# Reverse Engineering - 2

2021/11/26

Presented by LJP

# # whoami

- LJP / LJP-TW
- SQLab @ NYCU 碩一
- CTF @ 10sec / TSJ
- Pwner

# Outline

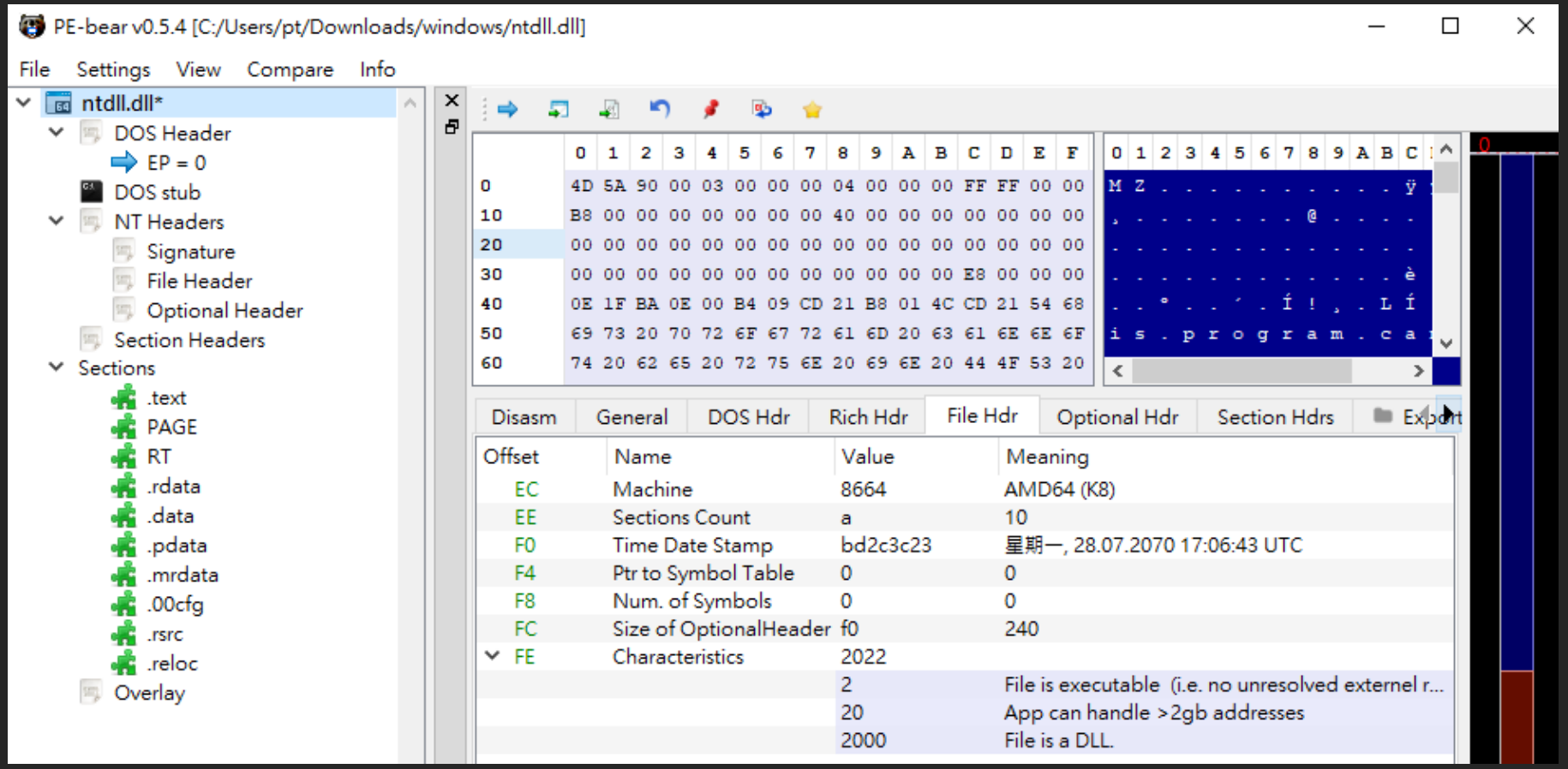
- Tools
- Where to start
- Address Space
- Speed-up
- DLL
- PE Format
- IAT
- EAT
- PEB

# Tools

PE-bear / x64dbg

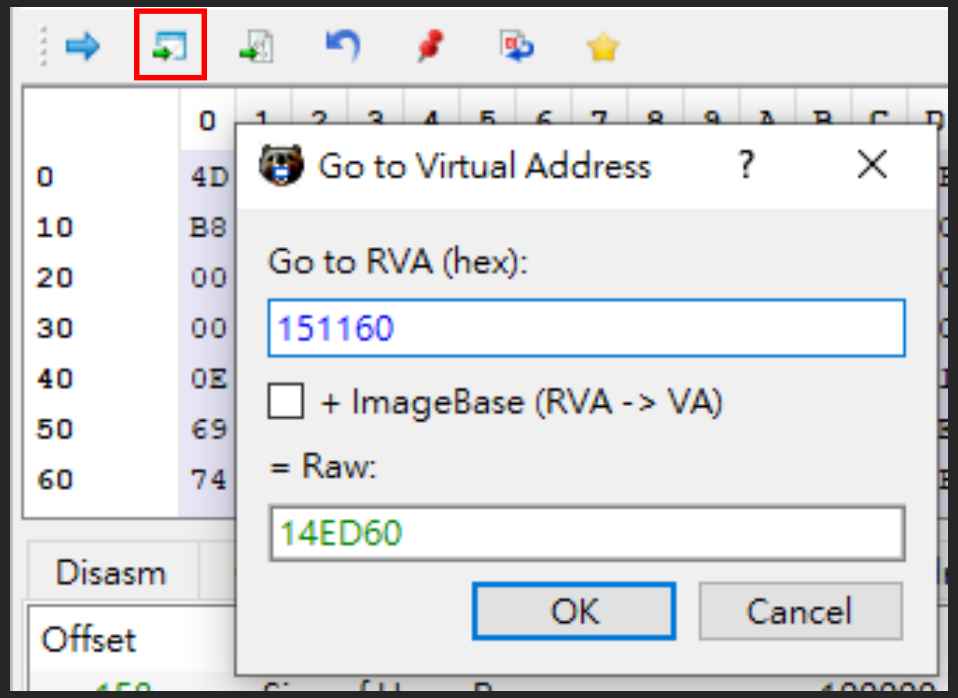
# Tools - PE-Bear

- 觀察 PE format

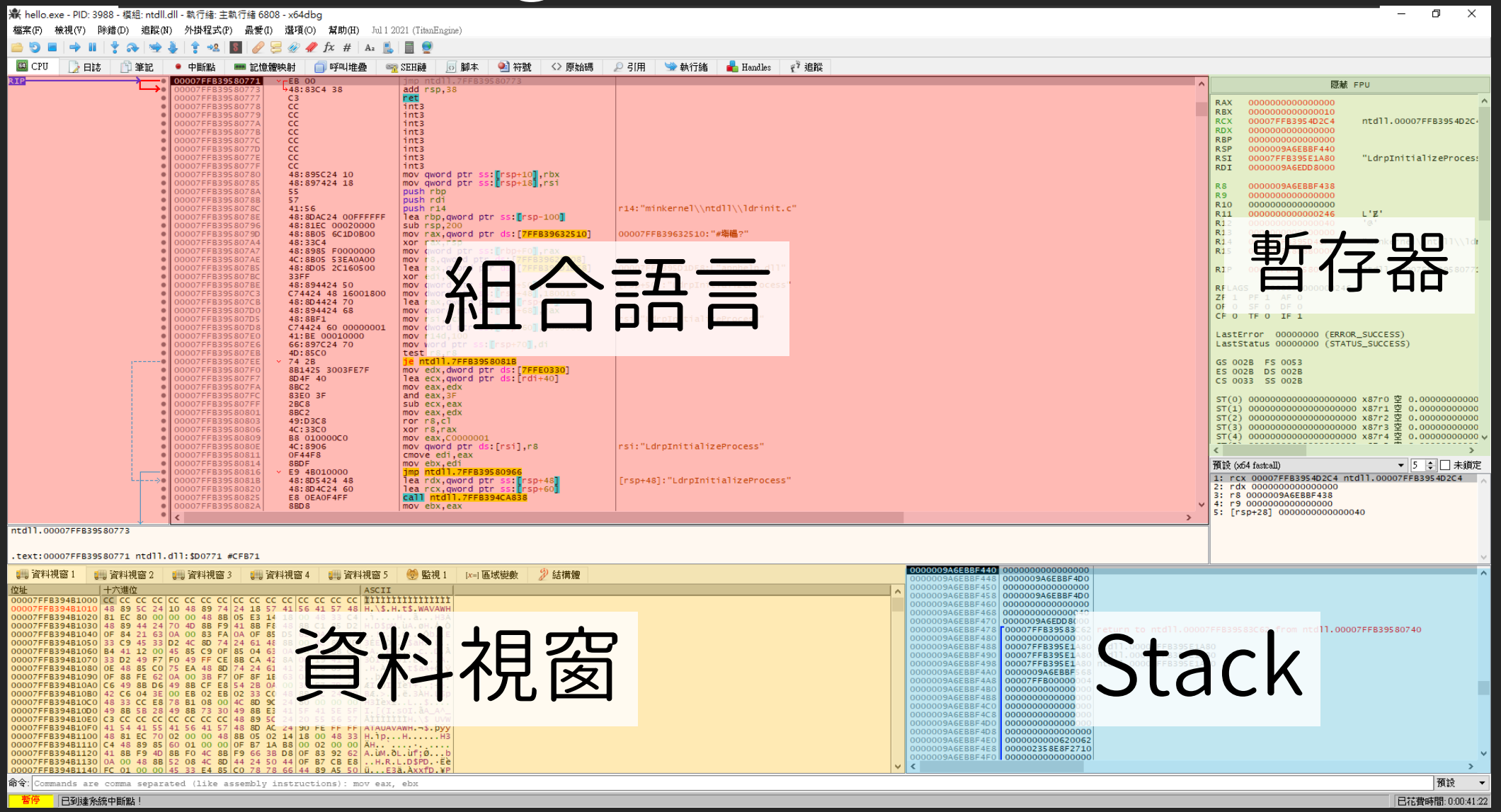


# Tools - PE-Bear

- 換算 RVA <-> Raw address



# Tools - x64dbg



# Tools - x64dbg


Ref: https://www.facebook.com/10sec.tw/photos/a.102934474545443/410257010479853/

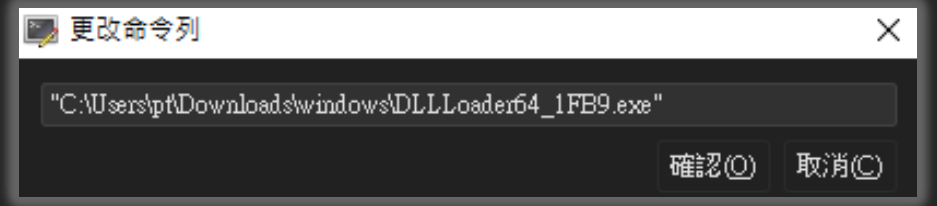
# Tools - x64dbg

# Tools - x64dbg

- 常用快捷鍵
  - F2: 設定中斷點
  - F9: 繼續執行
  - F8: 步過
  - F7: 步入
  - Ctrl+F9: 執行到 ret
  - Ctrl+G: goto
  - Space: 組譯

