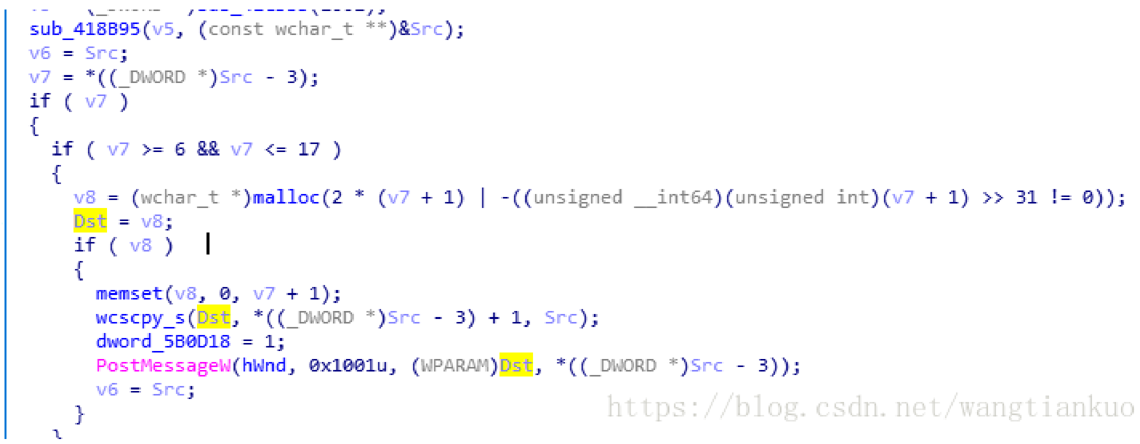
百度杯“百度杯”CTF比赛 十一月场 Reverse CrackMe01 Writeup

原创

wangtiankuo 于 2018-07-25 14:16:01 发布 2511 收藏 2
分类专栏： writeup

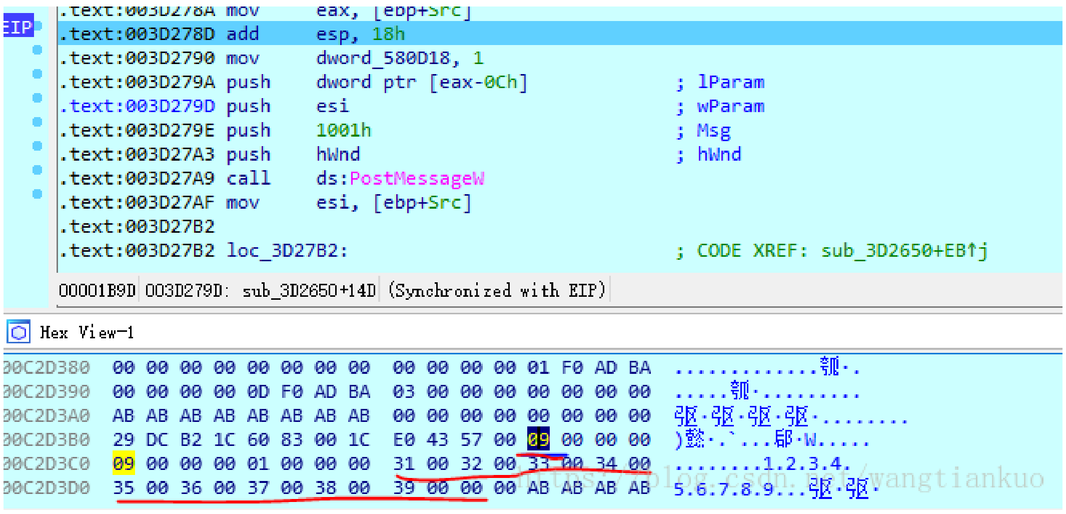
writeup 专栏收录该内容
3 篇文章 1 订阅
订阅专栏
题目链接：https://www.ichunqiu.com/battalion

