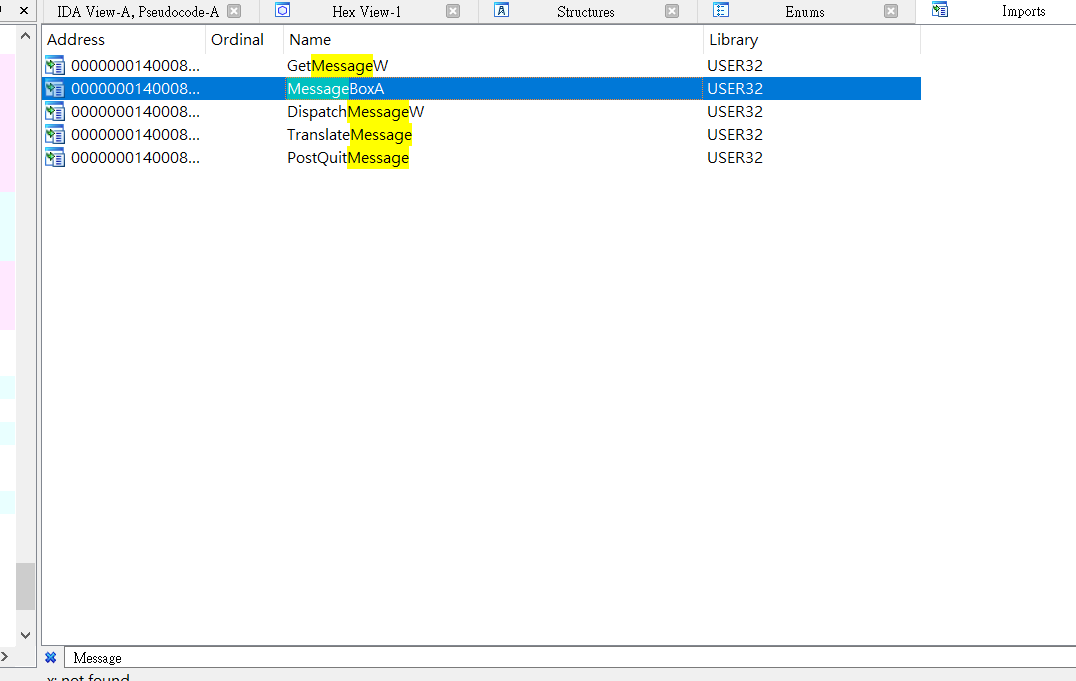
**OOXX**

這個程式發現是一個圈圈叉叉的遊戲，玩起來營不了，所以用ida pro 開啟，發現每次遊戲結束後，都會有一個跳出的訊息視窗，查了後發現該功能很可能透過Winodows api提供的MessageBoxA

有關所以試試看在import的地方是否找的到這個api，發現這支程式果然有用



之後再查詢其在main程式中使用到的地方，發現主要在一些if else中判斷，很有可能就是根據玩出不同結果，而給對應不同的訊息視窗邏輯。

一張含有 文字 的圖片

自動產生的描述

在仔細看懂這部分邏輯後發現，140001871的這步是關鍵，因為我們不可能贏過，程式敵人所以上面圖片sub\_140001640()判斷贏的函式永遠都是false，導致1871這步jz永遠都會跳到loc\_14000194D的地方，