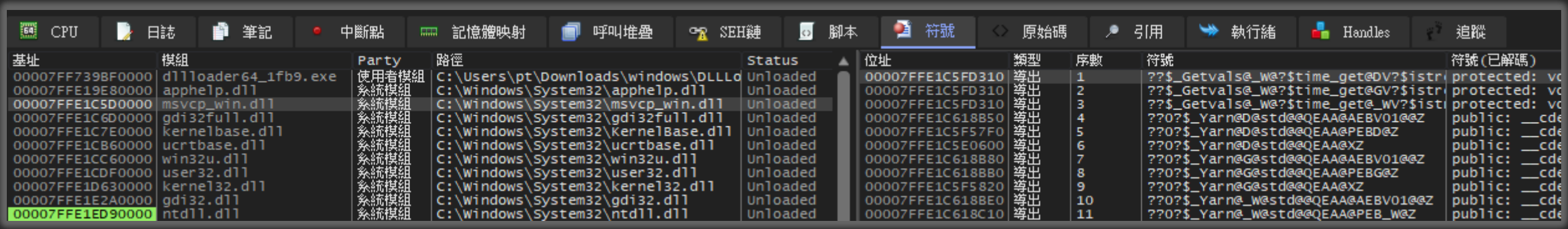
# Tools - x64dbg

- 常用功能
  - 設定指令列

# Tools - x64dbg

- 常用功能
  - 查看載入了哪些 module

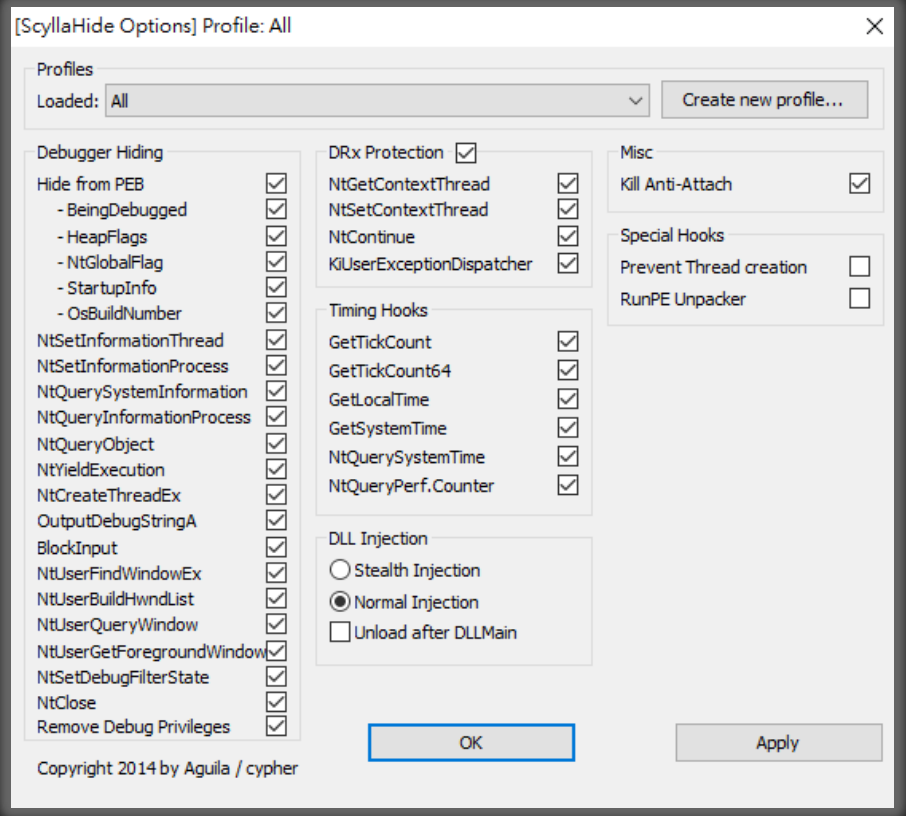


# Tools - x64dbg

- 常用功能
  - 呼叫堆疊 (call stack)
  - 記憶體映射 (memory mapping)
  - …

# Tools - x64dbg

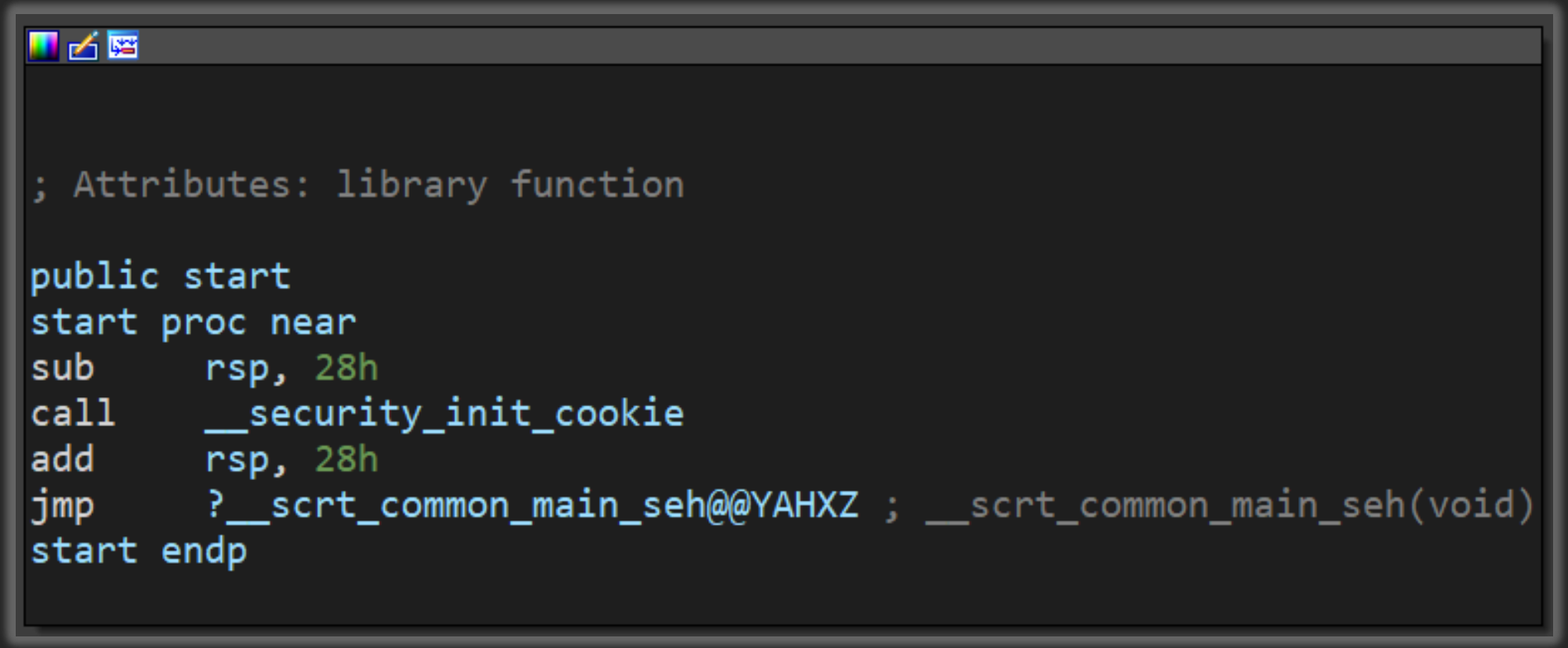
- 外掛
  - ScyllaHide: https://github.com/x64dbg/ScyllaHide

# Where to start?

# main

- Where is main?

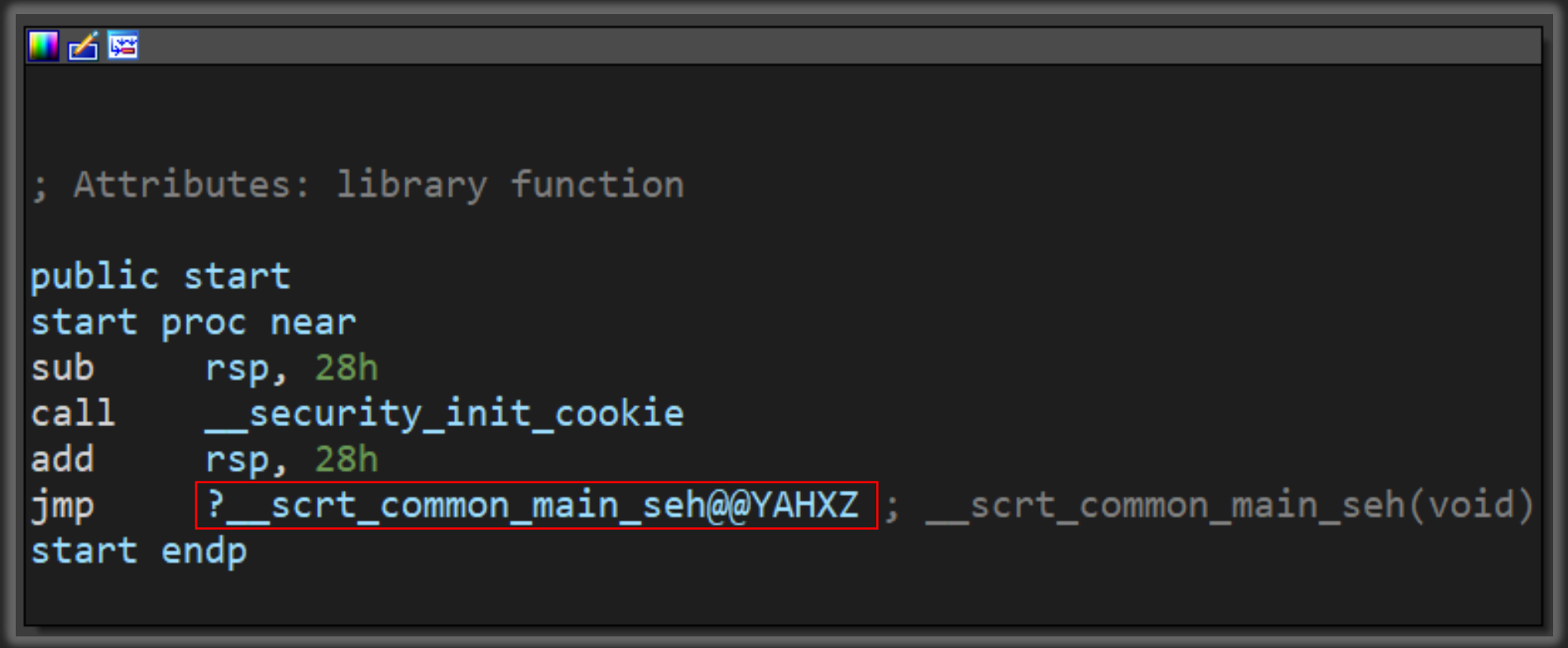


```
; Attributes: library function

public start
start proc near
sub     rsp, 28h
call    __security_init_cookie
add     rsp, 28h
jmp     ?__scrt_common_main_seh@@YAHXZ ; __scrt_common_main_seh(void)
start endp
```

# main

- Where is main?
- 在這裡面找有三個參數的 function call



# main

```
loc_1400014C2:                  ; Microsoft VisualC 64bit universal runtime
call    unknown_libname_11
mov     rdi, rax
call    sub_14000A9E0
mov     rbx, [rax]
call    sub_14000A9D8
mov     r8, rdi                 ; envp
mov     rdx, rbx                ; argv
mov     ecx, [rax]              ; argc
call    main
```

# Before main

- 比 main 還要早跑的 init 的部分: _initterm_e

```
mov     cs:dword_140024BB0, 1
lea     rdx, unk_1400192C0
lea     rcx, unk_140019288
call    _initterm_e
test    eax, eax
jz      short loc_140001716
```

```
.rdata:0000000140019288 qword_140019288 dq 0                    ; DATA XREF: __scr
.rdata:0000000140019290                 dq offset sub_1400015C4
.rdata:0000000140019298                 dq offset ?post_pgo_initialization@@YAHXZ
.rdata:00000001400192A0                 dq offset __acrt_initialize_stdio
.rdata:00000001400192A8                 dq offset ?initialize_multibyte@@YAHXZ ; i
.rdata:00000001400192B0                 dq offset sub_140010F68
.rdata:00000001400192B8                 dq offset __acrt_initialize_fma3
```

# Before main

- 比 main 還要早跑的 init 的部分: _initterm

```
loc_140001716:
lea     rdx, unk_140019280
lea     rcx, unk_140019268
call    _initterm
mov     cs:dword_140024BB0, 2
jmp     short loc_14000173D
```

```
.rdata:0000000140019268 qword_140019268 dq 0                    ; DATA XREF: __scr
.rdata:0000000140019270                 dq offset ?pre_cpp_initialization@@YAXXZ ;
.rdata:0000000140019278                 dq offset sub_1400011D0
.rdata:0000000140019280 unk_140019280   db    0                 ; DATA XREF: __scr
```

# After main

- 在 main 結束後才跑的部分, 需用以下 API 註冊要跑的函數
  - atexit
  - _onexit, _onexit_m
  - __dllonexit

# Address Space

# Address Space

- 目前申請到的記憶體空間
- 從 x64dbg 記憶體映射觀察
- 起始位址 / 大小 / 權限 (E: 可執行; R: 可讀; W: 可寫)

| | | | | | | |
|---|---|---|---|---|---|---|
| 00007FF66C390000 | 0000000000001000 | hellowindows.exe | | IMG | -R--- | ERWC- |
| 00007FF66C391000 | 0000000000018000 | ".text" | 可執行代碼 | IMG | ER--- | ERWC- |
| 00007FF66C3A9000 | 000000000000B000 | ".rdata" | 唯讀的已初始化的資料 | IMG | -R--- | ERWC- |
| 00007FF66C3B4000 | 0000000000002000 | ".data" | 已初始化的資料 | IMG | -RW-- | ERWC- |
| 00007FF66C3B6000 | 0000000000002000 | ".pdata" | 例外資料 | IMG | -R--- | ERWC- |
| 00007FF66C3B8000 | 0000000000001000 | "_RDATA" | | IMG | -R--- | ERWC- |
| 00007FF66C3B9000 | 0000000000001000 | ".rsrc" | 資源 | IMG | -R--- | ERWC- |
| 00007FF66C3BA000 | 0000000000001000 | ".reloc" | Base relocations | IMG | -R--- | ERWC- |

