != -1; ++i )
18    ;
19  v5 = 0;
20  v9 = (_BYTE *)encode(flllag);
21  v10 = (_BYTE *)encode(sttr_home);
22  for ( j = 0; j <= 63; j += 2 )
23    v11[v5++] = sub_123887656((unsigned int)sttr_home[j], (unsigned int)sttr_home[j + 1]);
24  *v9 = encode(flllag);
25  for ( k = 0; k < v5; ++k )
26  {
27    if ( *flllag != (k ^ *((char *)&v11[64] + k) ^ v11[k]) )
28    {
29      printf("Your file is wrong!! try again");
30      return 0;
31    }
32    ++flllag;
33  }
34  *v10 = encode(sttr_home);
```

核心是异或操作，flllag虽然看着被encode加密了，但*flllag地址里的值还是没变，这点小坑，接着就是写脚本，将文件作为参数带入就可以了，v11动调可拿

```
v11=[0x66,0x4e,0x6,0x22,0x66,0x25,0x42,0x5d,0x56,0x2e,0x76,0x6e,0x4,0x2d,0x42,0x2c,0x7,0x2c,0x45,0x69,0x2d,0x12,
0x5c,0x7e,0x65,0x52,0x60,0x69,0x54,0x64,0x66,0x43]
flllag=bytearray(b'flag{hello_player_come_on_hahah}')
flag=[0]*32
for i in range(0x20):
