Day 5 - Reverse Engineering

各種 reversing 實例經驗談 - Aaron Luo

Outline

- 不會講述控制EIP, RIP 這類邪惡的技巧 XD
- 絕對正派, 課程目標"Make Your Life Easier"
  - Microsoft Office 彈出試用期到了的框框？
  - 打 Online game 贏不了台幣戰士？
  - 忘了帶軟體的 USB Dongle 是否困擾著你？
  - 想加入別人 program 功能到自己的 Program 但他沒有 Open source？
  - 鍵盤加密晶片讓你的 keylogger 失效了嗎？
    - 監控你的*小孩*
  - 不希望看到討厭的人上 ggc？

AWEIMEOW *絕對正派！*
YURILOVE… *監控小孩...__*
OWEN L *果然正派*
ИССЛЕДО… *監控小孩！！！！！*

How to bypss program registration?

```
if ( registered) //find out the compare point
{
   do someting
}
else
{
   do something
}
```

Demo
破解 iOS 雜誌軟體的付費限制
用 GDB 載入 iPhone 的 program
先把 APP 從 Memory Dump 出來(丟IDA
　預設狀況有加密, run 起來後可以解析他的 code
會有類似 "isfree"

通常判斷的地方之後會有一個跳轉
跳轉的地方下斷點，改變執行流程
直接下斷點會無法 access memory
因為 iPhone APP 會有 ASLR 動態載入 Memory 所以會跟靜態分析的結果不一樣

王茂林 *https://github.com/stefanesser/dumpdecrypted*

寫一個程式去搜尋，每次疊加一個 page 的大小（4kb）
從 2222 搜尋到 100000 的位置為止
找到真實被 ASLR 塞到的記憶體位置，之後就可以使用 GDB 載入
下斷點之後再次點訂閱（付費功能）會被斷點攔下來，使用指令看組語
ARM 切換為 Thumb 的 code 這邊因為顯示的是 ARM 的 code 所以是會有問題

HSIEH P *看不清楚指令 OTZ*

程式會確認兩次，下載跟訂閱都會確認，兩邊都修改之後就破解成功了
這樣的破解邏輯可以應用在很多的地方，像是把微軟 Office 的框框（nag 框）去掉

How to avoid the NAG window ?(M$沒註冊會跳這種NAG框

- Ollydbg - CTRL + F9 * N 大法！(直到RET才停
- 原理

```
if ( not reggistered) {
    Show NagDialog();
    // if you press CTRL + F9, after window close will return to here
}
```

Ollydbg Load Word and Exec
看到 NAG 之後按暫停，連續按多次的 CTRL + F9（執行到返回）
把框結束掉就會斷點在他返回的下一行
停下來的就是他處理程序的下一行，這時候就可以開始找判斷條件的位置
CTRL + F9 把流程簡化，然後再按一次
看到 jnz 有時候就是判斷點，跳過就好，不過微軟沒那麼簡單
所以再按一下，看到一個跳躍的點下 breakpoint 觀察
微軟還會對程式進行 check sum 的檢查（檢查程式完整性是否被修改）
INT3 在 Memory 的 code 是 0xcc 所以說你下 breakpoint 他只要跟原本的 Memory 檢查是否一樣就可以知道是否有下斷點

Avoid INT3 Check

- Use the Hardware Execution Breakpoint.（不會在 memory 留下int 3指令

Word 裡面有使用 Windows 的 ASLR，那可以透過竄改 PE 結構來修改

DBCharacteristics 中的 DLL can move 勾勾取消 overwrite 另存

好處是每次 DLL 在 Memory 裡面的位置都會一樣

比較方便 DBG 相對於去修改 ImagePath 等等
改好之後我們再開啟一次
出現 check sum 檢查錯誤，因為有三個 breakpoint，可能有一個以上被檢查到
找一個試試看，下hr指令
hr 指令在Ollydbg為 hr = hardware on read breakpoint(硬件訪問斷點

restart program 發現沒有斷下來，可能判點在其他的，那再到另外幾個 breakpoint 試試
hardware breakpoint 一次只可以下四個，現在法剩下的也下斷點，好，被斷下來了
會發現他的 code 都是在做計算的，看附近有很多 constant 存在就很有可能在計算東西
計算發現值不對就會 return error
一般在做 trace 的時候有一個通用的技巧是：不要走回頭路