# Lab 1

# Speed up!

# Speed up

- 更快的找到重要程式碼
- 直接通靈程式會做什麼
- 看引用了哪些外部函數來當作通靈依據

# Speed up

- socket, connect, recv, send
  - 網路連線
- CreateService, RegCreateKeyEx
  - 猜測在嘗試持久化程式執行
- CryptEncrypt, CryptDecrypt
  - 加解密

Ref: https://book.hacktricks.xyz/reversing/common-api-used-in-malware

# Speed up

- VirtualAlloc, VirtualProtect, CreateRemoteThread, CreateProcess, ResumeThread
  - 惡意程式愛用的 API

Ref: https://book.hacktricks.xyz/reversing/common-api-used-in-malware

# Speed up

- RegisterClass 家族
  - MFC application 用來註冊 window class 的 API

```
ATOM RegisterClassW(
  [in] const WNDCLASSW *lpWndClass
);
```

```
typedef struct tagWNDCLASSA {
  UINT      style;
  WNDPROC   lpfnWndProc;
  int       cbClsExtra;
  int       cbWndExtra;
  HINSTANCE hInstance;
  HICON     hIcon;
  HCURSOR   hCursor;
  HBRUSH    hbrBackground;
  LPCSTR    lpszMenuName;
  LPCSTR    lpszClassName;
} WNDCLASSA, *PWNDCLASSA, *NPWNDCLASSA, *LPWNDCLASSA;
```

Ref: https://docs.microsoft.com/en-us/windows/win32/api/winuser/nf-winuser-registerclassw

# Speed up

- RegisterClass 家族
  - MFC application 用來註冊 window class 的 API

Callback function

```
ATOM RegisterClassW(
  [in] const WNDCLASSW *lpWndClass
);
```

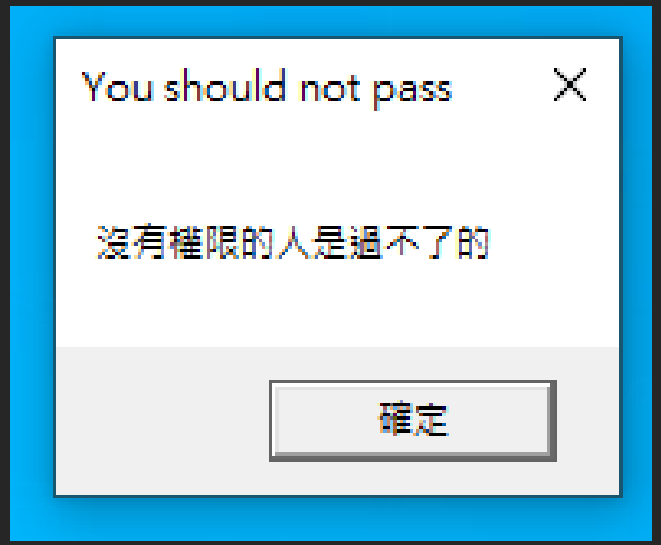
```
typedef struct tagWNDCLASSA {
  UINT      style;
  WNDPROC   lpfnWndProc;
  int       cbClsExtra;
  int       cbWndExtra;
  HINSTANCE hInstance;
  HICON     hIcon;
  HCURSOR   hCursor;
  HBRUSH    hbrBackground;
  LPCSTR    lpszMenuName;
  LPCSTR    lpszClassName;
} WNDCLASSA, *PWNDCLASSA, *NPWNDCLASSA, *LPWNDCLASSA;
```

Ref: https://docs.microsoft.com/en-us/windows/win32/api/winuser/nf-winuser-registerclassw

# Speed up

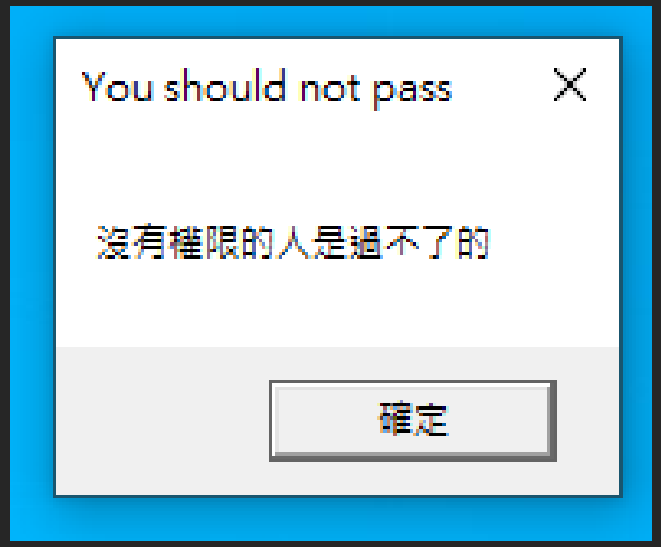
- 在覺得會執行的 API 設定中斷點
- 真的停在中斷點後, 從 call stack 往回找是哪邊呼叫到這個 API

# Speed up



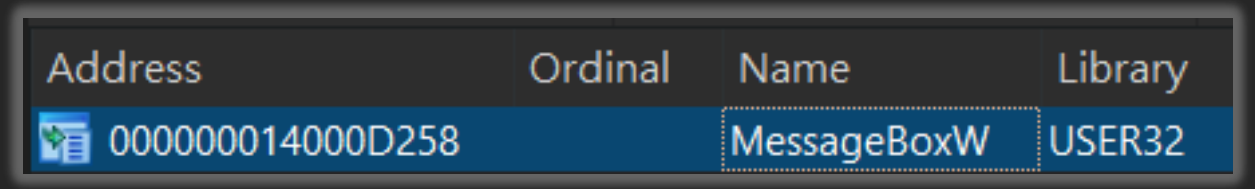
# Speed up

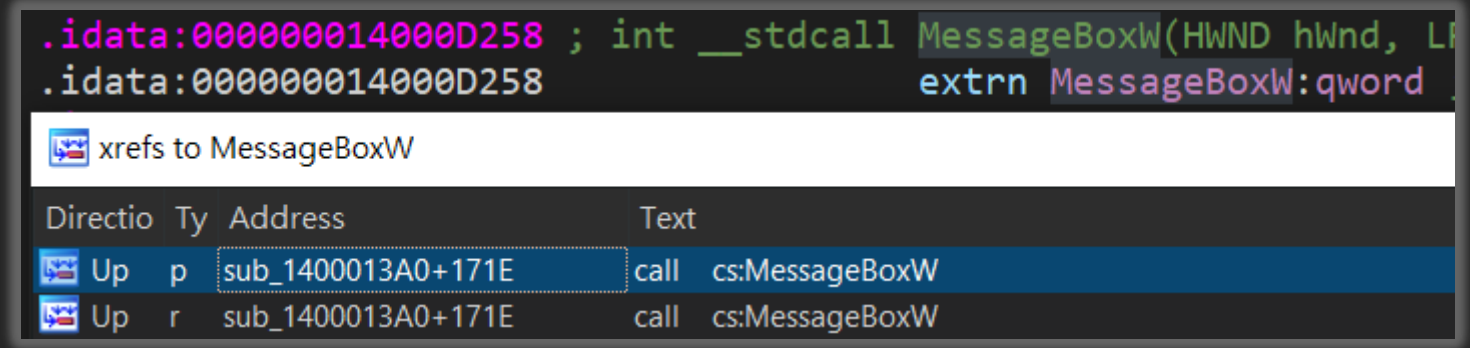
- 猜是呼叫 MessageBox 家族 API 所創出的 window



# Speed up

- 從 IDA Imports 看, 發現有引用

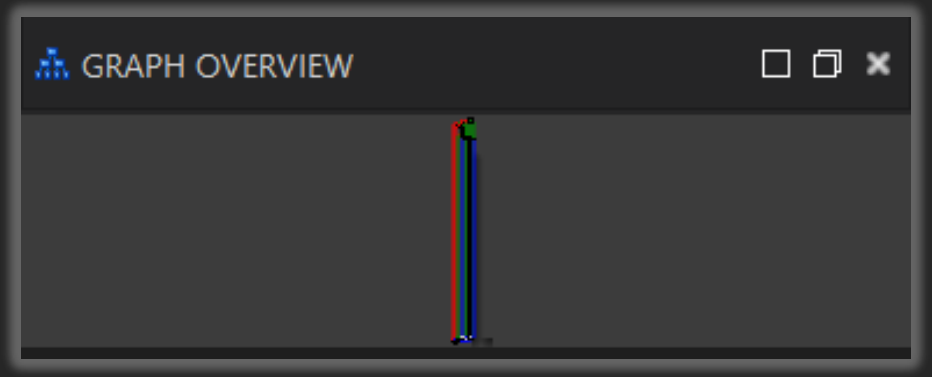


# Speed up

- 往回追到關鍵程式碼區域

```
loc_140002AB8:                  ; uType
xor     r9d, r9d
mov     rcx, rsi                ; hWnd
call    cs:MessageBoxW
```