 flag[i]=hex(i^flllag[i]^v11[i])
print(' '.join(flag))
```

winhex打开，将hex值一个一个写入就可了

```
 0  1  2  3  4  5  6  7   8  9 10 11 12 13 14 15      ANSI ASCII
00 23 65 46 19 48 21 36  32 48 23 15 64 41 35 46  #eF H!62H# dA5F
65 62 34 15 54 62 15 06  13 14 12 13 20 18 10 21  eb4 Tb         !
```

flag是将文件的md5

914a7b9df69eab5b74b9edb7070e53e8

flag{914a7b9df69eab5b74b9edb7070e53e8}

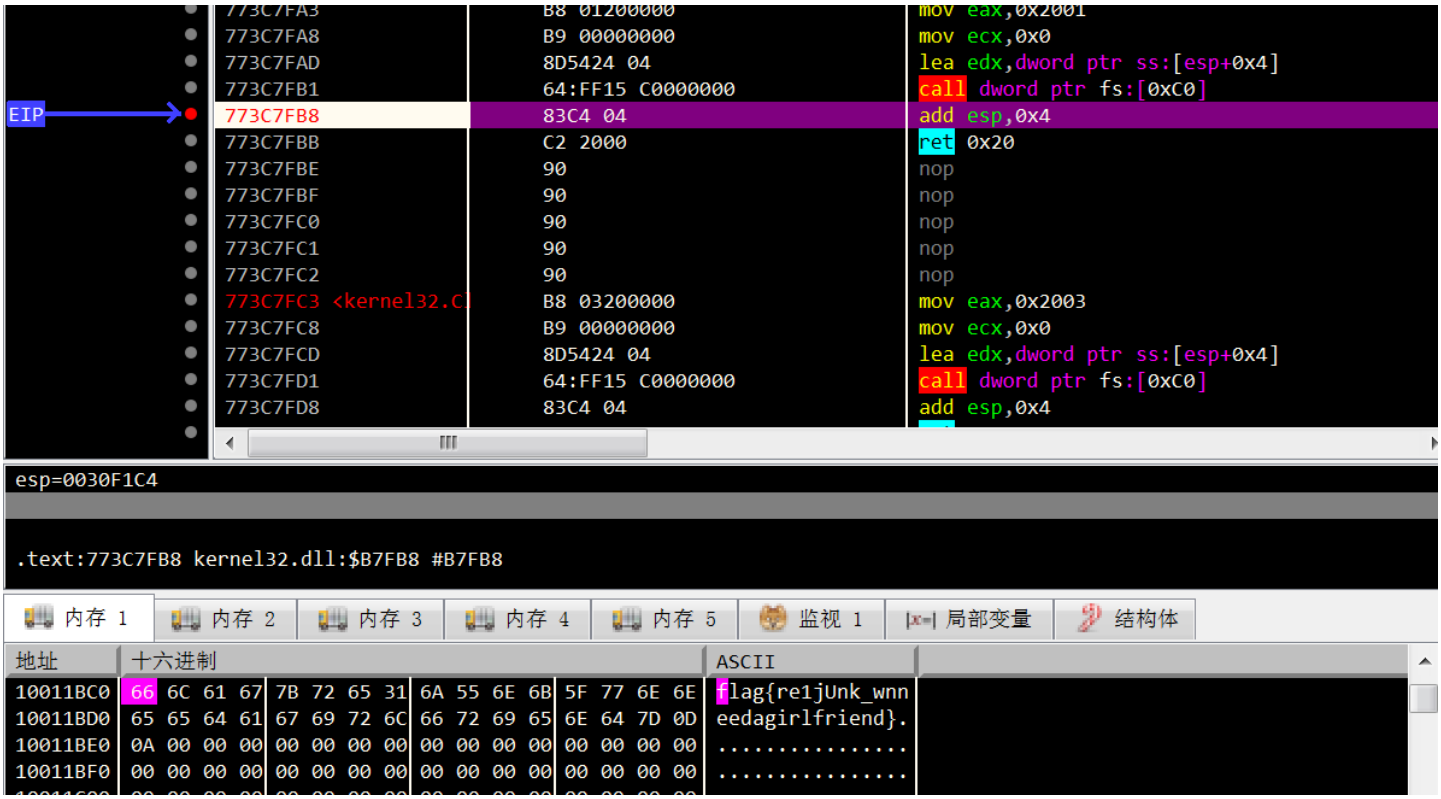