General Analysis Skill

- Don't get back (不要走回頭路)

因為一般程式都是從上而下執行如果往回跳的就不要讓他跳回去，一般來說會往回跳都是迴圈，所以遇到時就在下一行下斷點讓他繼續走

這時候再按 F8（不進 CALL）向下執行，這時候就會跳出 ERROR 所以就知道是這邊在檢查
把它 NOP 掉之後下次修改其他位置應該就不會再跳出 ERROR 了
Ollydbg 可以直接把修改過後的執行檔存檔

- 右鍵 --> Copy to Executable --> Selection
- 再按一次右鍵 --> save

可以按到這一次開啟後就正常沒有跳到 Exception 的地方
之後就繼續來看我們剛找到可能是 NAG 的判斷點
把他修改掉之後應該就不會跳 NAG 窗出來，可以看到修改之後不會跳出 NAG 框就可以正常使用了
把它另存回去剛的執行檔，現在 WORD 可以正常的使用了！

How to hacking a game?

- Simulate keyboard / Mouse Input
- Modify Memory
- Modify Packets
- Modify Program

DEMO

典型 MMORPG 的角色扮演遊戲
進入遊戲可以看到有各國（韓國人、中國人）已經上線了

HSIEH P *中國人常常叫做抗韓什麼的*
不過玩遊戲就是要帶點情緒嘛

通常這種遊戲我會想找人物的座標、移動速度等等
通常找到座標就可以任意移動所以很方便
使用 IDA Pro 來找 Strings 有時候開發人員為了方便會把 debug 資訊也編進去遊戲
所以常常可以看到一些奇怪的資訊
程式如果在 dbg mode 會 print 出一些給開發人員看的東西
可以直接在 debug flag 下 breakpoint 下斷點，這邊我直接把值改成 1
出現了只能給開發人員看的資訊，pos、state 什麼的
分別尋找座標的三個值在記憶體中的位置
看 IDA Pro 中 pos 的位置是如何寫入的
看起來是某種類似 printf 的東西，先把它寫入 list 再 call api
找到填入的方式就可以找到寫入的位置
透過 gdb 把找到的值 print 出來確認，跟遊戲內一樣！
之後就可以來修改看看，有 X,Y,Z 三個軸，把 Z 軸改成 1000，就會飛到空中
隨便按一個鍵就可以.. 讓角色摔死！
總不能寫一個外掛就是讓自己死啊
寫一個小程式，透過類似 Windows DLL Injection 的方式
把它注入到手機裡面，遊戲裡面會多一個黑點，點了之後

HSIEH P *網路炸惹，DEMO 也炸惹*

AWEIMEOW *所以把影片放在本地是好的選擇 (X*

這個外掛可以任意的修改三個軸
按下去之後會印三個 bar 到銀幕上
快被怪物 K 死就可以飛到半空中躲避
最好不要再人多的時候降落，不然容易被檢舉

看看小地圖就可以快速移動惹
https://www.youtube.com/watch?v=9rw9K372bkE
然後.. 又摔死惹
寫外掛就是要找到可以控的記憶體，想辦法去操控他
然後就是要讓他好操作
通常就是寫 DLL 去弄

Gerneral Analysis Skill

- Search Memory and find out whole procedure change it.
  - Cheat Engine is good for this

DEMO
飛機躲子彈的遊戲 (特訓99!!)

HSIEH P *其實還蠻好玩的*
我必需讓自己死掉，不然沒有時間惹..

子彈有 57 顆，先用 cheate engine 搜尋 57
之後打到子彈剩 50 顆再次搜尋
找到之後把他修改成 0
Ya 沒有子彈了，可是這樣無法抵達計分板，所以改個一顆好了
也可以把他改成255顆，這樣就必死無疑了
如果去 trace 他的 memory 也可以找到無敵之類的
額...

