XCTF easyhook

原创

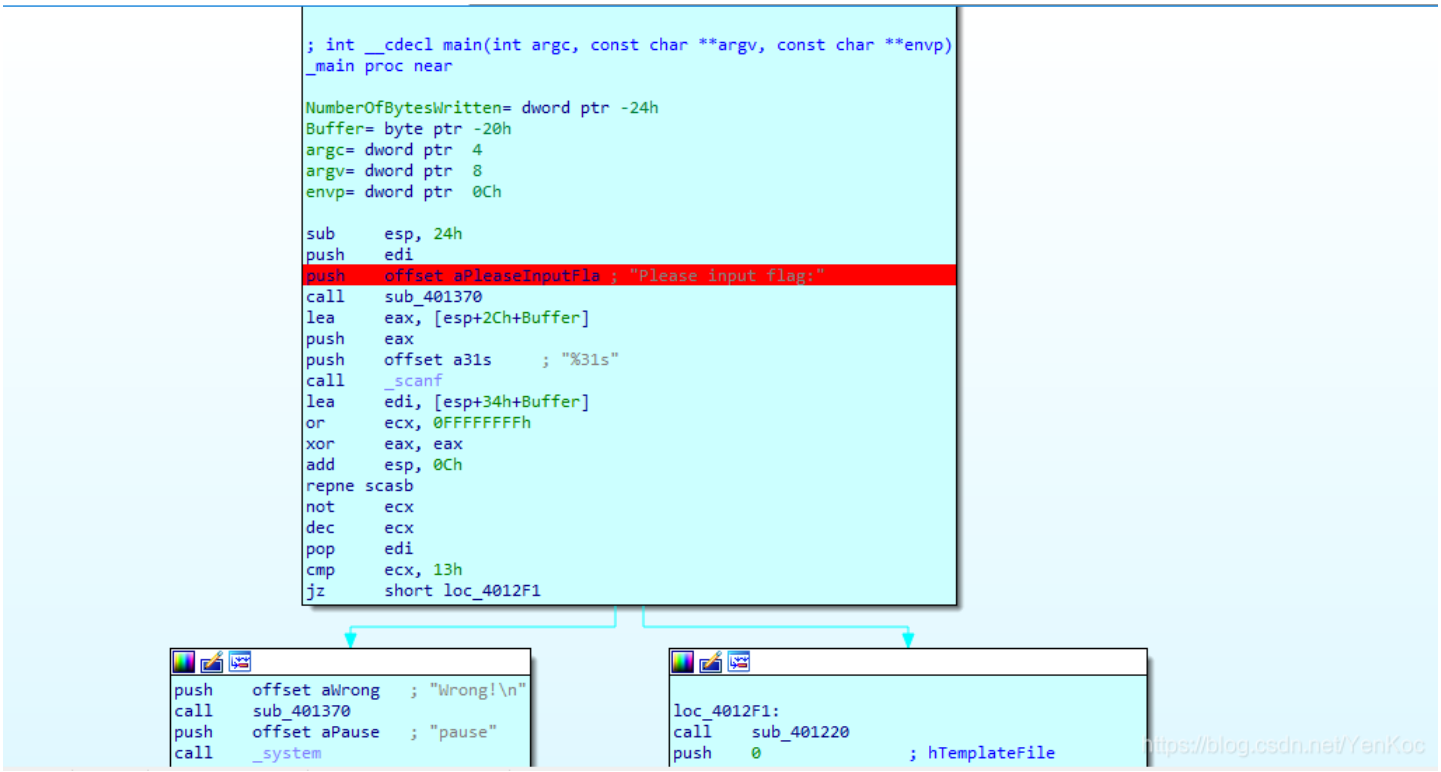
YenKoc 于 2020-04-20 14:57:32 发布 367 收藏

分类专栏： XCTF

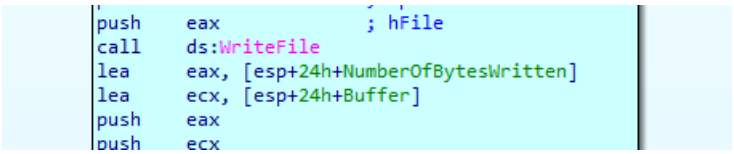
本文链接：https://blog.csdn.net/YenKoc/article/details/105631031