而在它下面接續的CODE其實很可能就是取得FLAG的MessageBox。所以我使用x64dbg找到1871的位置

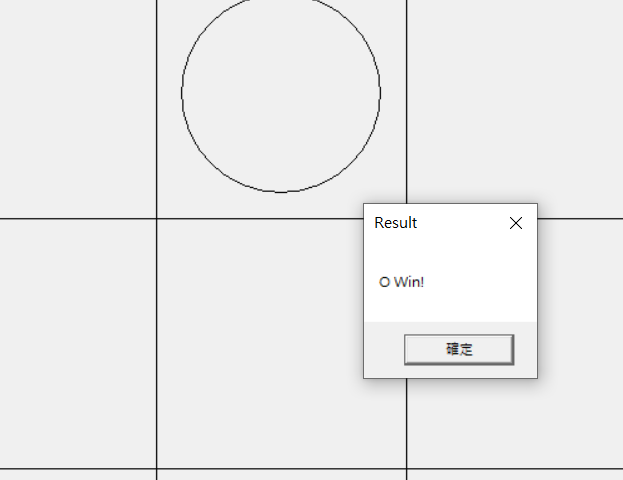


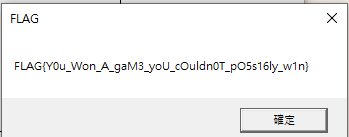
將其patch成nop，讓它不要跳過它之後的code

一張含有 桌 的圖片

自動產生的描述

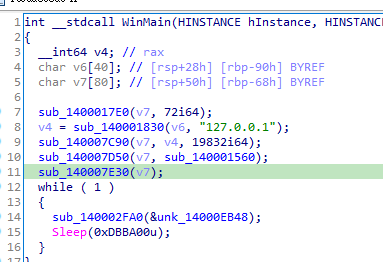
之後在x64dbg，在按下一個圈後，成功拿到flag。



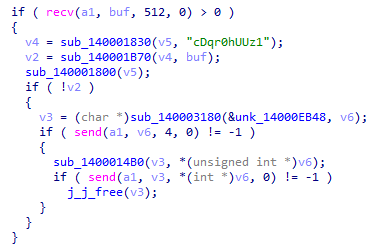


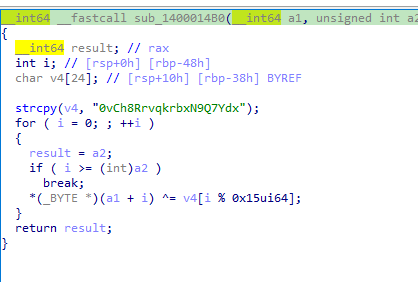
**SSTrojan**

從main funtion裡面的7E30可以知道這個程式會跟localhost:19832建立連線。



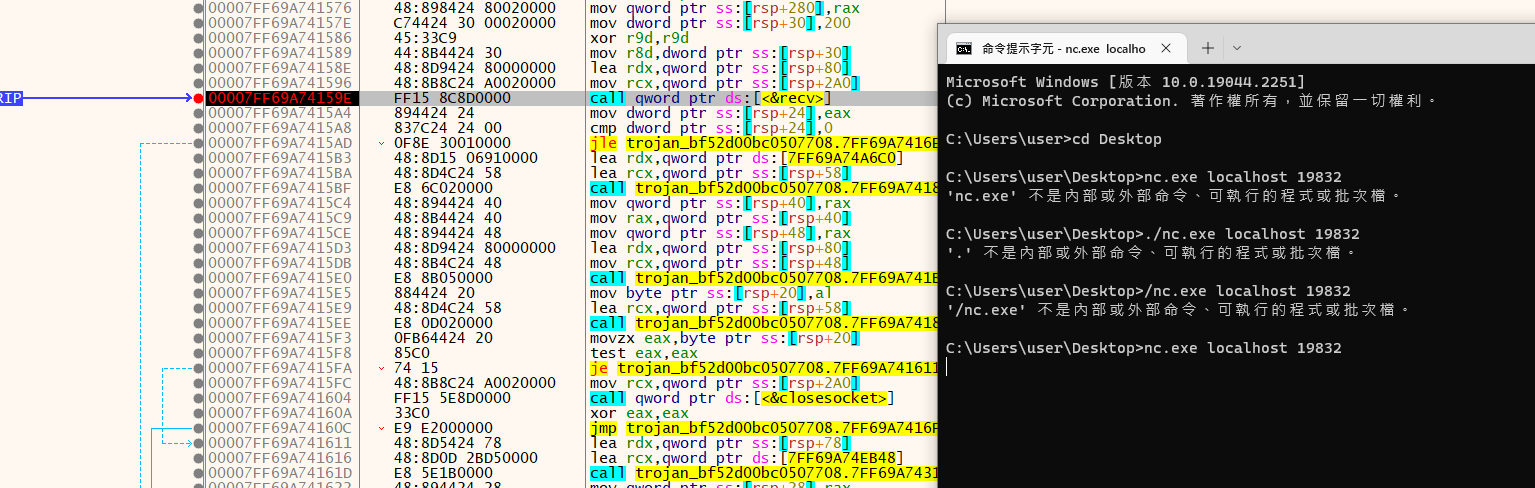
在1560裡面發現unk\_14000EB48為一PNG且經過加密後傳到本機。



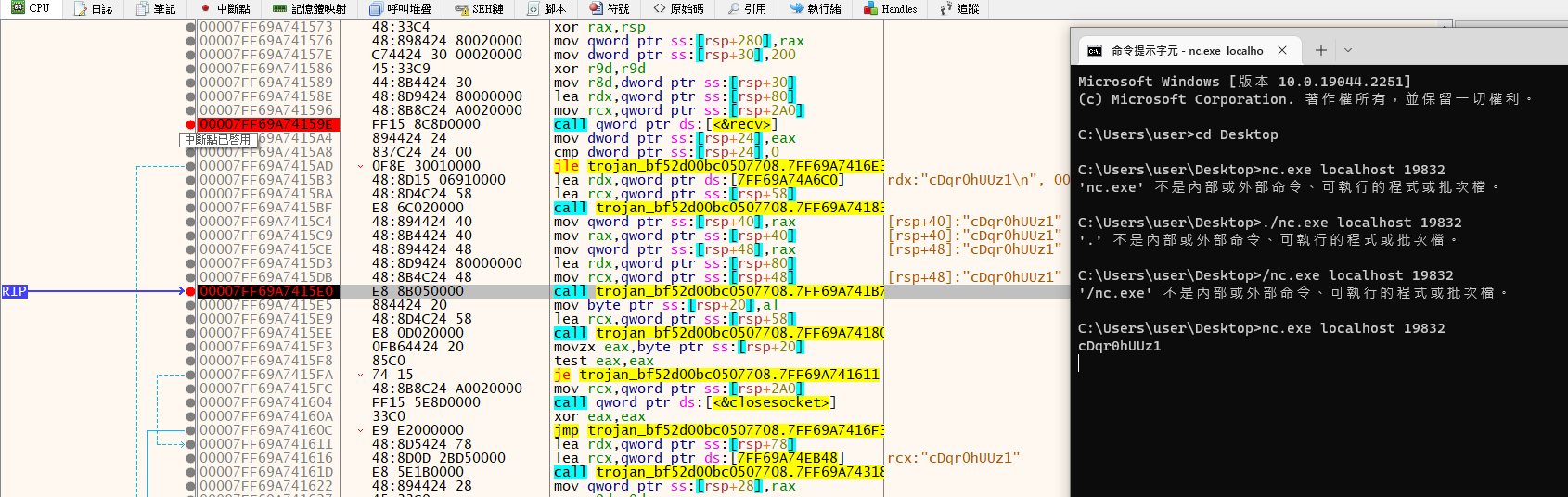