HSIEH P

Never Live Demo

這個外掛可能有些 bug 讓我現在再改一次好惹
這外掛是我八、九年前寫的 code 所以很髒，這邊有幾個 pointer
這邊給他個 JMP 你看，這樣他就完全不怕子彈了

透過遊戲學習 Reversing 是比較有趣的
遊戲變化多端，可以透過來學習各方的技巧
投影片在介紹 Reversing 的各式特色，可能大家看了有興趣可以回去再研究

How to Create Memory Hack

- OpenProcess
- ReadProcessMemory / WriteProcessMemory

Create a Runtime Loader

- Use PyDBG
- Use Windows API
  - WaitForDebugEvent
  - ContinueDebugEvent
- DLL Injection

HSIEH P *可以讓 DLL 跟程式共享同一塊 Virtual Memory 這樣就可以直接修改一些記憶體*

- Pin Tool

HSIEH P *Intel 的工具，到 instruction 會有一個 callback 這樣你就可以去 trace 一些狀態*
有些遊戲會去偵測有沒有 dbg flag
可是 Pin tool 是基於 CPU 架構，所以不會被偵測到
Anti-Anti-Anti-Anti-DBG 是比較好的

破解鍵盤加密機制

奧樂舉辦駭客擂台
鍵盤輸入都會經過他的 IC 透過時間之類的去 Encrypted
所以通訊時會被加密
Hook 大量 Windows API 所以取得的資訊大部分會是錯誤的
晶片很小（大概就 ethernet2thunderbolt 的大小？）然後這邊是 USB，他接到電腦上

HSIEH P *他的程式的圖跟 hello kitty 一樣沒有嘴巴，可是他更慘，連鼻子都沒有，現在是七月，有點恐怖*

KeyLogeer 採用的方式

- GetAsyncKeyState
  - 不需要DLL

- WH_KEYBOARD Hook
  - USER32.DLL 去做交換，Hook 他裡面交換的點，輸入東西時就會被攔截到
- IME Hook (*Input method editor*)
  - 可以抓到中文
- IDT Hook
  - 寫驅動程式
  - 透過 interrupt instruction table 去攔截

透過加密晶片我輸入 abc 他會變成其他的東西

Keyboard > (Scan Code) > Keyboard Device Driver > (Message)
> System message queue > (Message) > Thread Message Queue > ()

(Thread message loop)
Window Pre..

先開測試程式，用 Ollydbg 載入
HSIEH P *（破音～～）*
所有 User Message 都會在這邊被攔下
Windows 有很多 Message 像是滑鼠經過、鍵盤按下等等會進入queue
Message code 可以參考 MSDN

- WM_KEYDOWN (鍵盤按下去的瞬間都會 call 到的)
- https://msdn.microsoft.com/zh-tw/library/windows/desktop/ms646280(v=vs.85).aspx

OllyDBG 有 Conditional Breakpoint 可以設定條件攔下來
他會把所有的 code 列出來
這邊跳了一個錯誤，是因為載入了一個外部 DLL
這個 DLL 可能有加殼所以會 Anti-Debug
脫殼怕會有 CRC 太麻煩
有個 plugin 叫 stealth64 我們使用全勾（Hook ALL）來 Ant-Anti-Debug 就不相信他比這個還厲害
好，現在成功了
這邊主要是要測試 WM_KEYDOWN
現在這邊設定 100
按個按鍵應該就會停下來
輸入 A 卻回傳 9，所以可以知道這個加密晶片有處理掉，所以 Hook 他沒有用
有一個有東西叫做 WM_CHAR
https://msdn.microsoft.com/zh-tw/library/windows/desktop/ms646276(v=vs.85).aspx
這邊一樣去 hook 他，就成功了，可能處理不是很完善
進制轉換小工具 HexDecChar