not_only_smc

所见非所得

smc+花指令

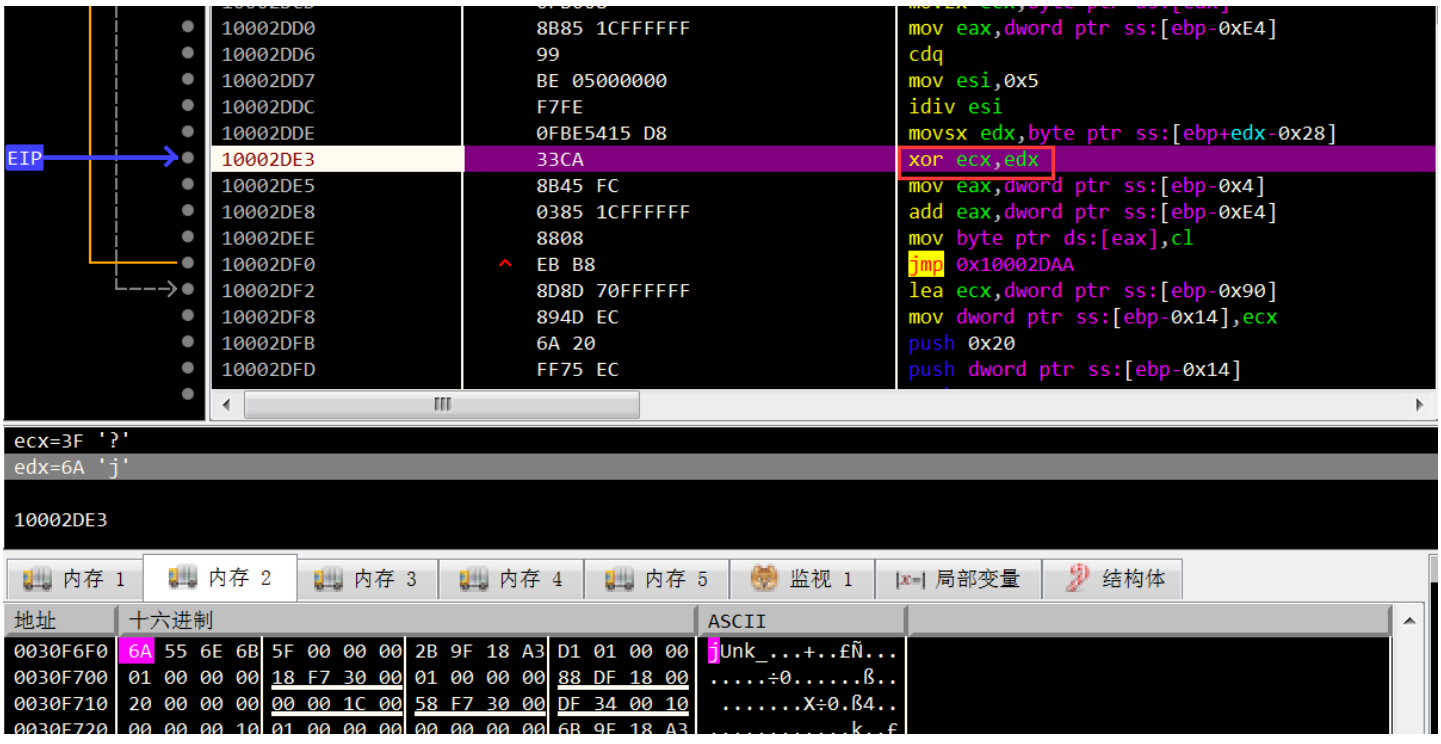
DIE查壳后发现UPX壳，esp手脱，修复导入表后，还是打不开，只能带壳分析，在VirtualAlloc下断，F9运行，直到停到输入，点击EIP找到输入点的位置下断,输入字符串，这个字符串是随便输入，目的是找到输入字符串在内存的位置

```
773C7FA1    90          nop
773C7FA2    90          nop
```

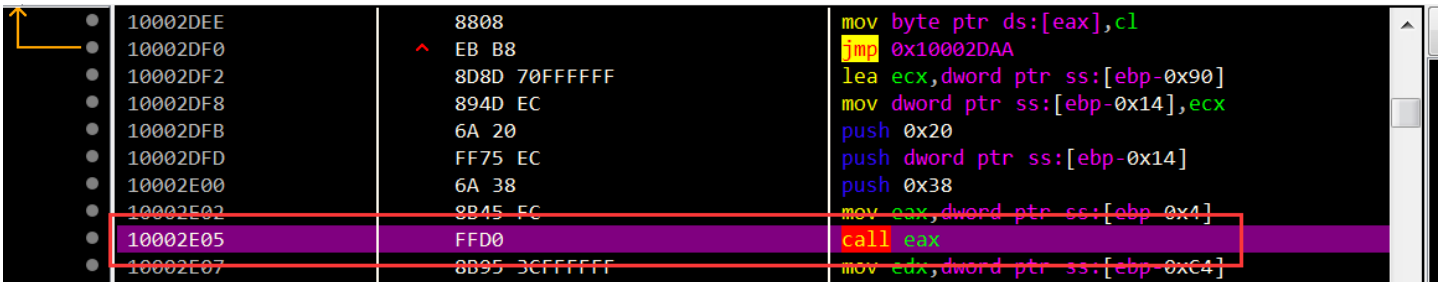


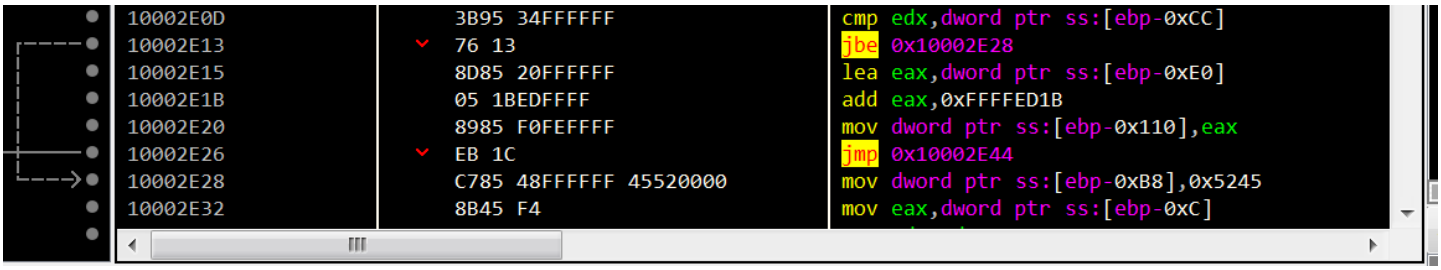
接着F9，程序会在VirtualAlloc停下，然后单步F8就能走到程序领空。

这还是一个smc，取输入的第8位的后五位为key进行异或解密，这怎么搞？key不知道，连真正的结果也异或不出。



