

B05901182 電機四 潘彥銘

- 解題思路:

- 靜態分析

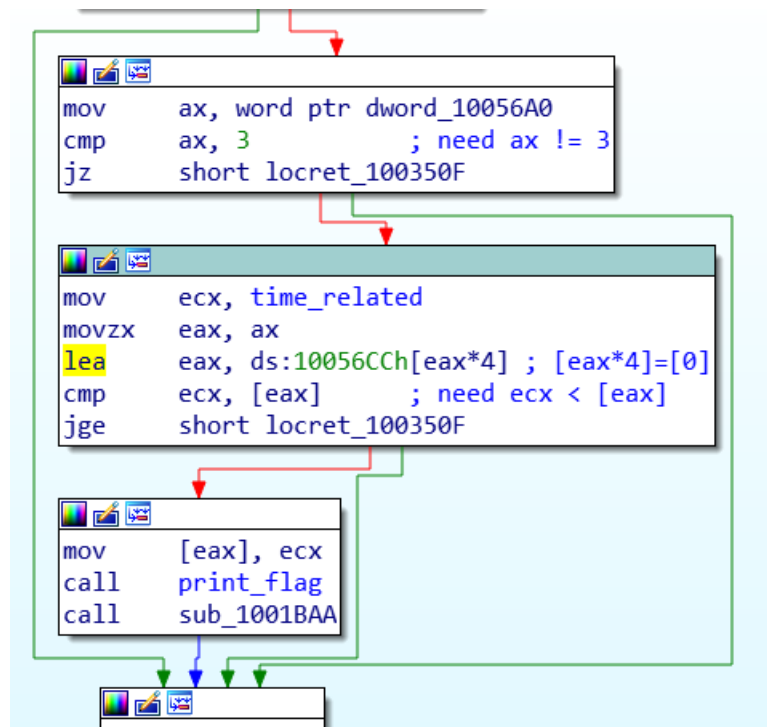
1. 用 IDA pro 打開 Winmine.exe, 在 start()裡檢視每個被呼叫的 function, 其中 sub_10021F0 裡面跟改變 window 相關的 function, 且呼叫完 sub_10021F0 後整個 exe 檔看起來算是執行完成了, 因此猜 sub_10021F0 是 main function。
2. 檢視 sub_10021F0 裡的 function, 首先上網查 CreateWindowExW, 發現 call 這個 function 之前會先 call RegisterClassW, 因此再 google RegisterClassW, 發現裡面有個 pointer 指向 WNDCLASS 這個 structure, 看一下 WNDCLASS 發現還要再看 WindowProc, 這個 WindowProc 會處理傳給 window 的訊息。然後發現 sub_10021F0 裡有和 WindowProc 對應的東西, 叫 lpfnWndProc, 並發現透過 mov 從 sub_1001BC9 到[ebp+WndClass.lpfnWndProc], 因此看一下 sub_1001BC9 在幹嘛。透過檢視 sub_1001BC9 裡的變數我們可以確定他就是 WindowProc。
3. 由於剛剛說 WindowProc 會處理傳給 window 的訊息, 因此我們想知道按滑鼠左鍵會整個程式會怎麼運行, 透過助教投影片附的網站我們可以知道跟滑鼠左鍵相關的指令 hex 碼是 0x202。
4. 點進 WindowProc 後, 此時 edx 是 0x202, 比 esi 大, 又把 edx-201 後存進 eax, 然後發現 eax 比 11 小, 這時候運行到 byte_10021DE 這個 array 裡, 且 byte_10021DE[eax]回傳 1, 並把 1 mov 到 eax 裡。再來看到 off_10021C2[eax*4], 當把 eax*4 傳入後會對應到 loc_1001FDF, 因此可以知道 loc_1001FDF 是拿來處理按下滑鼠左鍵後會觸發的訊息。