很難知道 windows 是在哪裡接收 user message
Hook user32.InternalCallWinProc 就可以讓鍵盤加密的機制完全瓦解掉
使用 CheateEngine 寫了個小 Script 去注入目標 Process
如果是 WM_CHAR 就攔截，並儲存至 [haha] 裡面
現在我們輸入 a

HSIEH P *啊，我按成 q 了*
AWEIMEOW *講者好有趣 XDD*

現在可以看到 113，有完整的攔截到
Hook InternalCallWinProc 判斷 module 是否為 OK1001.dll（加密晶片透過此傳遞）

HSIEH P *有人說 Live Demo 會遇到一些像魔咒一樣的事情*
本來測試都是 Work 的，可是現在不 Work 了
就當作你們有看到就好了啦
SYA *Never live demo*
HSIEH P *讓我們進到下一個階段*
... 額，出了點小狀況
可能是剛剛 hook 爆炸了所以電腦有點怪怪的
（當機中）

Global Hook 會將 DLL 注入到每個 Process 裡面，本來是只彈一個 Hook OK! 沒想到現在.. 一直彈一直彈

DLL hijacking型惡意程式分析 & 提取ShellCode並編寫Loader

什麼是 DLL Hijacking?

- 利用DLL裝載器的規則
- 駭客讓原先應載入的DLL，改載入自己的DLL
- 如下圖所示 dnsapi.dll (net cat)

Windows DLL裝載器的搜尋順序

- 程式所在資料夾
- System資料夾（GetSystemDirectory）
- Windows資料夾（）

BRUCE C *http://yonsm.net/aheadlib/*

將惡意DLL加入預載DLL可使得惡意DLL被執行
若某 process 是高權限使用者執行，可能可以拿來提權

惡意程式主要有兩個檔案

- nslookup.exe

- DNSAPI.DLL

使用 InsidePE 觀察 nslookup.exe的import table可以發現，import table會載入 DnsFreeConfigStructure、DnsQueryConfigAllocEx 這兩個 API

先看 DllEntryPoint

PEB->LDR->InMemoryOrderHmodule
DLL_ATTACH 判斷載入
在 Module base 透過動態載入各個 API 的位置

分析動態取得的 API function
VirtualProtect 修改記憶體屬性，可讀寫可執行 .text區段 只有讀取和執行
API 主要針對 Reg 跟 File 跟 Memory 而已
修改 Entry Point 的 function

分析修改完的 OEP，判斷是否為正確的 nslookup.exe
解密 shellcode (xor 0x63)
return shellcode
自我啟動並在註冊表內註冊自動啟動

使用 Ollydbg 分析 Shellcode 並且提取執行
解出 ShellCode 更可迅速分析惡意程式，並且可自己建構 Shellcode_Loader 來載入，不用依賴原本的惡意程式

DLL_Base+0x149D 修改 OEP 的地方下 Hardware On Execute 的斷點，並執行程式

這邊是一個把 shellcode 做 copy 的動作
單步執行發現 OEP 已經被修改
右鍵 New origin here 你的 Eip 就會被修改至這邊
判斷是否為 nslookup.exe 要 patch 掉
停在 shellcode decrypted 的地方
透過 plugin 可以直接把它取出來

Wireshark 可看出他會連到 help.2012hi.hk 443 port
此外解出 ShellCode 後發現該 ShellCode 有組很特別的字組 “VoqA.14" 把該這個特殊字組拿去 Google，可發現這是 Poison IVY 木馬所產生的 shellcode

ПОСЛЕДО… *看到現在...黑黑無誤*

秋馨 羅 *說好的正派呢(伸手*

ИССЛЕДО… *calc.exe (遮*

編寫IDC Script輔助分析惡意程式

有大量的 function 塞在裡面，但是執行的地方兩邊是沒有關係的
撰寫 IDA Pro 的 Script 讓兩邊關聯起來
先看看 Assembly Code 是不是有什麼關聯存在
API Name 放到 Ebx 裡面，下面的 Call 取得 API 的位置之後 Mov 到一個變數當中
...
...
...
GetFuncName(offsetfunctionname)
那把 word 轉成 ascii 再轉成 string
...