取文本框中内容的函数是GetWindowTextW，给GetWindowTextW下断点，运行，5次F9之后，才可以在文本框中输入字符，随便输入123456789，鼠标放在click上后就从running状态变成了Paused，继续运行，会将输入拷贝到另一个地方，然后调用PostMessage函数。



```
  sub_418B95(v5, (const wchar_t **)&Src);
  v6 = Src;
  v7 = *((_DWORD *)Src - 3);
  if ( v7 )
  {
    if ( v7 >= 6 && v7 <= 17 )
    {
      v8 = (wchar_t *)malloc(2 * (v7 + 1) | -((unsigned __int64)(unsigned int)(v7 + 1) >> 31 != 0));
      Dst = v8;
      if ( v8 )
      {
        memset(v8, 0, v7 + 1);
        wcscpy_s(Dst, *((_DWORD *)Src - 3) + 1, Src);
        dword_5B0D18 = 1;
        PostMessageW(hWnd, 0x1001u, (WPARAM)Dst, *((_DWORD *)Src - 3));
        v6 = Src;
      }
    }
```

用IDA动态调试，可以看到PostMessage函数各个参数的值，lParam是输入的个数，wParam是输入的内容。hWnd是接收消息的窗口句柄。



```
.text:003D278A mov     eax, [ebp+Src]
.text:003D278D add     esp, 18h
.text:003D2790 mov     dword_580D18, 1
.text:003D279A push    dword ptr [eax-0Ch]          ; lParam
.text:003D279D push    esi                          ; wParam
.text:003D279E push    1001h                        ; Msg
.text:003D27A3 push    hWnd                         ; hWnd
.text:003D27A9 call    ds:PostMessageW
.text:003D27AF mov     esi, [ebp+Src]
.text:003D27B2
.text:003D27B2 loc_3D27B2:                          ; CODE XREF: sub_3D2650+EB↑j
00001B9D 003D279D: sub_3D2650+14D (Synchronized with EIP)
```

Hex View-1

```
00C2D380  00 00 00 00 00 00 00 00  00 00 00 00 01 F0 AD BA  .............鄔·.
00C2D390  00 00 00 00 0D F0 AD BA  03 00 00 00 00 00 00 00  .....鄔·.........
00C2D3A0  AB AB AB AB AB AB AB AB  00 00 00 00 00 00 00 00  張·張·張·張·........
00C2D3B0  29 DC B2 1C 60 83 00 1C  E0 43 57 00 09 00 00 00  )鷙·.`...郈·W.....
00C2D3C0  09 00 00 00 01 00 00 00  31 00 32 00 33 00 34 00  ........1.2.3.4.
00C2D3D0  35 00 36 00 37 00 38 00  39 00 00 00 AB AB AB AB  5.6.7.8.9...張·張·
```

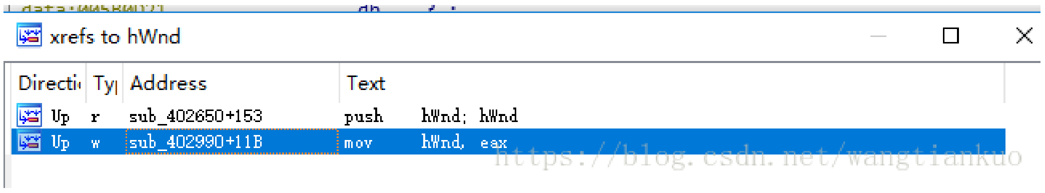
（只有前几次调试的时候会执行到PostMessage，后面调试都一直在循环调用GetWindowTextW出不来）