# Lab 2

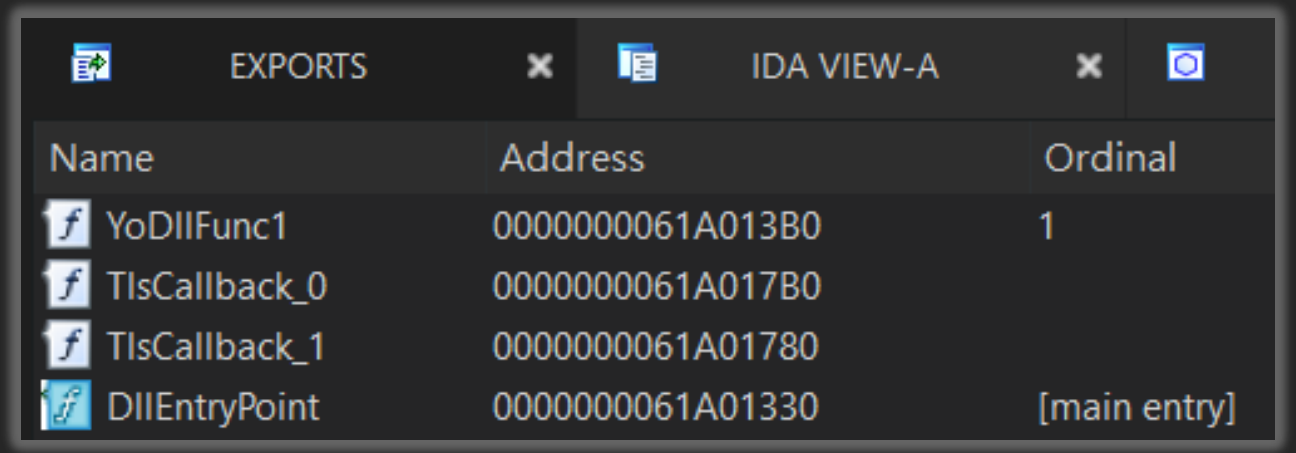
# DLL

# DLL

- Dynamic-Link Library
- 提供函數給程式使用
- Export function

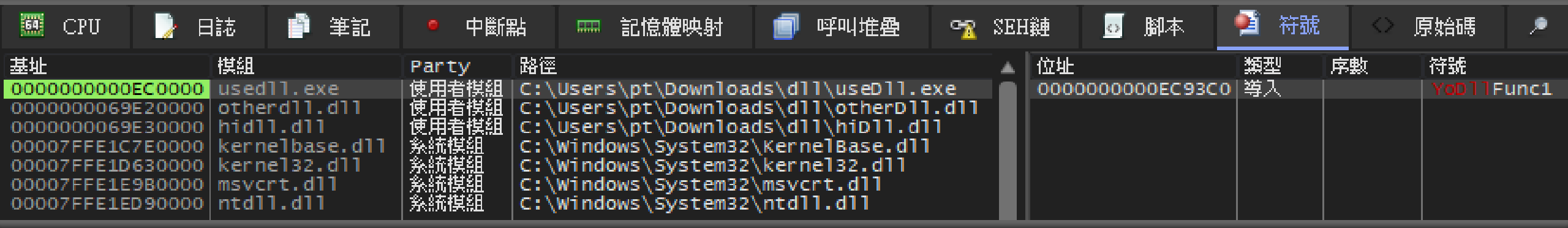
# DLL

- 查看提供了哪些函數



# DLL

- x64dbg 符號這一區還不錯用
- 看目前引用了哪些 module
- 看 module import/export 哪些函數



# DLL

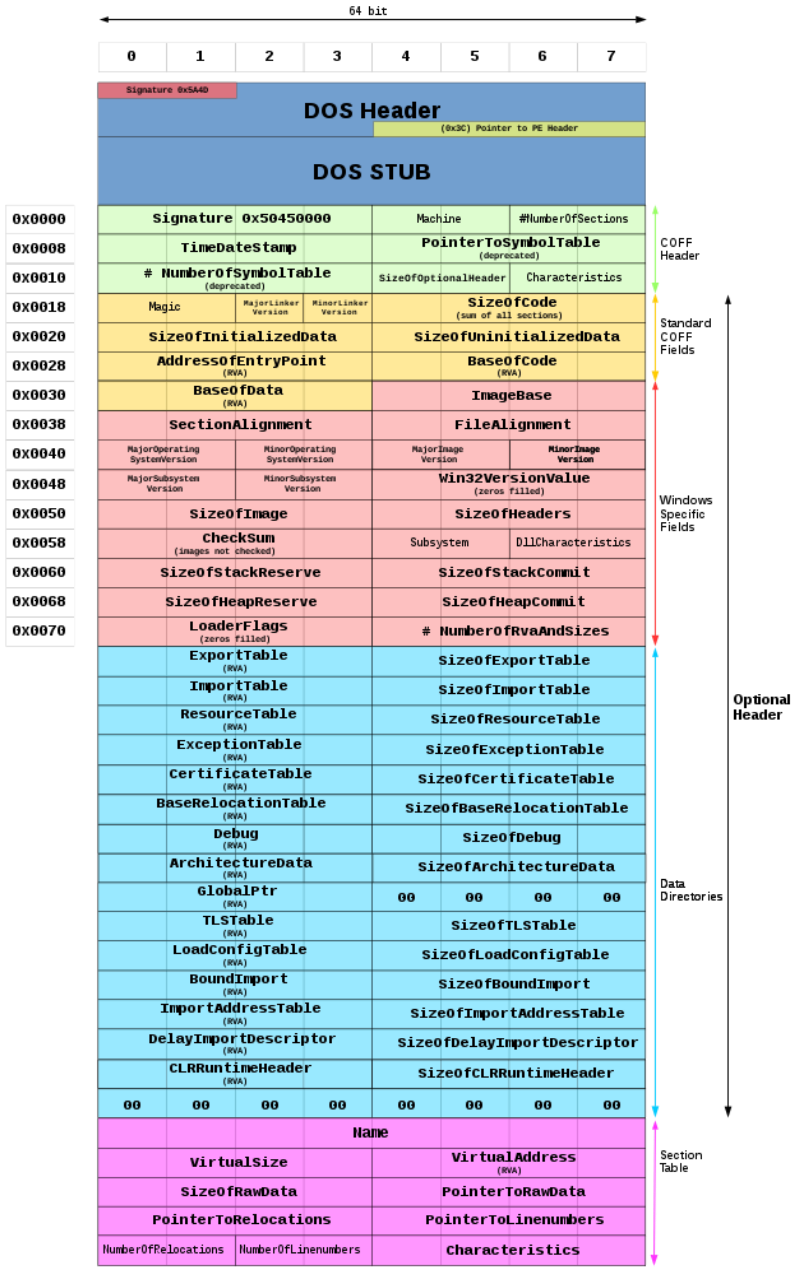
- DllMain(HINSTANCE hinstDLL, DWORD fdwReason, LPVOID lpReserved )
- 會在各種時機呼叫到 (e.g. 剛載入時, 要卸載時)
- 可以設定, 也可以不設定

# Lab 3

# PE format

# PE format

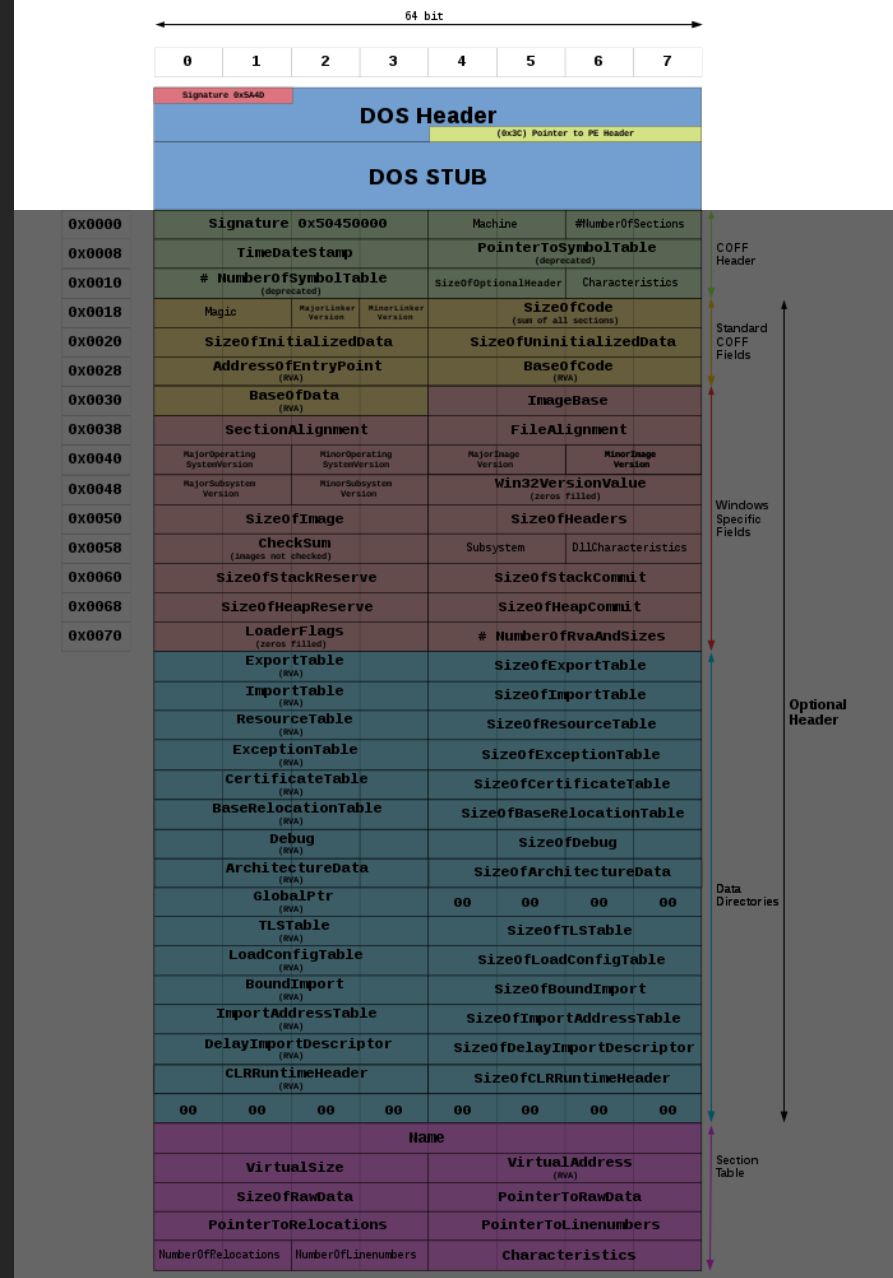
- 這就是你熟悉的 exe 的結構
- 是不是看了就頭痛
- 讓我們一點一點地拆開來說