这题感觉主要是考一个动调，学会要观察堆栈和寄存器，以此来找到加密的函数

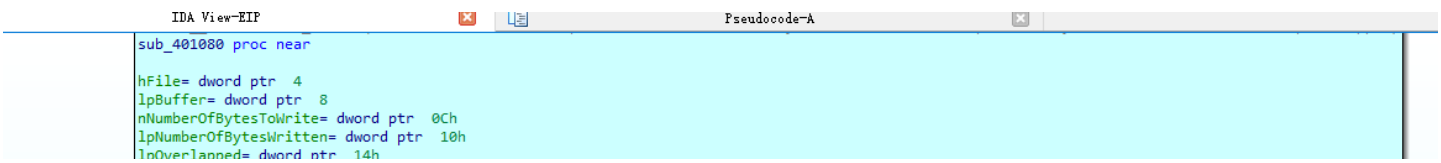
载入ida，开始动调。



先F8开始，看看经过哪里，字符串就被加密了，



F7单步进去看看。



```
push    ebx
push    ebp
push    esi
mov     esi, [esp+0Ch+nNumberOfBytesToWrite]
push    edi
mov     edi, [esp+10h+lpBuffer]
push    esi
push    edi
call    sub_401000
add     esp, 8
mov     ebx, eax
call    sub_401140
mov     eax, [esp+10h+lpOverlapped]
mov     ebp, [esp+10h+lpNumberOfBytesWritten]
mov     ecx, [esp+10h+hFile]
push    eax             ; lpOverlapped
push    ebp             ; lpNumberOfBytesWritten
push    esi             ; nNumberOfBytesToWrite
push    edi             ; lpBuffer
push    ecx             ; hFile
call    ds:WriteFile
test    ebx, ebx
jz      short loc_4010BF
```


那个标红的地方就是加密函数。

```
signed int __cdecl sub_401000(int a1, signed int a2)
{
  char v2; // al
  char v3; // bl
  char v4; // cl
  int v5; // eax

  v2 = 0;
  if ( a2 > 0 )
  {
    do
    {
      if ( v2 == 18 )
      {
        *(_BYTE *)(a1 + 18) ^= 0x13u;
      }
      else
      {
        if ( v2 % 2 )
          v3 = *(_BYTE *)(v2 + a1) - v2;
        else
          v3 = *(_BYTE *)(v2 + a1 + 2);
        *(_BYTE *)(v2 + a1) = v2 ^ v3;
      }
      ++v2;
    }
    while ( v2 < a2 );
  }
  v4 = 0;
  if ( a2 <= 0 )
    return 1;
  v5 = 0;
  while ( byte_40A030[v5] == *(_BYTE *)(v5 + a1) )
  {
    v5 = ++v4;
    if ( v4 >= a2 )
```


然后跳回去的话，有个地方很坑，后面判断长度为21。。。。一开始就已经说明了是19的长度，又看到那个字符串是this_is_not_flag，大概就猜到是个幌子了。。。无视它，直接写脚本跑

```
res=[
    0x61,0x6A, 0x79, 0x67, 0x6B, 0x46, 0x6D, 0x2E, 0x7F, 0x5F, 0x7E,0x2D, 0x53, 0x56, 0x7B, 0x38, 0x6D, 0x4C, 0x
6E, 0x00
]
#奇数时 (input[i]-i)^i
#偶数  (input[i+2])^i
flag=list("12345678912345678900")
for i in range(0,18):
    if i%2==1:
        flag[i]=chr((res[i]^i)+i)
    else:
        flag[i+2]=chr(res[i]^i)
print("".join(flag))
print(res[18]^0x13)
```