发送消息的函数：SendMessage

寄送消息的函数：PostMessage

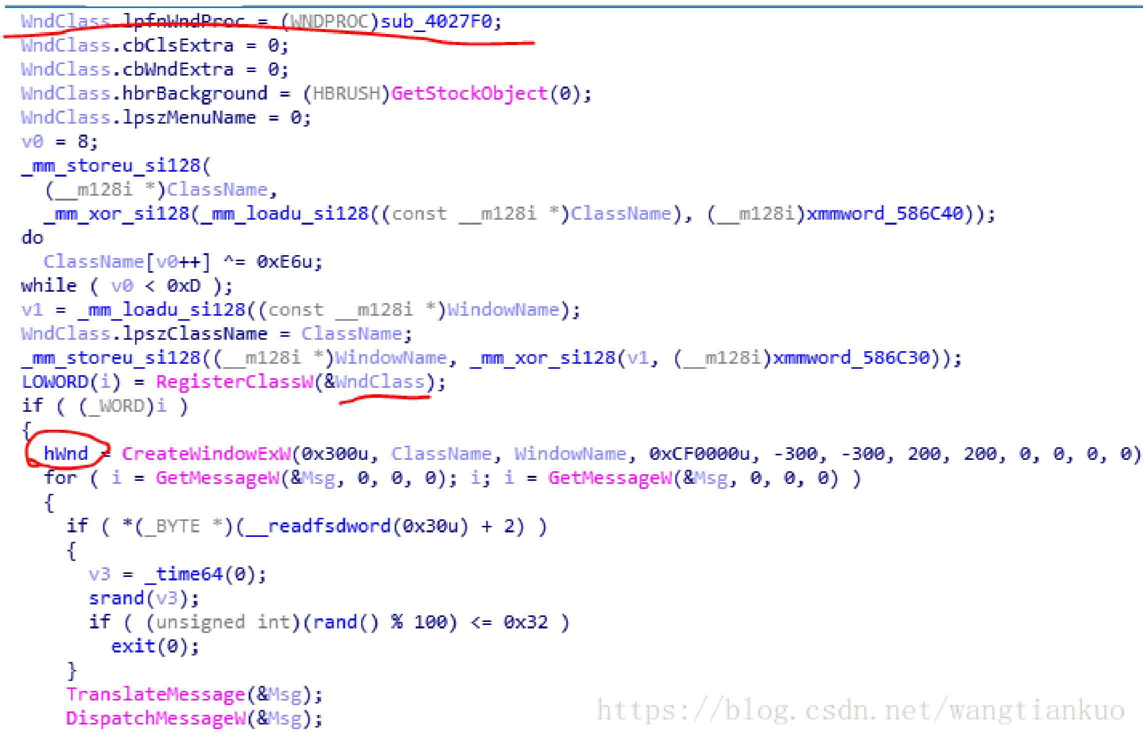
消息的接收：GetMessage、PeekMessage

PostMessageW的第一个参数是接收消息的窗口句柄，查看它的交叉引用，发现了一共被两个函数使用。



```
xrefs to hWnd
Directi Ty  Address          Text
Up      r   sub_402650+153   push    hWnd; hWnd
Up      w   sub_402990+11B   mov     hWnd, eax
```

查看402990，hWnd是调用CreateWindowExW之后得到的窗口句柄。



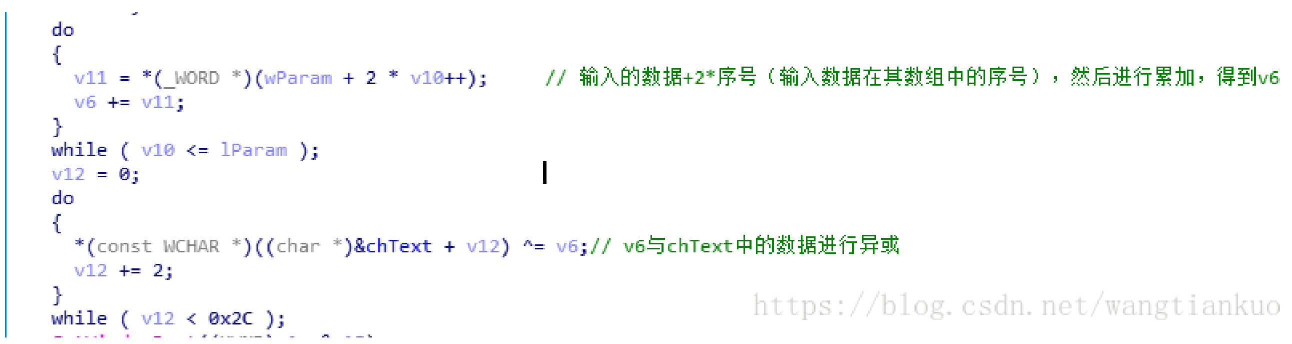
```
WndClass.lpfnWndProc = (WNDPROC)sub_4027F0;
WndClass.cbClsExtra = 0;
WndClass.cbWndExtra = 0;
WndClass.hbrBackground = (HBRUSH)GetStockObject(0);
WndClass.lpszMenuName = 0;
v0 = 8;
_mm_storeu_si128(
  (__m128i *)ClassName,
  _mm_xor_si128(_mm_loadu_si128((const __m128i *)ClassName), (__m128i)xmmword_586C40));
do
  ClassName[v0++] ^= 0xE6u;
while ( v0 < 0xD );
v1 = _mm_loadu_si128((const __m128i *)WindowName);
WndClass.lpszClassName = ClassName;
_mm_storeu_si128((__m128i *)WindowName, _mm_xor_si128(v1, (__m128i)xmmword_586C30));
LOWORD(i) = RegisterClassW(&WndClass);
if ( (_WORD)i )
{
  hWnd = CreateWindowExW(0x300u, ClassName, WindowName, 0xCF0000u, -300, -300, 200, 200, 0, 0, 0, 0);
  for ( i = GetMessageW(&Msg, 0, 0, 0); i; i = GetMessageW(&Msg, 0, 0, 0) )
  {
    if ( *(_BYTE *)(__readfsdword(0x30u) + 2) )
    {
      v3 = _time64(0);
      srand(v3);
      if ( (unsigned int)(rand() % 100) <= 0x32 )
        exit(0);
    }
    TranslateMessage(&Msg);
    DispatchMessageW(&Msg);
```