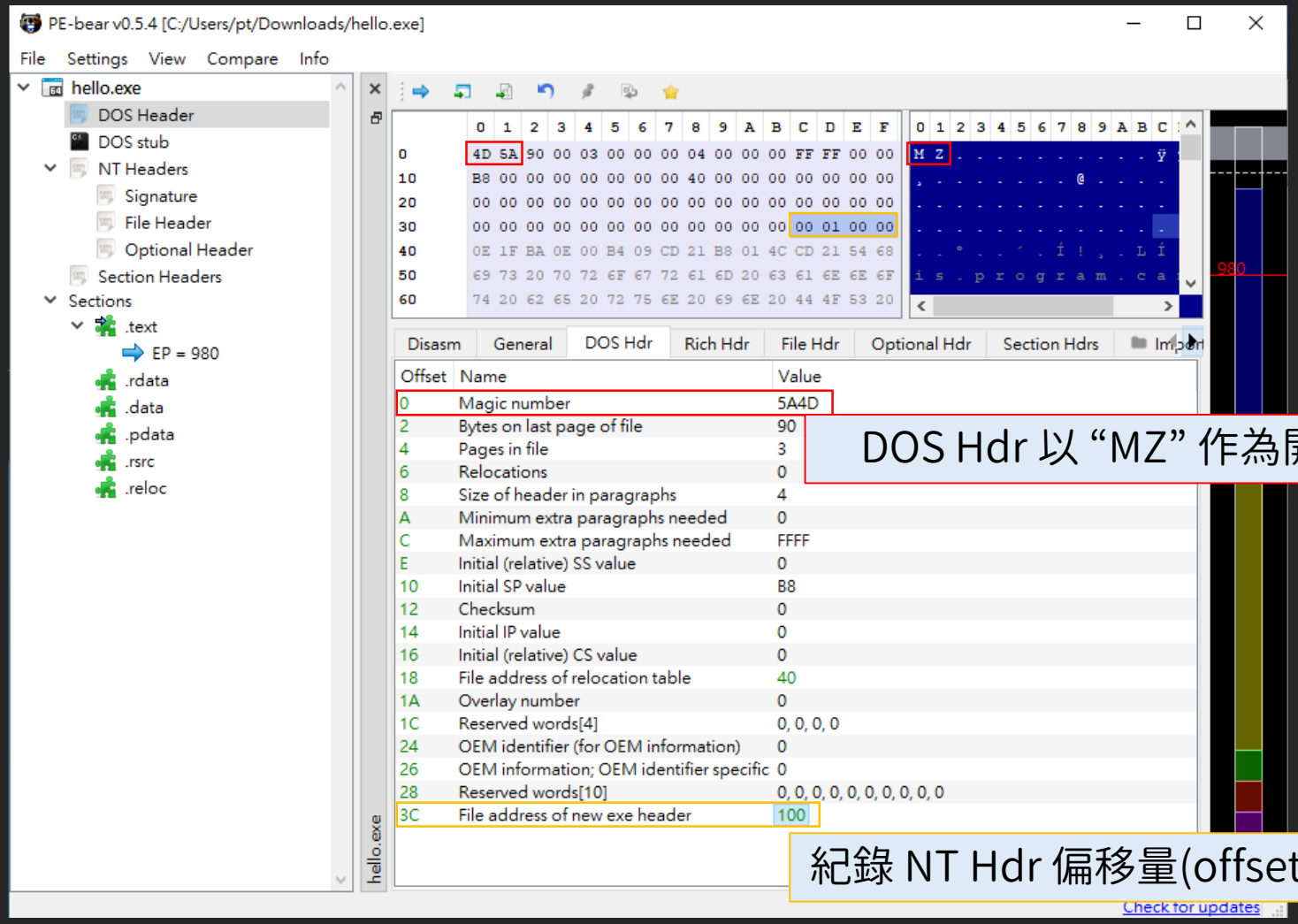

Ref: https://en.wikipedia.org/wiki/Portable_Executable

# PE format

- 首先看最上面的 DOS Header
- 用 PE-Bear 來展示一下



# PE format

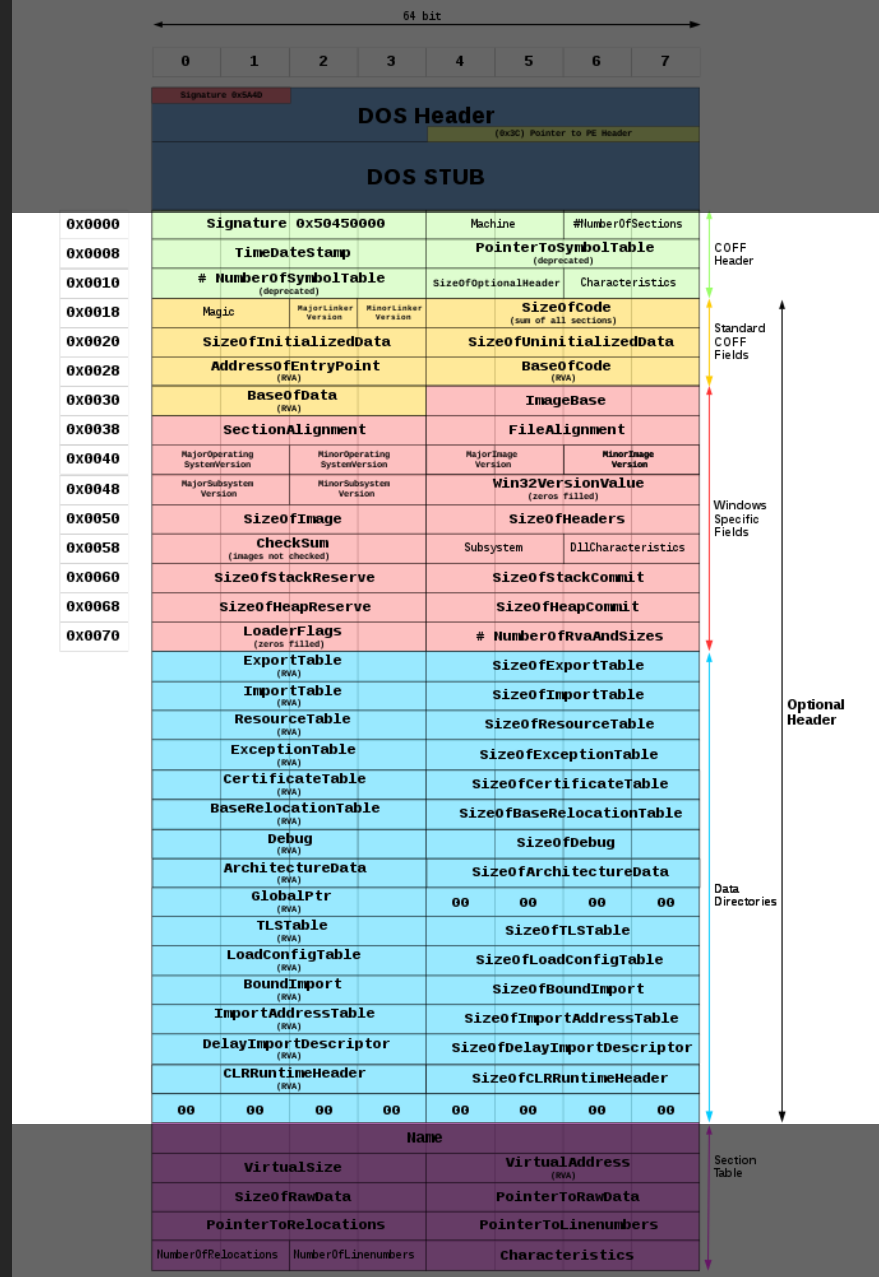



DOS Hdr 以 “MZ” 作為開頭

紀錄 NT Hdr 偏移量(offset)多少

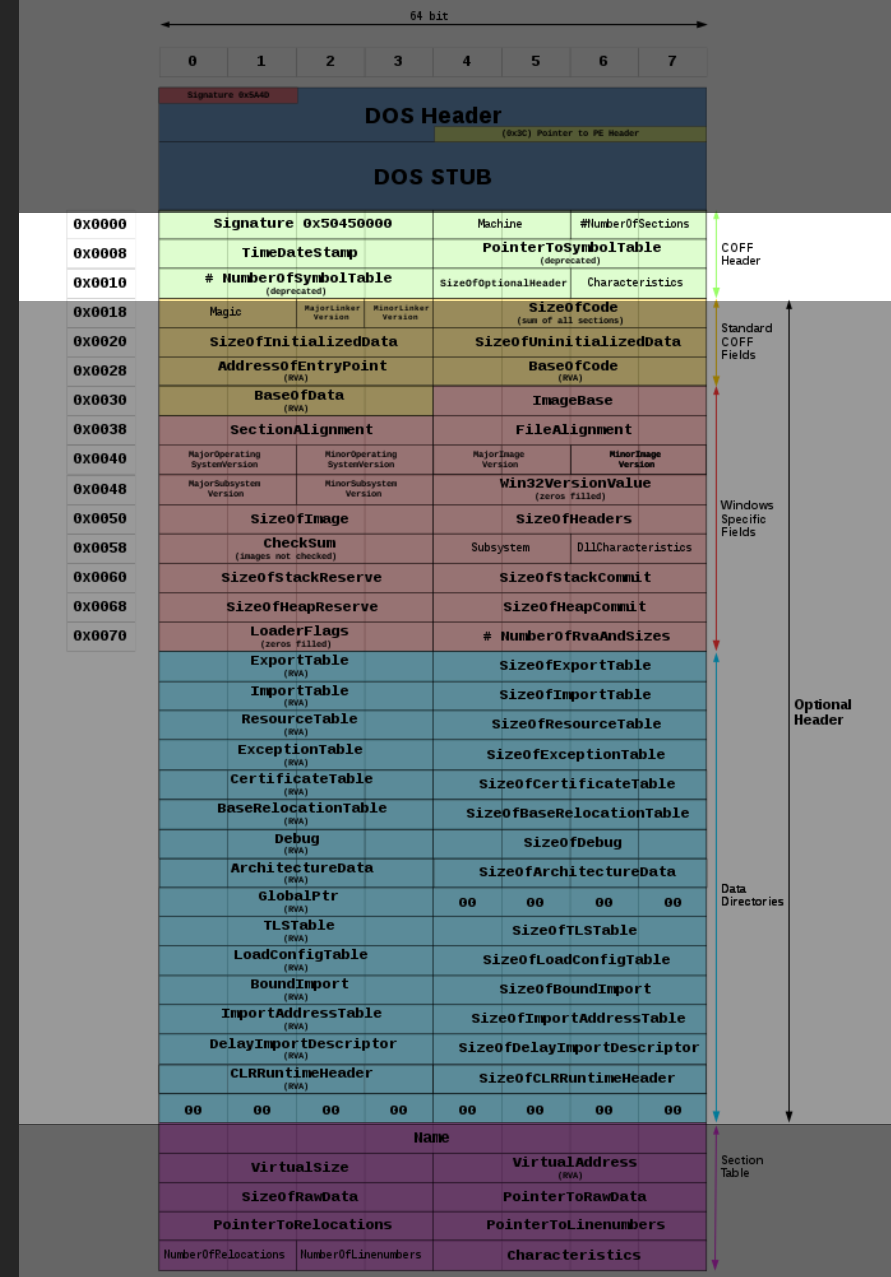
# PE format

- 再來是 NT Hdr (或稱 PE Hdr)
- 其包含了



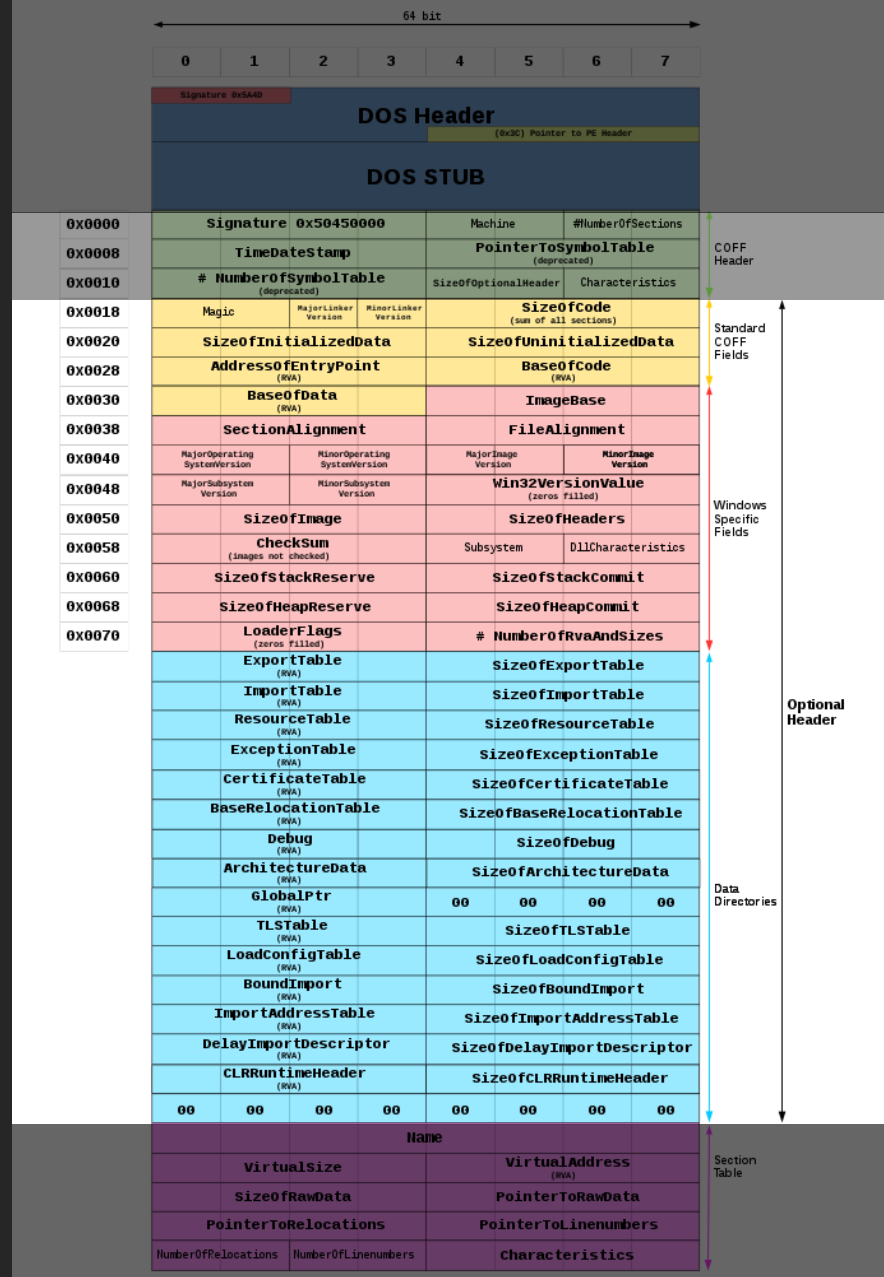
# PE format

- 再來是 NT Hdr (或稱 PE Hdr)
- 其包含了
  - COFF Hdr (或稱 File Hdr)