于是在下面看到了，他去call这个位置

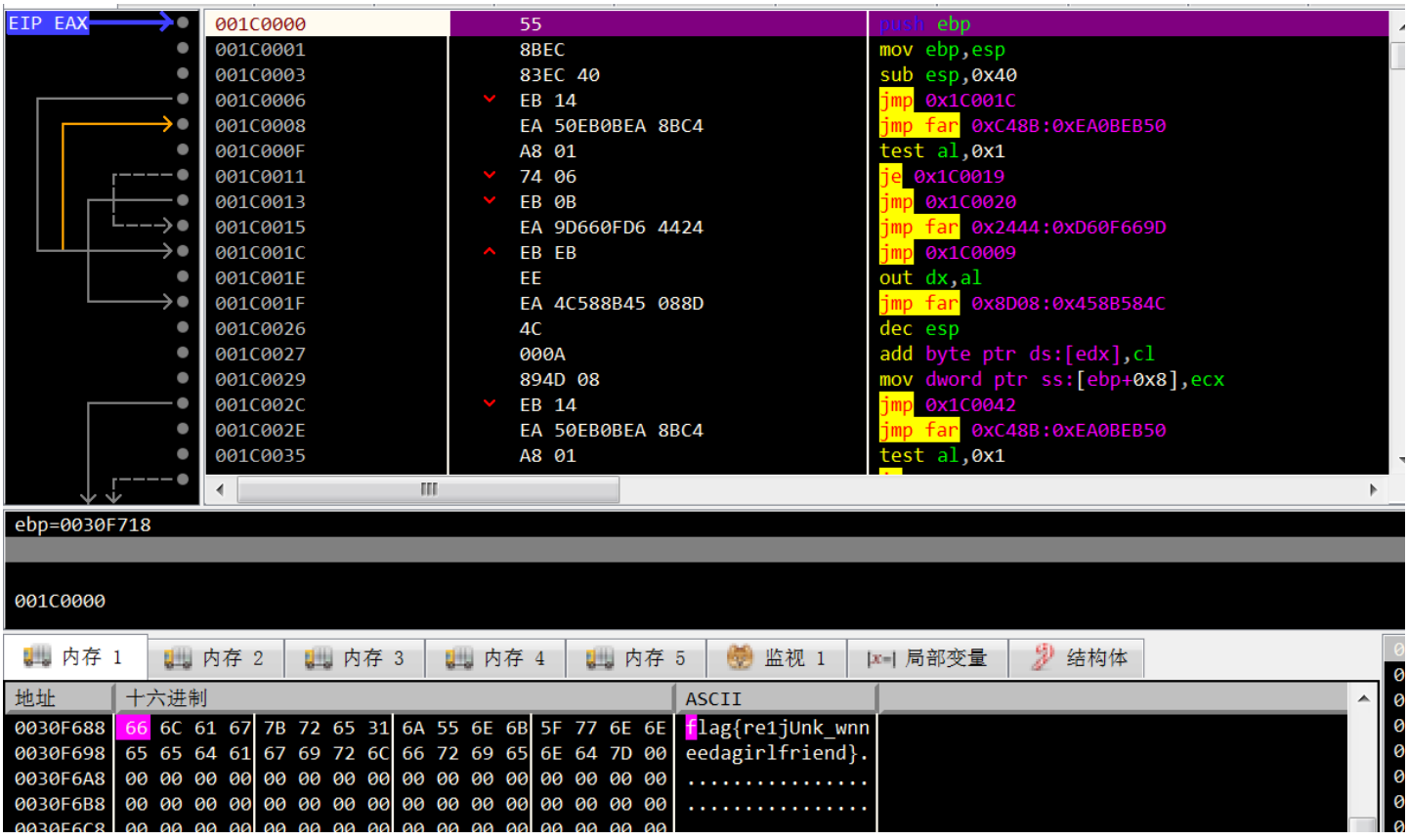




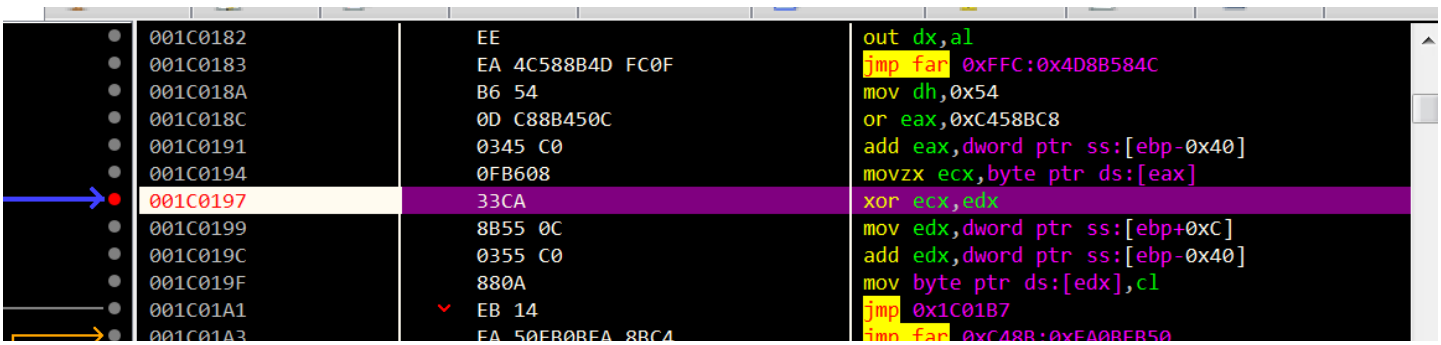
由于这种函数的头部的硬编码固定，所以可以异或出真正的key

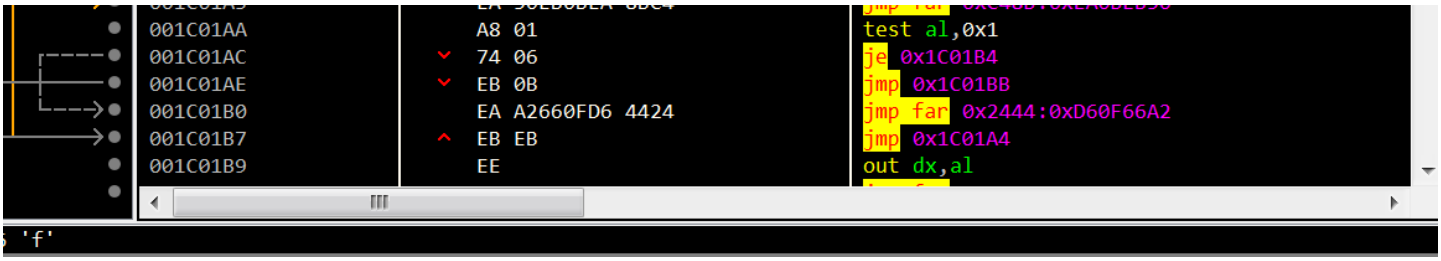
```
tmp=[0x55,0x8B,0xEC,0x81,0xEC]
tmp0=[0x3f,0xde,0x82,0xea,0xb3]
flag=[0]*5
for i in range(5):
 flag[i]=tmp[i]^tmp0[i]
print(''.join(map(chr,flag)))
#jUnk_
```

所以接着拼凑测试flag，flag{re1jUnk_wnneedagirlfriend}，接着输入，
遇到了大片的花指令