RegisterClassW注册一个窗口类，以便在调用CreateWindow或CreateWindowEx函数时使用。CreateWindow只是将某个WNDCLASS定义的窗体变成实例。 WndClass中的成员lpfnWndProc指向一个回调函数，是窗体的消息处理函数，回调函数格式如下。

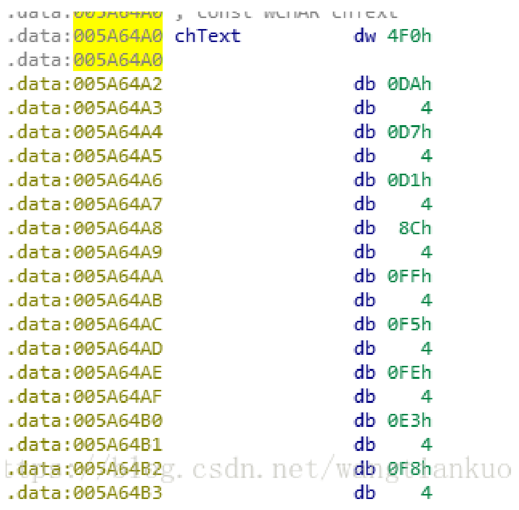
```
LRESULT CALLBACK WindowProc (
  _In_ HWND hwnd,
  _In_ UINT uMsg,
  _In_ WPARAM wParam,
  _In_ LPARAM lParam
);
```