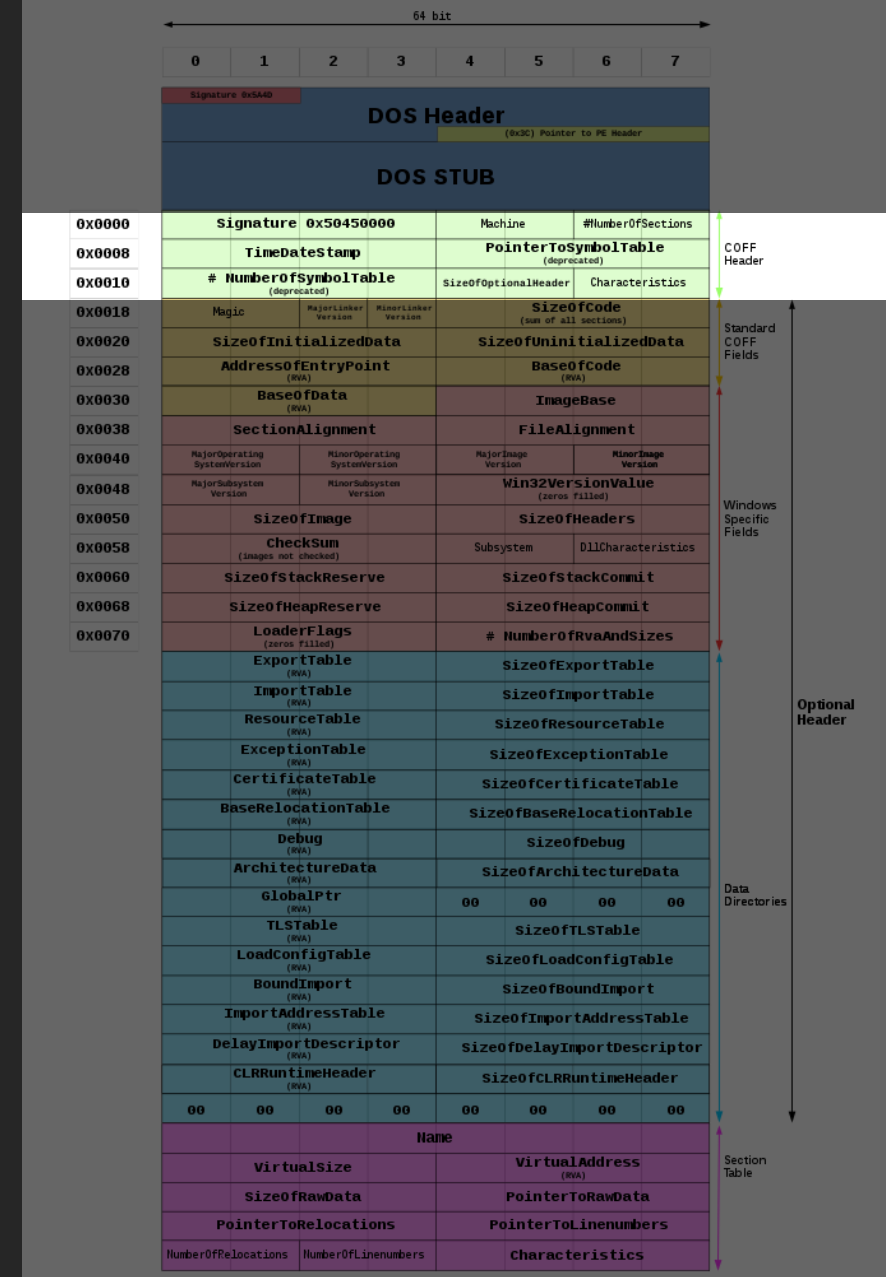
# PE format

- 再來是 NT Hdr (或稱 PE Hdr)
- 其包含了
  - COFF Hdr (或稱 File Hdr)
  - Optional Hdr

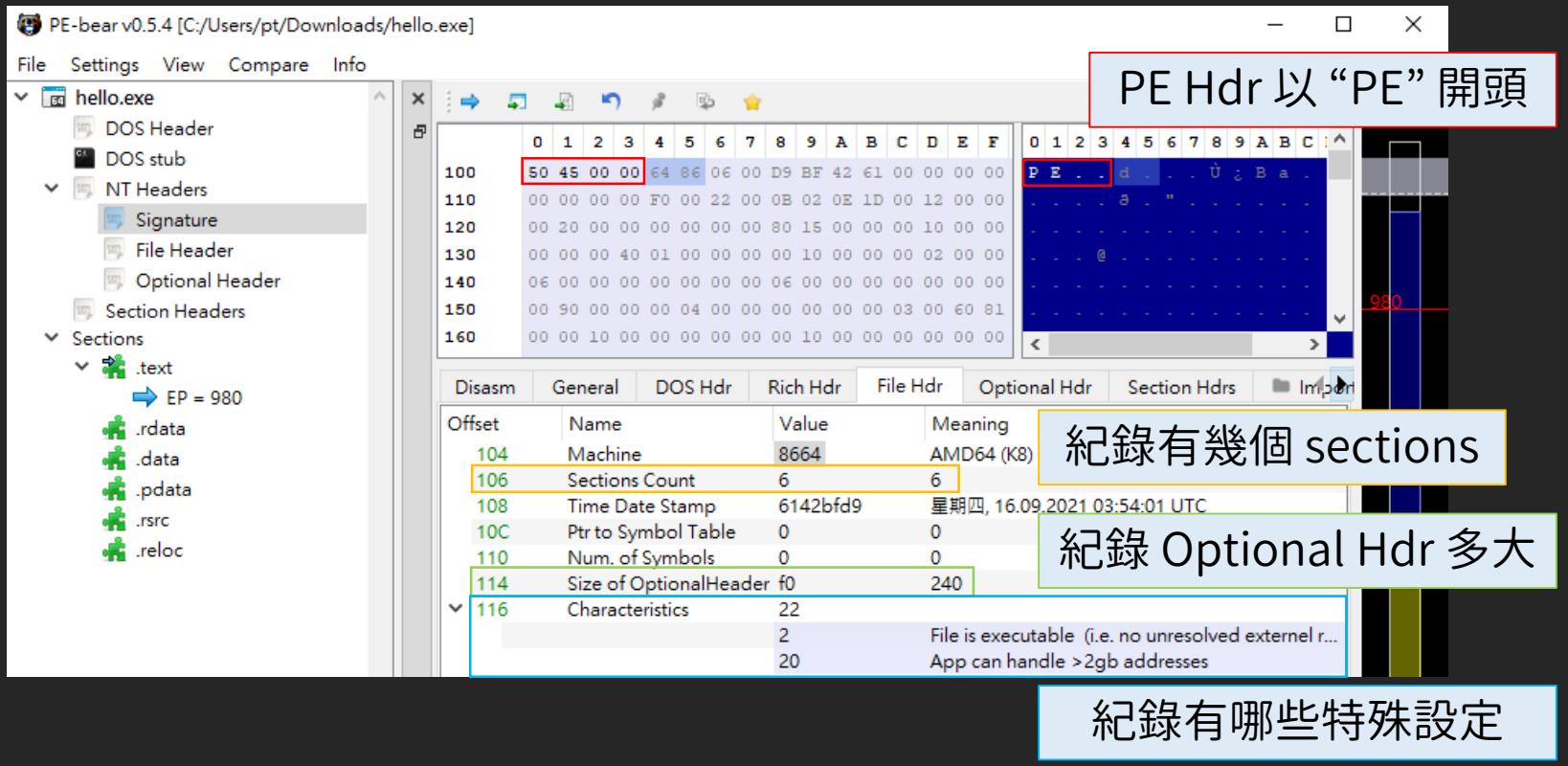


# PE format

- COFF Hdr (或稱 File Hdr)
- 用 PE-Bear 來展示一下



# PE format

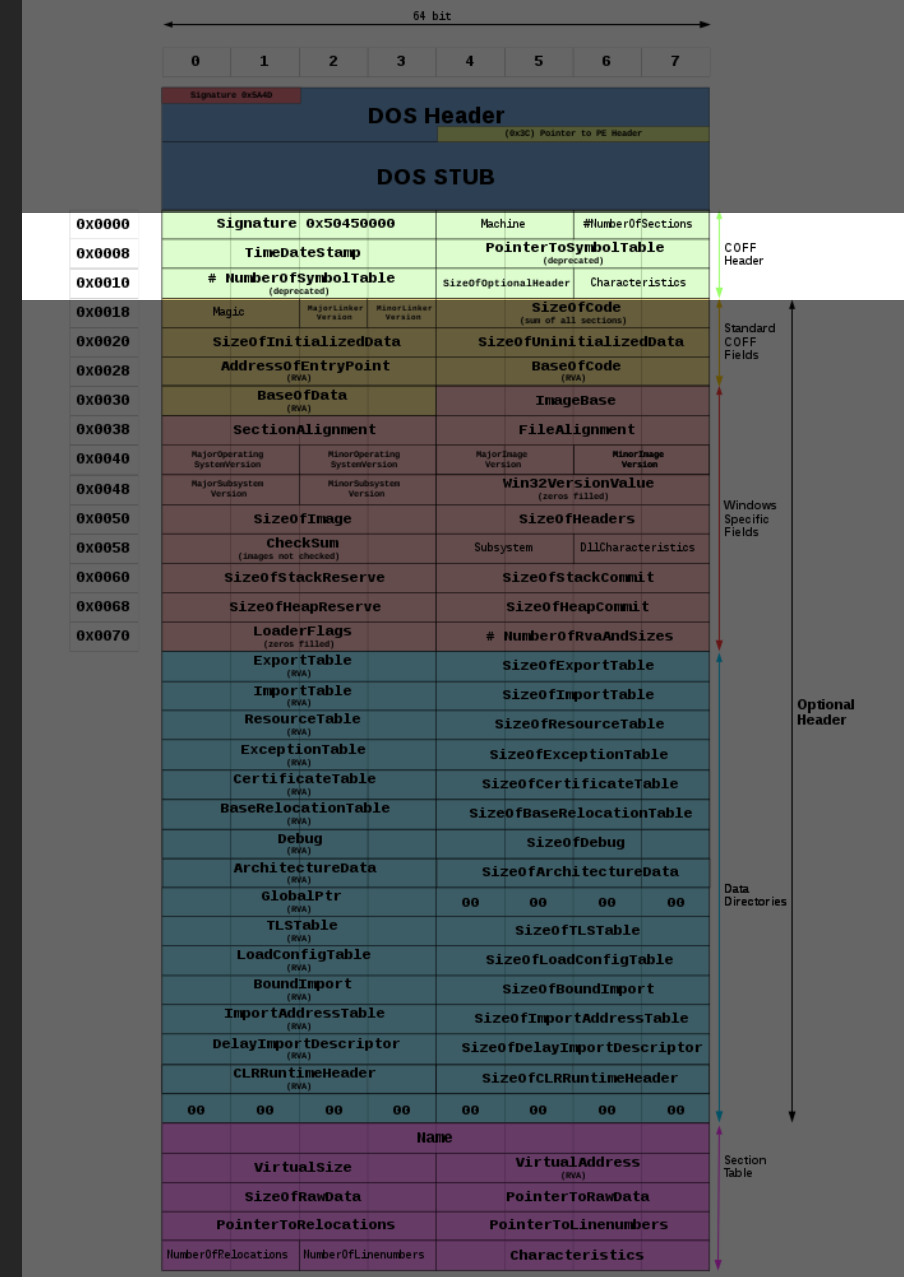



PE Hdr 以 “PE” 開頭

紀錄有幾個 sections

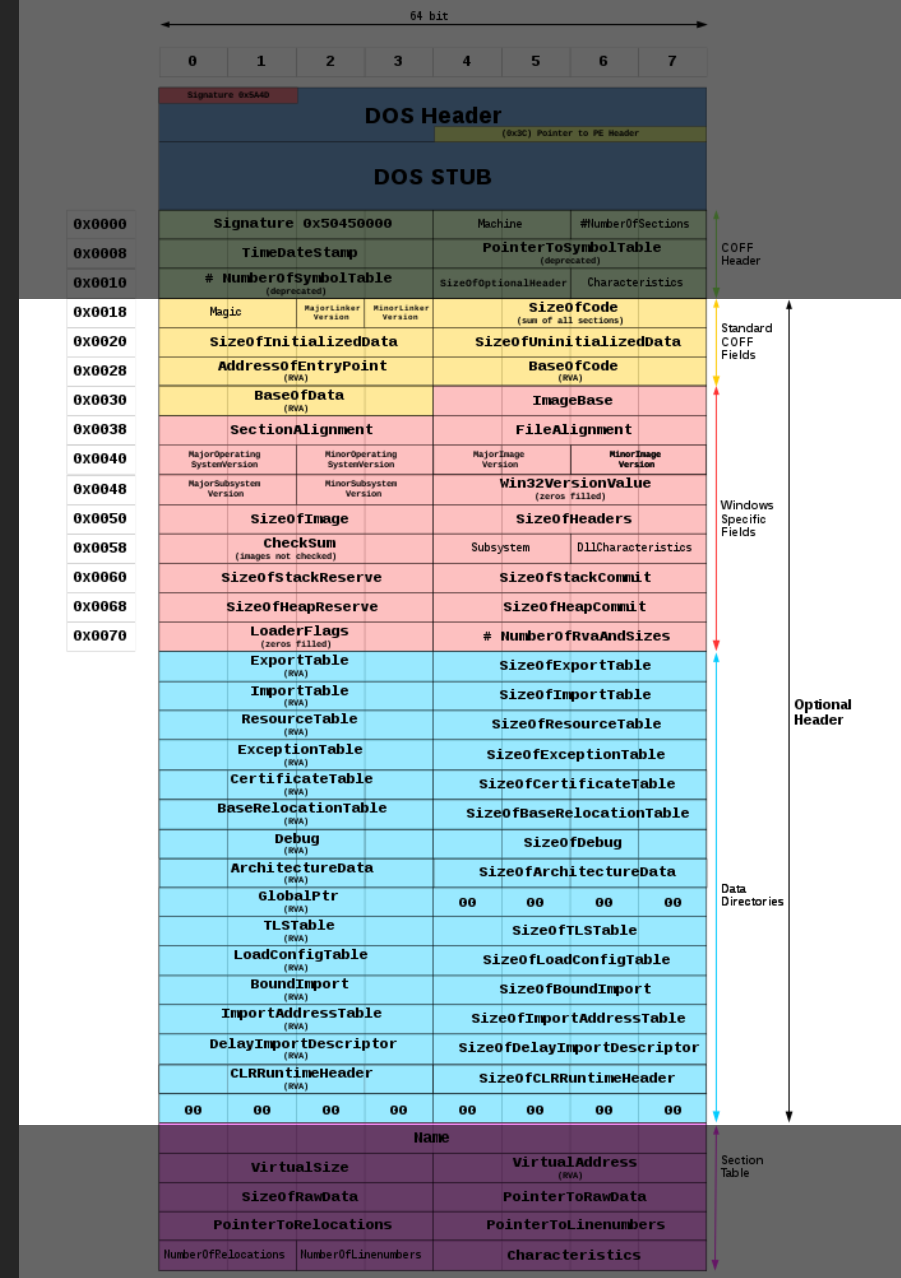
紀錄 Optional Hdr 多大

紀錄有哪些特殊設定



# PE format

- Optional Hdr
- 用 PE-Bear 來展示一下



# PE format

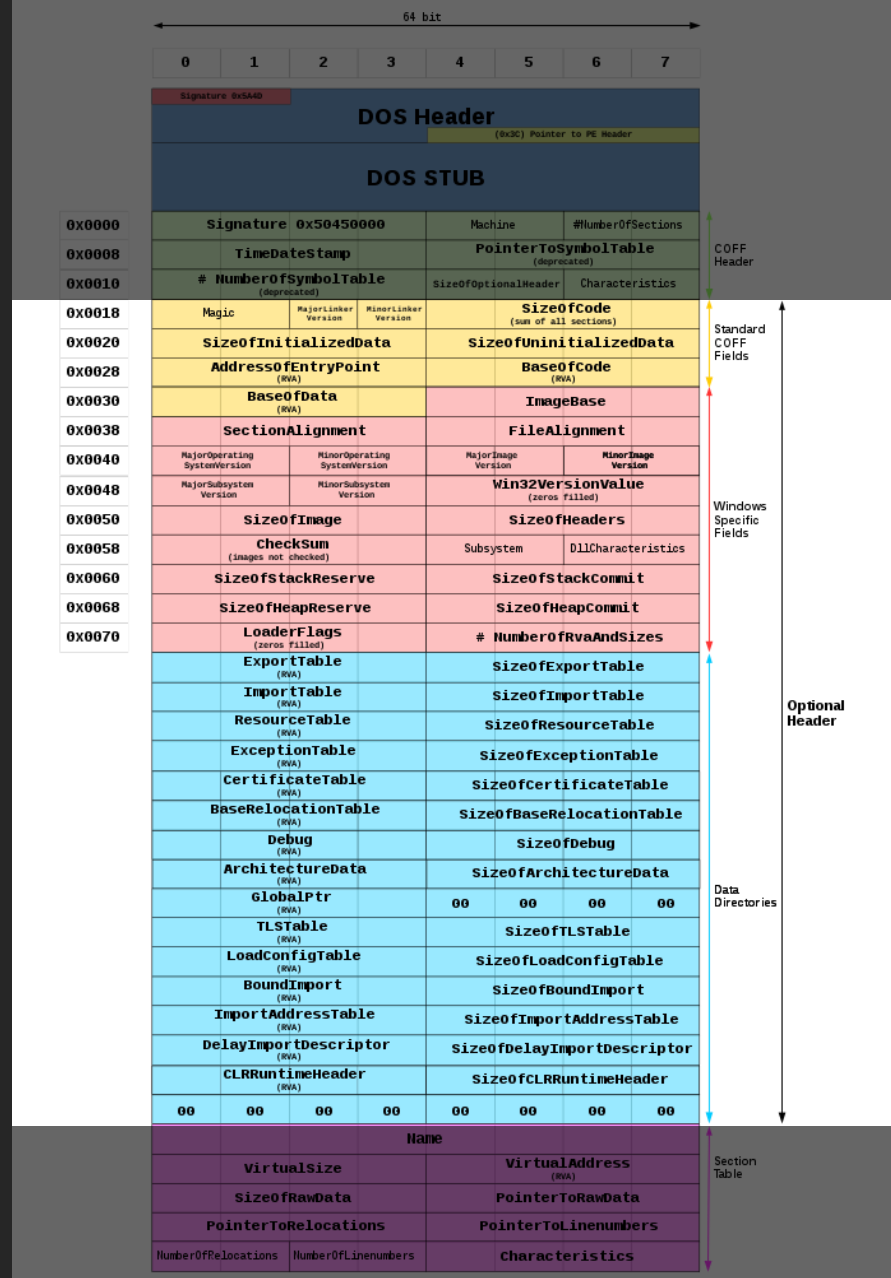
Disasm | General | DOS Hdr | Rich Hdr | File Hdr | Optional Hdr | Section Hdrs | Im

| Offset | Name | Value | Value |
|---|---|---|---|
| 118 | Magic | 20B | NT64 |
| 11A | Linker Ver. (Major) | E | |
| 11B | Linker Ver. (Minor) | 1D | |
| 11C | Size of Code | 1200 | |
| 120 | Size of Initialized Data | 2000 | |
| 124 | Size of Uninitialized Data | 0 | |
| 128 | Entry Point | 1580 | |
| 12C | Base of Code | 1000 | |
| 130 | Image Base | 140000000 | |
| 138 | Section Alignment | 1000 | |
| 13C | File Alignment | 200 | |
| 140 | OS Ver. (Major) | 6 | Windows Vista / Server 2008 |
| 142 | OS Ver. (Minor) | 0 | |
| 144 | Image Ver. (Major) | 0 | |
| 146 | Image Ver. (Minor) | 0 | |
| 148 | Subsystem Ver. (Major) | 6 | |
| 14A | Subsystem Ver. Minor) | 0 | |
| 14C | Win32 Version Value | 0 | |
| 150 | Size of Image | 9000 | |
| 154 | Size of Headers | 400 | |
| 158 | Checksum | 0 | |
| 15C | Subsystem | 3 | Windows console |
| 15E | DLL Characteristics | 8160 | |
| | | 40 | DLL can move |
| | | 100 | Image is NX compatible |
| | | 8000 | TerminalServer aware |

程式進入點 RVA

程式基址

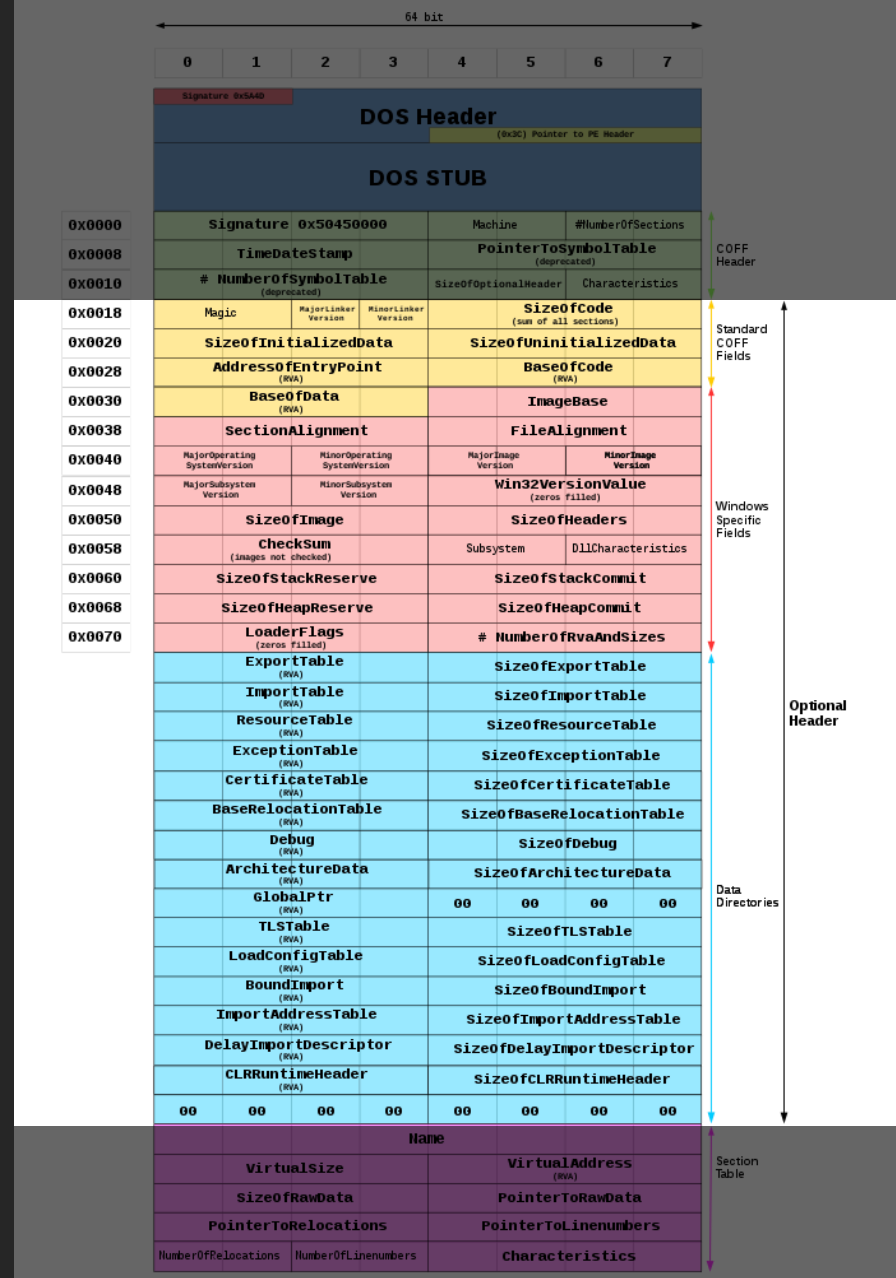
紀錄有哪些特殊設定



# PE format

| | | | |
|---|---|---|---|
| 160 | Size of Stack Reserve | 100000 | |
| 168 | Size of Stack Commit | 1000 | |
| 170 | Size of Heap Reserve | 100000 | |
| 178 | Size of Heap Commit | 1000 | |
| 180 | Loader Flags | 0 | |
| 184 | Number of RVAs and Sizes | 10 | |
| | Data Directory | Address | Size |
| 188 | Export Directory | 0 | 0 |
| 190 | Import Directory | 396C | C8 |
| 198 | Resource Directory | 7000 | 1E0 |
| 1A0 | Exception Directory | 6000 | 198 |
| 1A8 | Security Directory | 0 | 0 |
| 1B0 | Base Relocation Table | 8000 | 2C |
| 1B8 | Debug Directory | 3310 | 70 |
| 1C0 | Architecture Specific Data | 0 | 0 |
| 1C8 | RVA of GlobalPtr | 0 | 0 |
| 1D0 | TLS Directory | 0 | 0 |
| 1D8 | Load Configuration Directory | 3380 | 138 |
| 1E0 | Bound Import Directory in headers | 0 | 0 |
| 1E8 | Import Address Table | 3000 | 1F0 |
| 1F0 | Delay Load Import Descriptors | 0 | 0 |
| 1F8 | .NET header | 0 | 0 |

各種 Directory 位址及大小



# PE format

- 解釋一下 RVA (Relative Virtual Address)
- 首先先直接展示一個程式在跑的時候，記憶體位址的樣子

# PE format

| 位址 | 大小 | 資訊 | 內容 | 類型 | 保護 | 初始保護 |
|---|---|---|---|---|---|---|