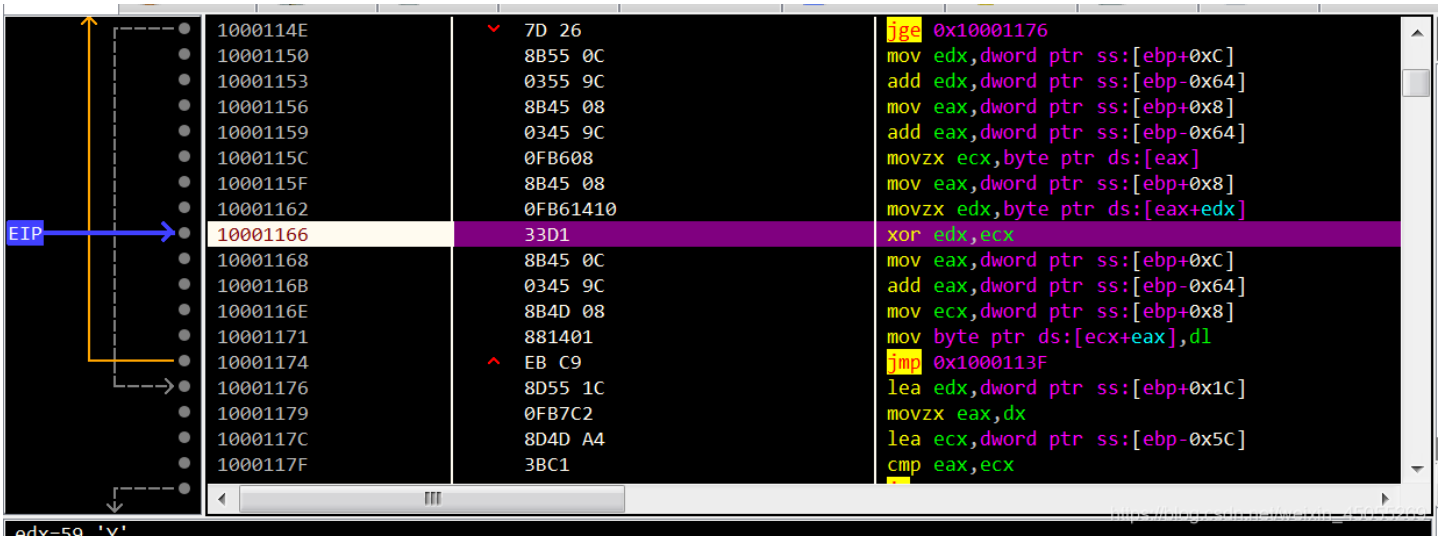


可以去花，或者这么调试也可，在字符串的内存位置下硬件断点，F9结合F2下断动调即可，是个体力活





以及换位异或，好难的

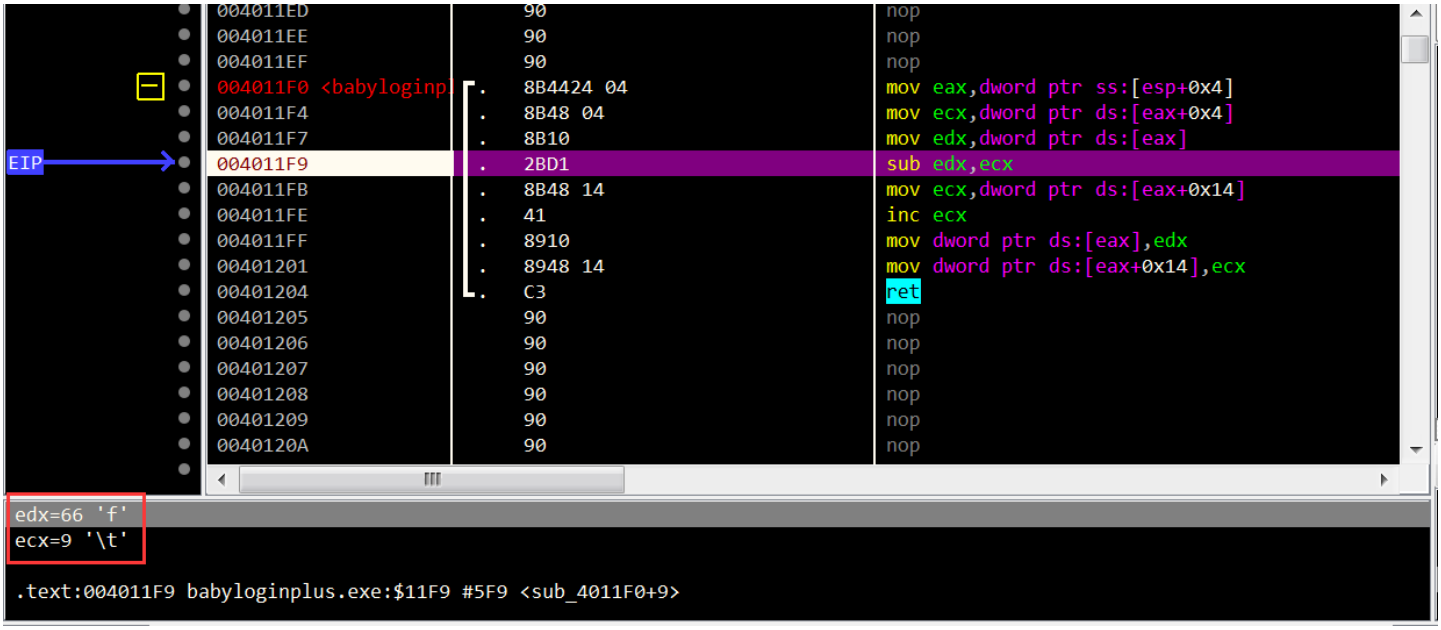


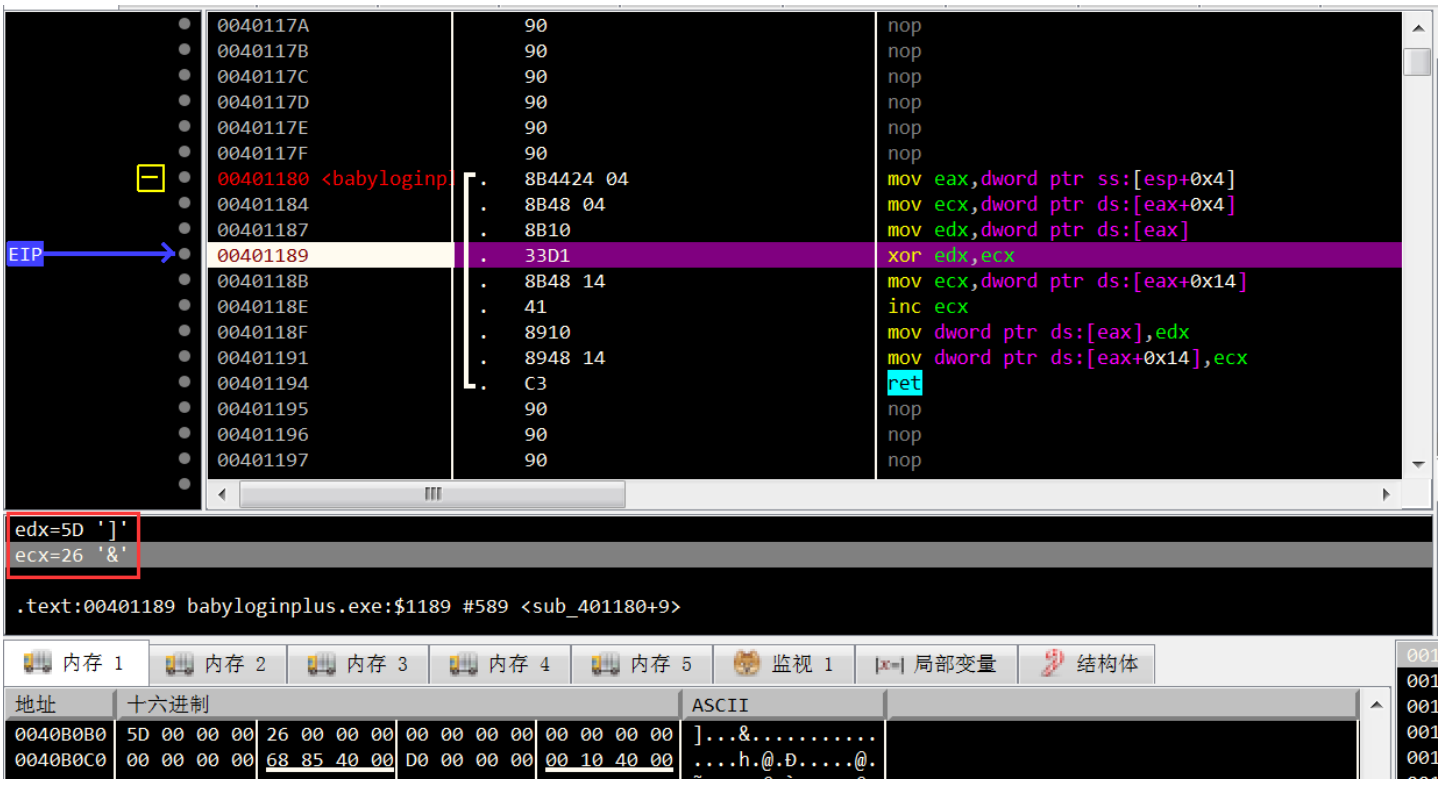

```python
ans = [0xE8, 0x90, 0x24, 0xE6, 0x0A, 0xE3, 0xF7, 0xA8, 0x09, 0xC0, 0x35, 0x74, 0x26, 0x4D, 0xA0, 0x2D,
       0xD6, 0x1A, 0x5A, 0x5C, 0x16, 0x2D, 0xF0, 0x46, 0x44, 0x10, 0x5F, 0x83, 0x5B, 0xBE, 0x86, 0xAC]
key = [0xBB, 0x35, 0xAF, 0x29, 0xA3, 0x1D, 0x97, 0x11, 0x8B, 0x05, 0x7F, 0xF9, 0x73, 0xED, 0x67, 0xE1,
       0x5B, 0xD5, 0x4F, 0xC9, 0x43, 0xBD, 0x37, 0xB1, 0x2B, 0xA5, 0x1F, 0x99, 0x13, 0x8D, 0x07, 0x81]
for i in range(65):
    j = 0x10
    while j > 0:
        for k in range(j):
            ans[j+k] = ans[j+k] ^ ans[k]
        j = j >> 1
for i in range(32):
    ans[i]^=key[i]
    print(chr(ans[i]),end='')
#SMc_AnD_jUnk_C0de_1s_s0_fuunn~!