對圖片進行加密

再用x64dbg先連上並設斷點在recv

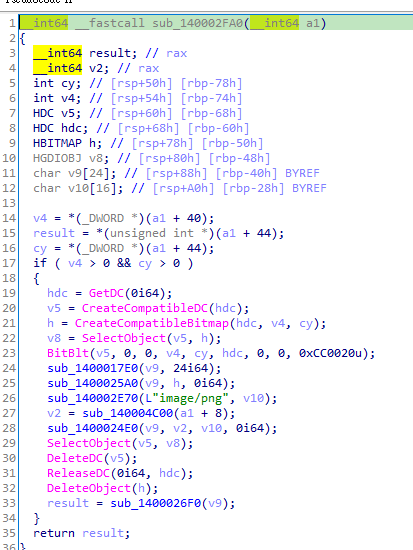


傳入比較相符的值後

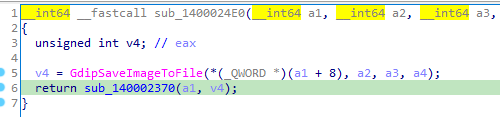


將je的判斷是改成jmp這樣就不會被closesocket並且永遠執行下面send的指令，且可以看到都是加密過後的亂碼。

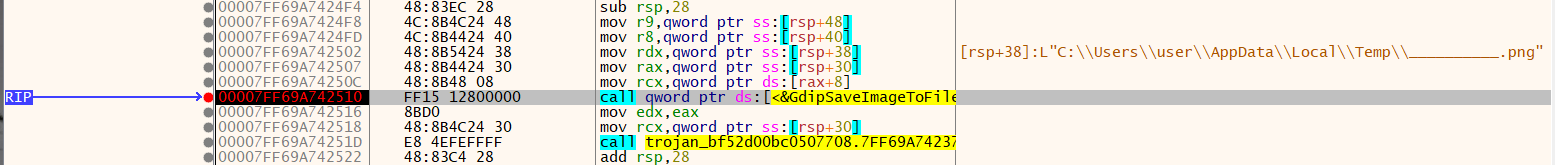


在main function的2FA0裡在進行螢幕截圖的程式，所以就得知PNG檔為螢幕截圖。

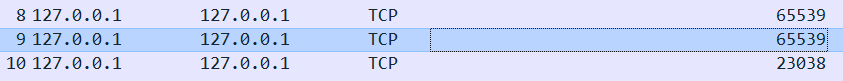
在24E0中知道會存在我們的電腦裡

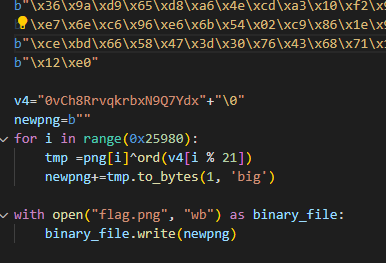


並用x64dbg到GdipSaveImageToFile設斷點可以知道他把圖片存在右邊的那個位址，然後點開後即為我們的螢幕截圖。



最後我用題目給的Pcap檔裡傳輸的Data紀錄去作解密





得到flag

(好想要apex cheater…..)