| 0000002DAA5A4000 | 000000000005C000 | Reserved (0000002DA | | PRV | | -RW-- |
| 0000002DAA600000 | 00000000000FC000 | Reserved | | PRV | | -RW-- |
| 0000002DAA6FC000 | 0000000000004000 | | | PRV | -RW-G | -RW-- |
| 000001FAD43A0000 | 0000000000010000 | | | MAP | -RW-- | -RW-- |
| 000001FAD43B0000 | 0000000000001000 | \Device\HarddiskVol | | MAP | -R--- | -R--- |
| 000001FAD43C0000 | 000000000001D000 | | | MAP | -R--- | -R--- |
| 000001FAD43E0000 | 0000000000004000 | | | MAP | -R--- | -R--- |
| 000001FAD43F0000 | 0000000000001000 | | | MAP | -R--- | -R--- |
| 000001FAD4400000 | 0000000000002000 | | | PRV | -RW-- | -RW-- |
| 000001FAD4410000 | 00000000000C9000 | \Device\HarddiskVol | | MAP | -R--- | -R--- |
| 000001FAD44E0000 | 0000000000001000 | | | MAP | -R--- | -R--- |
| 000001FAD44F0000 | 0000000000010000 | | | PRV | -RW-- | -RW-- |
| 000001FAD4500000 | 00000000000F0000 | Reserved (000001FAD | | PRV | | -RW-- |
| 000001FAD45F0000 | 0000000000001000 | | | MAP | -R--- | -R--- |
| 000001FAD4600000 | 0000000000001000 | | | MAP | -R--- | -R--- |
| 00007FF4F0140000 | 0000000000005000 | | | MAP | -R--- | -R--- |
| 00007FF4F0145000 | 00000000000FB000 | Reserved (00007FF4F | | MAP | | -R--- |
| 00007FF4F0240000 | 0000000100020000 | Reserved | | PRV | | -RW-- |
| 00007FF5F0260000 | 0000000002000000 | Reserved | | PRV | | -RW-- |
| 00007FF5F2260000 | 0000000000001000 | | | PRV | -RW-- | -RW-- |
| 00007FF5F2270000 | 0000000000001000 | | | MAP | -R--- | -R--- |
| 00007FF5F2280000 | 0000000000033000 | | | MAP | -R--- | -R--- |
| 00007FF676150000 | 0000000000001000 | hello.exe | | IMG | -R--- | ERWC- |
| 00007FF676151000 | 0000000000002000 | ".text" | 可執行代碼 | IMG | ER--- | ERWC- |
| 00007FF676153000 | 0000000000002000 | ".rdata" | 唯讀的已初始化的資料 | IMG | -R--- | ERWC- |
| 00007FF676155000 | 0000000000001000 | ".data" | 已初始化的資料 | IMG | -RW-- | ERWC- |
| 00007FF676156000 | 0000000000001000 | ".pdata" | 例外資料 | IMG | -R--- | ERWC- |
| 00007FF676157000 | 0000000000001000 | ".rsrc" | 資源 | IMG | -R--- | ERWC- |
| 00007FF676158000 | 0000000000001000 | ".reloc" | Base relocations | IMG | -R--- | ERWC- |
| 00007FFB29680000 | 0000000000001000 | vcruntime140.dll | | IMG | -R--- | ERWC- |
| 00007FFB29681000 | 0000000000010000 | ".text" | 可執行代碼 | IMG | ER--- | ERWC- |
| 00007FFB29691000 | 0000000000004000 | ".rdata" | 唯讀的已初始化的資料 | IMG | -R--- | ERWC- |
| 00007FFB29695000 | 0000000000001000 | ".data" | 已初始化的資料 | IMG | -RW-- | ERWC- |
| 00007FFB29696000 | 0000000000001000 | ".pdata" | 例外資料 | IMG | -R--- | ERWC- |
| 00007FFB29697000 | 0000000000001000 | "_RDATA" | | IMG | -R--- | ERWC- |
| 00007FFB29698000 | 0000000000001000 | ".rsrc" | 資源 | IMG | -R--- | ERWC- |
| 00007FFB29699000 | 0000000000001000 | ".reloc" | Base relocations | IMG | -R--- | ERWC- |