- 用function的binary(特徵)來找function實際的位置
  - 從Hex View找出fundtion的binary，取前8 byte
  - FindBinary ->從上到下找binary有match的點，有找到的話會return 一個funtionaddr
  - parse return的function (利用API : codeoffset<GetFunctionAttr(funtionaddr,FUNCATTR_END)>)
- Message[= printf]
  - Message("%s\n",GetDisam(codeoffset));
  - MakeName(Dword(codeoffset+1),"API_"+GetFuncName(offsetfuncname));

```
static GetFuncName(dwoffset)  //把 word 轉成 ascii 再轉成 string
{
   return GetString(Dword(dwoffset+1),-1, ASCSTR_C);
}
```

惡意程式分析＆封包解密

你要有 pcap 檔
winsvcex.dll 是惡意程式主體
沒有加殼，直接 IDA Pro 分析

由於是 DLL 所以不能直接執行

CreateThread 去執行主要的程式碼
Ollydbg 並 call export function
下斷點可以讀到本來的設定
看到解密跟 config file 的地方
把 .data 放到 ollydbg 的 memory dump 區段
找到 ip & domain & port 知道了一些 C2 的資訊
用 Http 去 request 受害者的一些資訊發送到 server 端
封包送出前有經過加密模組
看起來是 base64 其實不是，多了'__'的符號，去除後成功解碼
解密長度為 4C 與開頭的 hex 相同，故判斷前兩個 byte 是封包長度
後面的 4bytes 透過加密function 可以知道是 rand() + time() 當作 rc4 的 key
在之後的 4bytes 是 CRC32 的 checksum
知道結構資訊後就可寫解密程式了

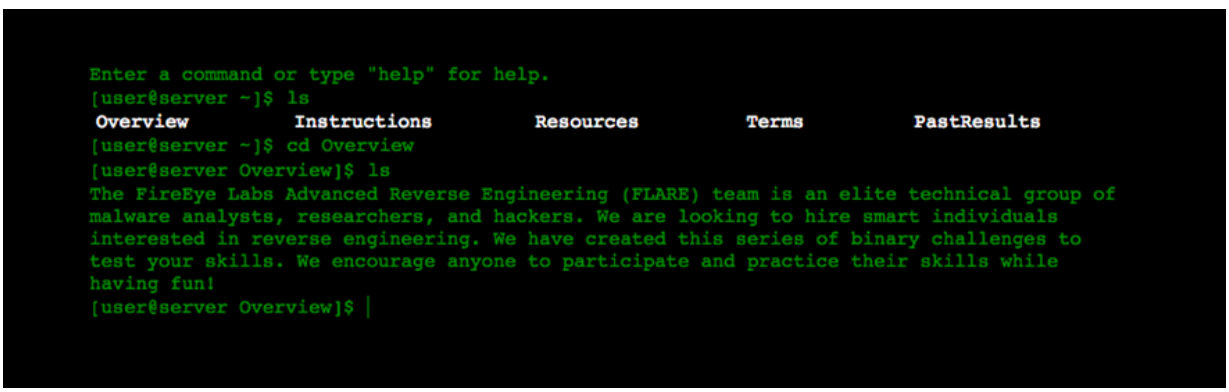
```
struct pkt_format {
    unsigned short pktlen;
    unsigned short unk_1;
    int rc4_key;
    int crc32_checksum;
}
```