- 動態分析

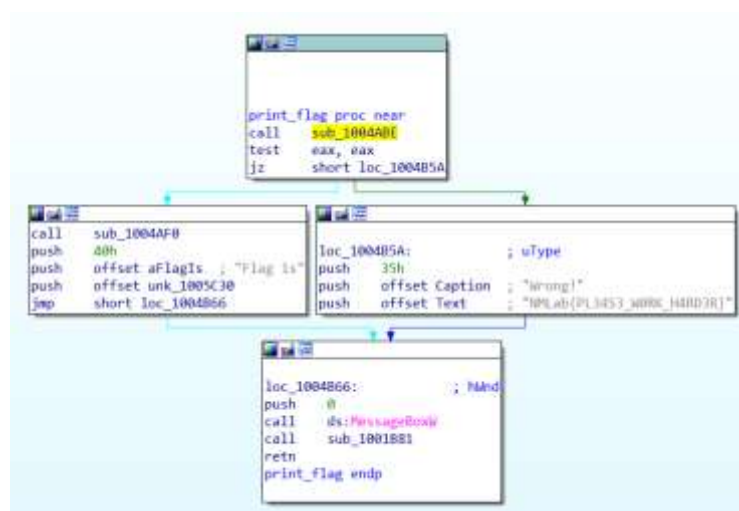
1. 用 x32dbg 打開 Winmine.exe, 在 10010FDF 設斷點, 看按下滑鼠左鍵後會發生什麼事。當執行到 jz loc_10021A9(也就是 01001FE7)時會有兩個方向可以走, 透過 x32dbg 來追蹤程式走哪一條路。
2. 追蹤 x32dbg 記憶體位置的變動, 發現他走了紅色那條路來到 01001FED, 繼續追蹤發現來到 01002005, 並 call 了一個 function, 我們可以知道他是處理按下滑鼠左鍵後續動作的 function。
3. 接下來我們知道按下右鍵後, window 理應出現地雷或是數字等動作, 因此我們繼續用 x32dbg 追蹤。
4. 透過追蹤我們得知 01001FDF 可能是處理格子下壓的動作, 0100382B 可能是跟聲音有關的 function, 01003512 裡面 call 的 0100304D 可能是處理格子顯示數字、0100358C 可能是處理顯示地雷的 function, 010038BC 是改變臉部表情的 function, 010035AB 可能是處理遊戲結束前的 function(印全圖地雷)
5. 透過 dbg 右欄檢視各變數儲存的值我們可以知道 eax 存 y 座標, esi 存 x 座標, 以及 dbg 下欄得知有處理 mine map 的 function
6. 上述步驟大致分析完了整個 Winmine.exe。

- 找 Flag

1. 透過 IDA pro 檢視 string 的功能，發現了一個'Flag is'的字串，從這裡往前追蹤，看程式遇到什麼樣的條件才會運行到'Flag is'這裡。
2. 發現 0100350F 這個 function 中會使用到有'Flag is'這個 string 的 function(簡稱 print_flag)，擷取 0100350F 的部分(如下圖)。



3. 透過追蹤，我們發現 $ecx < [eax]$ ，程式才會進入 `print_flag`，利用 dbg 把 `ecx` 的值直接設為 0。
4. 進入 `print_flag` 後(如下圖)，檢查 `sub_1004ABE` 這個 function，發現要讓 `eax=1`，程式才會往左走印出正確的 flag。此外 `sub_1004ABE` 裡面是檢查 01004EFC 跟 01005C00 兩個記憶體位置儲存的東西是否相同，因此透過 dbg 把 01005C00 儲存的值改的跟 01004EFC 一樣。



5. 執行到 call MessageBoxW 那行就會印出正確的 flag 。

| 資料視窗 1 | 資料視窗 2 | 資料視窗 3 | 資料視窗 4 | 資料視窗 5 | 監視 1 |
|----------|-------------|-------------|-------------|-------------|------------------|
| 位址 | 十六進位 | | | | ASCII |
| 01005C00 | 00 00 00 00 | 9D FF 16 FF | 56 FF B6 FF | 76 FF 7B FF |y.yvytyvyty |
| 01005C10 | BA FF 3C FF | DE FF 9E FF | 9C FF 7E FF | 5D FF D6 FE | °y<ypy.y.y~ylyöp |
| 01005C20 | D7 FE DC FE | 00 00 00 00 | 00 00 00 00 | 00 00 00 00 | xpüþ..... |
| 01005C30 | 4E 00 4D 00 | 4C 00 61 00 | 62 00 7B 00 | 59 00 6F 00 | N.M.L.a.b.{.Y.o. |
| 01005C40 | 75 00 5F 00 | 61 00 72 00 | 65 00 5F 00 | 67 00 6F 00 | u._.a.r.e._g.o. |
| 01005C50 | 6F 00 64 00 | 5F 00 61 00 | 74 00 5F 00 | 6D 00 69 00 | o.d._.a.t._m.i. |
| 01005C60 | 6E 00 69 00 | 6E 00 67 00 | 5F 00 3A 00 | 44 00 7D 00 | n.i.n.g._.:D.} |
| 01005C70 | 00 00 00 00 | 00 00 00 00 | 00 00 00 00 | 00 00 00 00 | |
| 01005C80 | 00 00 00 00 | 00 00 00 00 | 00 00 00 00 | 00 00 00 00 | |
| 01005C90 | 00 00 00 00 | 00 00 00 00 | 00 00 00 00 | 00 00 00 00 | |

Flag is



NMLab{You_are_good_at_mining_:D}

確定

- **Flag:**

NMLab{You_are_good_at_mining_:D}