会把输入的字符给wParam，字符个数给lParam。

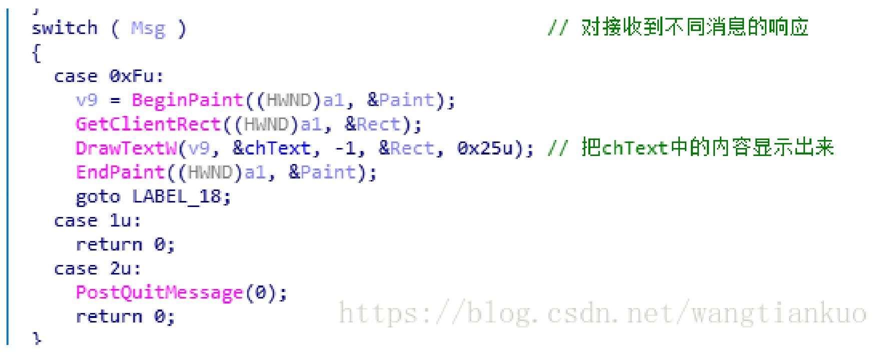
查看消息处理函数4027F0



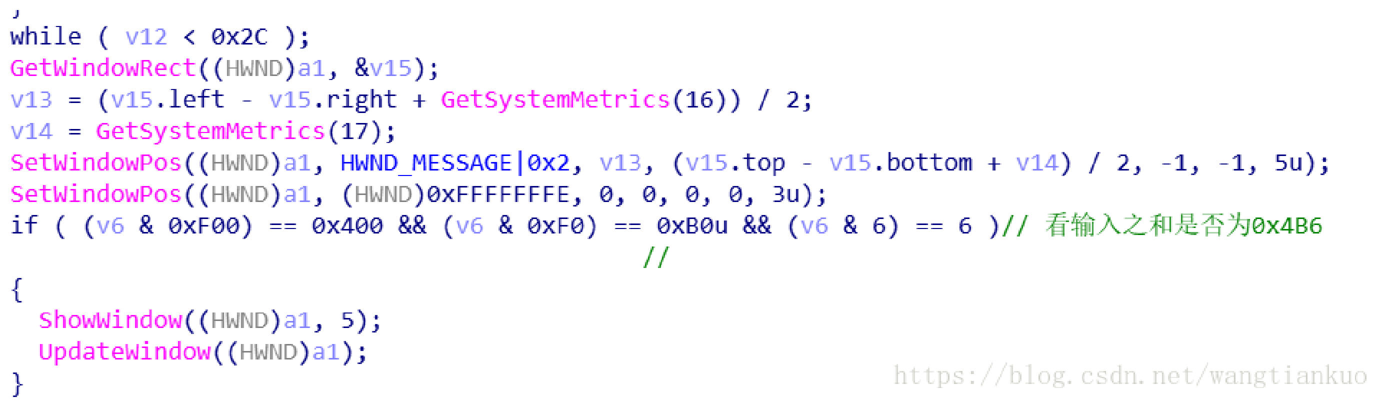
```
  do
  {
    v11 = *(_WORD *)(wParam + 2 * v10++);     // 输入的数据+2*序号（输入数据在其数组中的序号），然后进行累加，得到v6
    v6 += v11;
  }
  while ( v10 <= lParam );
  v12 = 0;
  do
  {
    *(const WCHAR *)((char *)&chText + v12) ^= v6;// v6与chText中的数据进行异或
    v12 += 2;
  }
  while ( v12 < 0x2C );
```



```
.data:005A64A0 ; const WCHAR chText
.data:005A64A0 chText          dw 4F0h
.data:005A64A0
.data:005A64A2                 db 0DAh
.data:005A64A3                 db    4
.data:005A64A4                 db 0D7h
.data:005A64A5                 db    4
.data:005A64A6                 db 0D1h
.data:005A64A7                 db    4
.data:005A64A8                 db  8Ch
.data:005A64A9                 db    4
.data:005A64AA                 db 0FFh
.data:005A64AB                 db    4
.data:005A64AC                 db 0F5h
.data:005A64AD                 db    4
.data:005A64AE                 db 0FEh
.data:005A64AF                 db    4
.data:005A64B0                 db 0E3h
.data:005A64B1                 db    4
.data:005A64B2                 db 0F8h
.data:005A64B3                 db    4
```



```
switch ( Msg )                          // 对接收到不同消息的响应
{
  case 0xFu:
    v9 = BeginPaint((HWND)a1, &Paint);
    GetClientRect((HWND)a1, &Rect);
    DrawTextW(v9, &chText, -1, &Rect, 0x25u); // 把chText中的内容显示出来
    EndPaint((HWND)a1, &Paint);
    goto LABEL_18;
  case 1u:
    return 0;
  case 2u:
    PostQuitMessage(0);
    return 0;
}
```



```
while ( v12 < 0x2C );
GetWindowRect((HWND)a1, &v15);
v13 = (v15.left - v15.right + GetSystemMetrics(16)) / 2;
v14 = GetSystemMetrics(17);
SetWindowPos((HWND)a1, HWND_MESSAGE|0x2, v13, (v15.top - v15.bottom + v14) / 2, -1, -1, 5u);
SetWindowPos((HWND)a1, (HWND)0xFFFFFFFE, 0, 0, 0, 0, 3u);
if ( (v6 & 0xF00) == 0x400 && (v6 & 0xF0) == 0xB0u && (v6 & 6) == 6 )// 看输入之和是否为0x4B6
                                                                      //
{
  ShowWindow((HWND)a1, 5);
  UpdateWindow((HWND)a1);
}
```

程序会对每一个输入做累加和异或，计算完之后，会判断一下输出+i*2累加之和是否为0x4B6，如果是，则显示窗口。写程序，将chText与0x4B6异或，得到FLAG。

```
#include "stdafx.h"

#include <Windows.h>

int _tmain(int argc, _TCHAR* argv[])

{

    int chText[]={0x4F0,0x4DA,0x4D7,0x4D1,0x48c,0x4ff,0x4f5,0x4fe,0x4e3,0x4f8,0x4e7,0x4ff,0x4e3,0x4e9,0x4f0

    for (int i=0;i<22;i++)

    {

        printf("%c",chText[i]^0x4B6);

    }

    system("pause");

    return 0;

}
```

flag{ICHUNQIU_FE362DBE}

这两天忙着准备比赛，没时间完成任务了，唉，从6月份就开始说7月份比赛，但是一直都偷懒没准备（因为只是替补）结果happy了，替补上场了，下次无论在什么位置，都要好好准备。临阵磨枪，ctf的题真的好难啊，比样本可难分析多了。