寫好程式後就完成 decrypted 的動作
最後解密完可以看到駭客下的指令
End 午休

The FLARE On Challenge

http://www.flare-on.com

ПОСЛЕДО... *今年是第二屆, fire-eye公司出的.保證好玩*



```
1  BOOL start()
2  {
3    HANDLE v0; // ST18_4@1
4    signed int v1; // ecx@1
5    HANDLE hFile; // [sp+8h] [bp-8h]@1
6    DWORD NumberOfBytesWritten; // [sp+Ch] [bp-4h]@1
7
8    v0 = GetStdHandle(0xFFFFFFF6);
9    hFile = GetStdHandle(0xFFFFFFF5);
10   WriteFile(hFile, aLetSStartOutEa, 0x2Au, &NumberOfBytesWritten, 0);
11   ReadFile(v0, byte_402158, 0x32u, &NumberOfBytesWritten, 0);
12   v1 = 0;
13   while ( ((unsigned __int8)byte_402158[v1] ^ 0x7D) == byte_402140[v1] )
14   {
15     ++v1;
16     if ( v1 >= 24 )
17       return WriteFile(hFile, aYouAreSuccess, 0x12u, &NumberOfBytesWritten, 0);
18   }
19   return WriteFile(hFile, aYouAreFailure, 0x12u, &NumberOfBytesWritten, 0);
20 }
```

幫補充一下..F5下又去就出來了, 把0x402140開始的char Array丟去xor 0x7D 答案就出來了

Ollydbg:

```
0040104D    8A81 40214000   MOV AL,BYTE PTR DS:[ECX+402140]
00401053  . 34 7D           XOR AL,7D
00401055    8881 40214000   MOV BYTE PTR DS:[ECX+402140],AL
0040105B    90              NOP
0040105C    90              NOP
0040105D    41              INC ECX
0040105E    83F9 18         CMP ECX,18
00401061  .^7C EA           JL SHORT i_am_hap.0040104D
```

直接把key用in line的asm修改，
那他總共會做18次，也就是10進位的...24次
a xor

ИССЛЕДО… *共筆倒店了!?????*
AWEIMEOW *跟不上講者的速度 QQ*
ИССЛЕДО… *喔喔*
AWEIMEOW *而且有時候是用快捷鍵呼叫出來的視窗，就不知道要怎麼跟著做了 orz*
王茂林 *而且有些 plugin 不知道 囧rz*
ИССЛЕДО… *哪可以請講者提供 :)*
賺一些plugin 回家
AWEIMEOW *好希望講者能看一下 chatroom XD*
ИССЛЕДО… *可以舉手請他講慢一點或是給一點時間做筆記!?*
flare-on 只有11題. :)
馬聖豪 *全live demo中*
ИССЛЕДО… *考驗ida pro/ollydbg的熟悉度嗎 ！？*
全部解完會收到下列畫面

Congratulations on completing the Flare-On Challenge! You deserve a prize. A FLARE Team member will be in touch. I don't use the term Hero lightly, but you are the greatest Hero in the history of the internet.

-FLARE

馬聖豪 *Flare-on 考到啥時候啊...沒記錯考完不是會發解答嗎*

ИССЛЕДО… *是的.時間到了後..大概再加1-2個月, 就會出解答*

AWEIMEOW *You r the greatest HERO in the history of the history！！*

LAYS *可惡 想解*

AWEIMEOW *求 ADR 的 Live Demo 直播*

馬聖豪 *幫不上啊 我是看IDA Pro F5配Olly撈資料 完全跟不上講者講到哪XD 全Olly肉眼解有點兇*

AWEIMEOW *這邊那麼多大神，有人知道要怎麼從OD左下角黑黑的那邊拉資料出來嗎 HexDump 那邊，就是例如有 MOV eax, SS:[...] 這種東西的時候，要怎麼拉資料 QQ*

LAYS *指令上按Enter資料視窗會自動Follow，然後選起來右鍵複製到剪貼簿*

AWEIMEOW *喔喔喔喔喔喔喔喔喔喔喔喔喔喔喔喔喔 謝謝Lays*

ИССЛЕДО… *flare-on 的截止日: close on Sept. 8, 2015 20:00EDT 還有四天..大家加油..:)*

AWEIMEOW *只！剩！四！天！*

ИССЛЕДО… *4天夠了*

鄭達達 *獎品是什麼啊?*

ИССЛЕДО… *好像是硬幣!!! (token)*

馬聖豪 *共筆休息惹QQ*

ИССЛЕДО… *共筆下午沒有很多資料耶*

HSIEH P *下午全部是題目啊逐字稿也太累，而且很多是操作不好描述，大部分都在聊天室討論了*