```

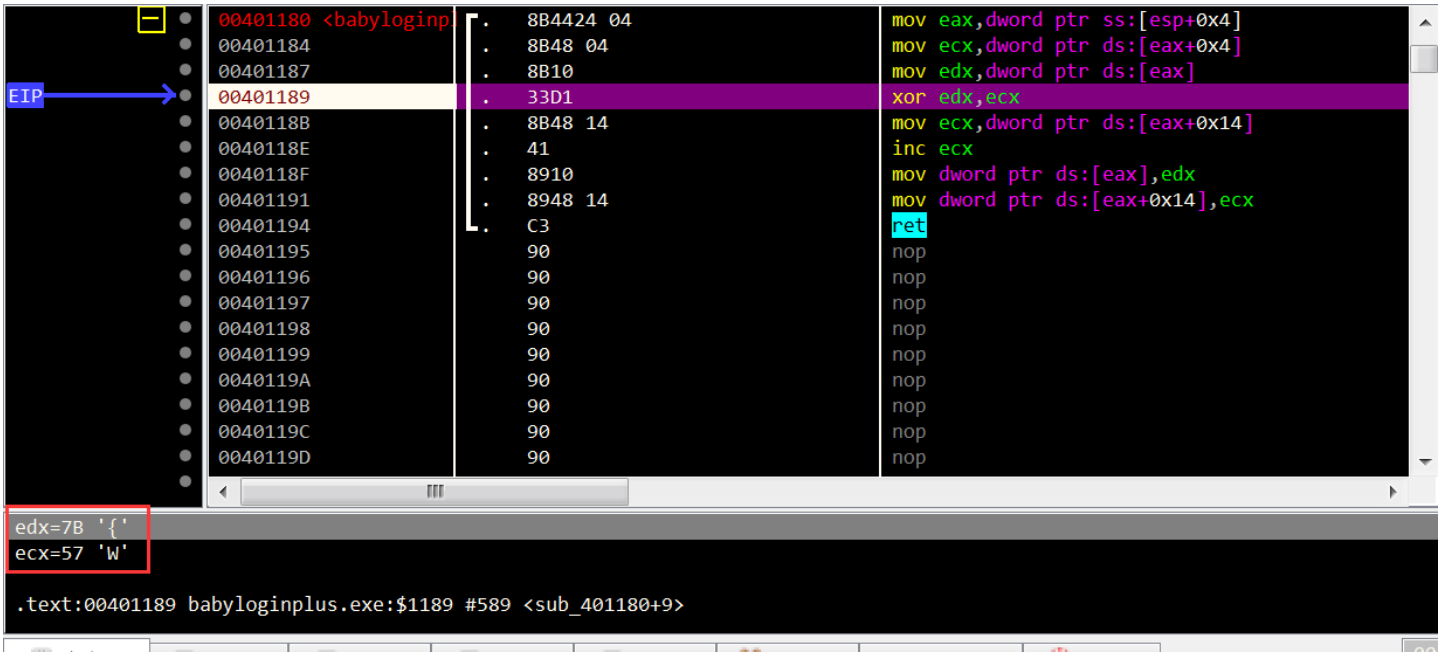
flag{SMc_AnD_jUnk_C0de_1s_s0_fuunn~!}

babyLoginPlus

新品发布会。『babyLogin Plus』哪一面，都是亮(汇)点(编)
vm类的题还是没有什么好的办法，找到输入，下硬件断点，跟就可







最后有个+0x6的操作

写脚本逆向

```
src = [0x32, 0x26, 0x18, 0x21, 0x41, 0x23, 0x2A, 0x57, 0x44, 0x29, 0x35, 0x12, 0x20, 0x17, 0x45, 0x1C,
       0x68, 0x2D, 0x7A, 0x79, 0x47, 0x7F, 0x44, 0x09, 0x1E, 0x75, 0x41, 0x2A, 0x19, 0x34, 0x76, 0x47,
       0x14, 0x50, 0x52, 0x76, 0x58]
key = [0x57, 0x65, 0x6C, 0x63, 0x6F, 0x6D, 0x65, 0x5F, 0x74, 0x6F, 0x5F, 0x73, 0x64, 0x6E, 0x69, 0x73,
       0x63, 0x5F, 0x32, 0x30, 0x31, 0x38, 0x5F, 0x42, 0x79, 0x2E, 0x5A, 0x65, 0x72, 0x6F, 0x00, 0x00,
       0x00, 0x00, 0x00, 0x00, 0x00]
flag = ''
for i in range(len(src)):
    flag += chr(((src[i]-0x6) ^ 0x26 ^ key[i])+9)
print(flag)
#flag{_p1us_babyL0gin_pPpPpPpPp_p1us_}
```