# PE format

| 位址 | 大小 | 資訊 | 內容 | 類型 | 保護 | 初始保護 |
|---|---|---|---|---|---|---|
| 0000002DAA5A4000 | 000000000005C000 | Reserved (0000002DA | | PRV | | -RW-- |
| 0000002DAA600000 | 000000000000FC000 | Reserved | | PRV | | -RW-- |
| 0000002DAA6FC000 | 0000000000004000 | | | PRV | -RW-G | -RW-- |
| 000001FAD43A0000 | 0000000000010000 | | | MAP | -RW-- | -RW-- |
| 000001FAD43B0000 | 0000000000001000 | \Device\HarddiskVol | | MAP | -R--- | -R--- |
| 000001FAD43C0000 | 000000000001D000 | | | MAP | -R--- | -R--- |
| 000001FAD43E0000 | 0000000000004000 | | | MAP | -R--- | -R--- |
| 000001FAD43F0000 | 0000000000001000 | | | MAP | -R--- | -R--- |
| 000001FAD4400000 | 0000000000002000 | | | PRV | -RW-- | -RW-- |
| 000001FAD4410000 | 00000000000C9000 | \Device\HarddiskVol | | MAP | -R--- | -R--- |
| 000001FAD44E0000 | 0000000000001000 | | | MAP | -R--- | -R--- |
| 000001FAD44F0000 | 0000000000010000 | | | PRV | -RW-- | -RW-- |
| 000001FAD4500000 | 00000000000F0000 | Reserved (000001FAD | | PRV | | -RW-- |
| 000001FAD45F0000 | 0000000000001000 | | | MAP | -R--- | -R--- |
| 000001FAD4600000 | 0000000000001000 | | | MAP | -R--- | -R--- |
| 00007FF4F0140000 | 0000000000005000 | | | MAP | -R--- | -R--- |
| 00007FF4F0145000 | 00000000000FB000 | Reserved (00007FF4F | | MAP | | -R--- |
| 00007FF4F0240000 | 0000000100020000 | Reserved | | PRV | | -RW-- |
| 00007FF5F0260000 | 0000000002000000 | Reserved | | PRV | | -RW-- |
| 00007FF5F2260000 | 0000000000001000 | | | PRV | -RW-- | -RW-- |
| 00007FF5F2270000 | 0000000000001000 | | | MAP | -R--- | -R--- |
| 00007FF5F2280000 | 0000000000033000 | | | MAP | -R--- | -R--- |
| 00007FF676150000 | 0000000000001000 | hello.exe | | IMG | -R--- | ERWC- |
| 00007FF676151000 | 0000000000002000 | ".text" | 可執行代碼 | IMG | ER--- | ERWC- |
| 00007FF676153000 | 0000000000002000 | ".rdata" | 唯讀的已初始化的資料 | IMG | -R--- | ERWC- |
| 00007FF676155000 | 0000000000001000 | ".data" | 已初始化的資料 | IMG | -RW-- | ERWC- |
| 00007FF676156000 | 0000000000001000 | ".pdata" | 例外資料 | IMG | -R--- | ERWC- |
| 00007FF676157000 | 0000000000001000 | ".rsrc" | 資源 | IMG | -R--- | ERWC- |
| 00007FF676158000 | 0000000000001000 | ".reloc" | Base relocations | IMG | -R--- | ERWC- |
| 00007FFB29680000 | 0000000000001000 | vcruntime140.dll | | IMG | -R--- | ERWC- |
| 00007FFB29681000 | 0000000000010000 | ".text" | 可執行代碼 | IMG | ER--- | ERWC- |
| 00007FFB29691000 | 0000000000004000 | ".rdata" | 唯讀的已初始化的資料 | IMG | -R--- | ERWC- |
| 00007FFB29695000 | 0000000000001000 | ".data" | 已初始化的資料 | IMG | -RW-- | ERWC- |
| 00007FFB29696000 | 0000000000001000 | ".pdata" | 例外資料 | IMG | -R--- | ERWC- |
| 00007FFB29697000 | 0000000000001000 | "_RDATA" | | IMG | -R--- | ERWC- |
| 00007FFB29698000 | 0000000000001000 | ".rsrc" | 資源 | IMG | -R--- | ERWC- |
| 00007FFB29699000 | 0000000000001000 | ".reloc" | Base relocations | IMG | -R--- | ERWC- |

# VA RVA

- 解釋 RVA (Relative Virtual Address) 之前
- 先解釋什麼是 VA (Virtual Address)
- 做一下小實驗，如果執行兩個 hello.exe，記憶體位址分布長怎樣?

# VA RVA

- 兩個 process 的記憶體位址有重疊耶?!
- 如果改掉 A process 記憶體內容 (地址重疊的部分), B process 的內容也會被改嗎?
- 實驗一下, 答案是不會的
- 所以那個記憶體位址到底是啥

# VA RVA

- 其實我們的程式所看到的記憶體位址，都是假的
- 都是虛擬記憶體位址 (Virtual Address)

# VA RVA

- 那為什麼要這麼複雜?
- 如果程式都能直接碰到實體記憶體位址 PA (Physical Address)
- 你要怎麼知道這個 PA 有沒有被其他程式占用?
- 這個問題很難，但現代 OS 幫你搞定了這個問題
- OS 只給你 VA，實際上存取時，OS 有他的方式，能夠 VA <-> PA

# VA RVA

- 正常狀況下 A process 的 0x55665566 VA
- 跟 B process 的 0x55665566 VA
- 不是對應到同一個 PA
- 解釋了剛剛的實驗結果

# VA RVA

- 搞懂 VA 了，可以講 RVA 了
- 只是一個方便 PE 結構不用寫這麼多字的東西
- VA = ImageBase + RVA
- 第一條指令位址 VA = ImageBase + Entry point RVA

# VA RVA

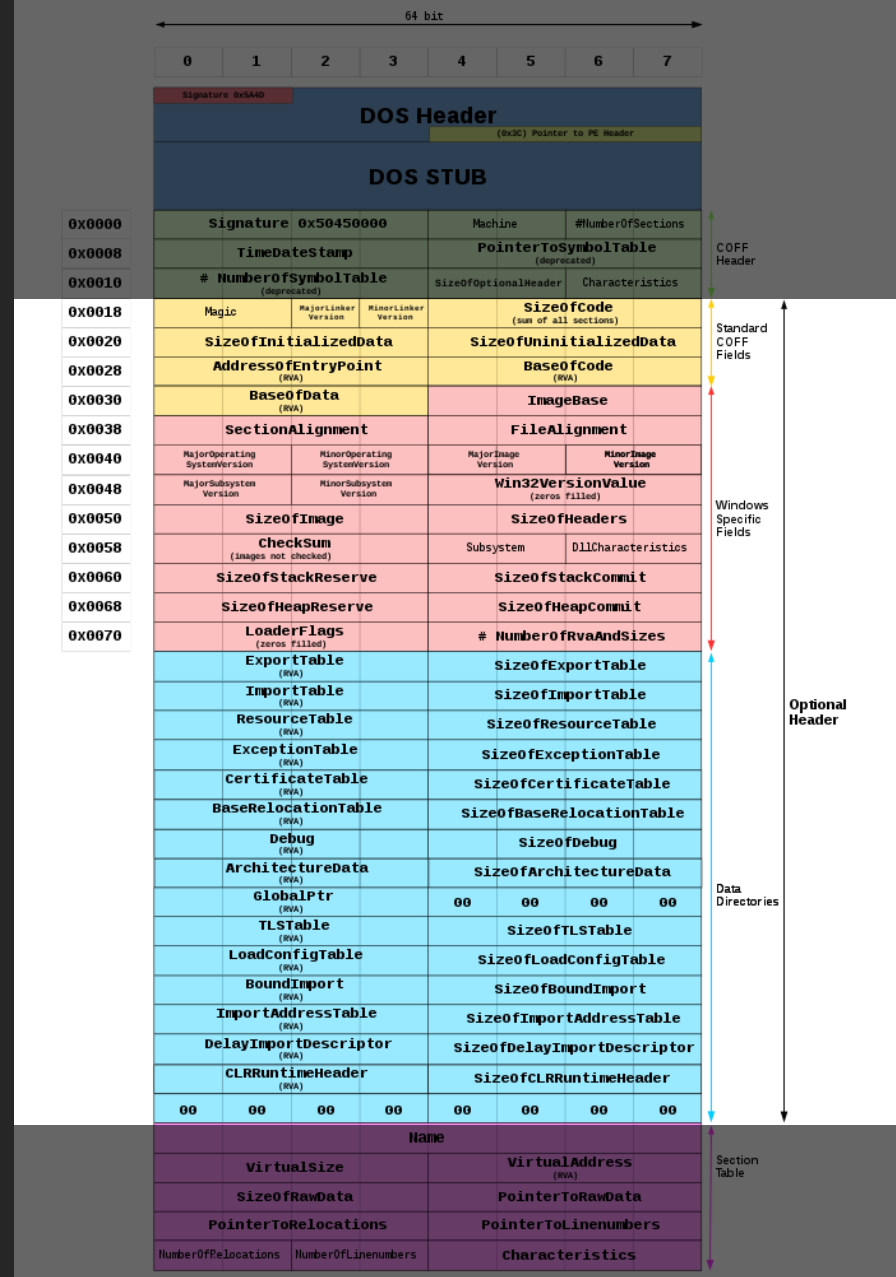
Disasm | General | DOS Hdr | Rich Hdr | File Hdr | Optional Hdr | Section Hdrs | Im

| Offset | Name | Value | Value |
|---|---|---|---|
| 118 | Magic | 20B | NT64 |
| 11A | Linker Ver. (Major) | E | |
| 11B | Linker Ver. (Minor) | 1D | |
| 11C | Size of Code | 1200 | |
| 120 | Size of Initialized Data | 2000 | |
| 124 | Size of Uninitialized Data | 0 | |
| 128 | Entry Point | 1580 | |
| 12C | Base of Code | 1000 | |
| 130 | Image Base | 140000000 | |
| 138 | Section Alignment | 1000 | |
| 13C | File Alignment | 200 | |
| 140 | OS Ver | | |
| 142 | OS Ver | | |
| 144 | Image Ver. (Major) | 0 | |
| 146 | Image Ver. (Minor) | 0 | |
| 148 | Subsystem Ver. (Major) | 6 | |
| 14A | Subsystem Ver. Minor) | 0 | |
| 14C | Win32 Version Value | 0 | |
| 150 | Size of Image | 9000 | |
| 154 | Size of Headers | 400 | |
| 158 | Checksum | 0 | |
| 15C | Subsystem | 3 | Windows console |
| 15E | DLL Characteristics | 8160 | |
| | | 40 | DLL can move |
| | | 100 | Image is NX compatible |
| | | 8000 | TerminalServer aware |

程式進入點 RVA

程式基址

第一條指令位址 = 0x140001580



# ASLR

- 可是實驗中, 我們的第一條指令位址顯然不是剛算的
- 原因是 ASLR (Address Space Layout Randomization)
- 記憶體位址每次執行時都是固定的話，會有安全問題
- 啟用了 ASLR, OS 就會隨機產生 ImageBase，原本提供的 ImageBase 就被忽略了 QQ

# ASLR

- 第一條指令位址 VA = ASLR 隨機產生的 ImageBase + 進入點 RVA

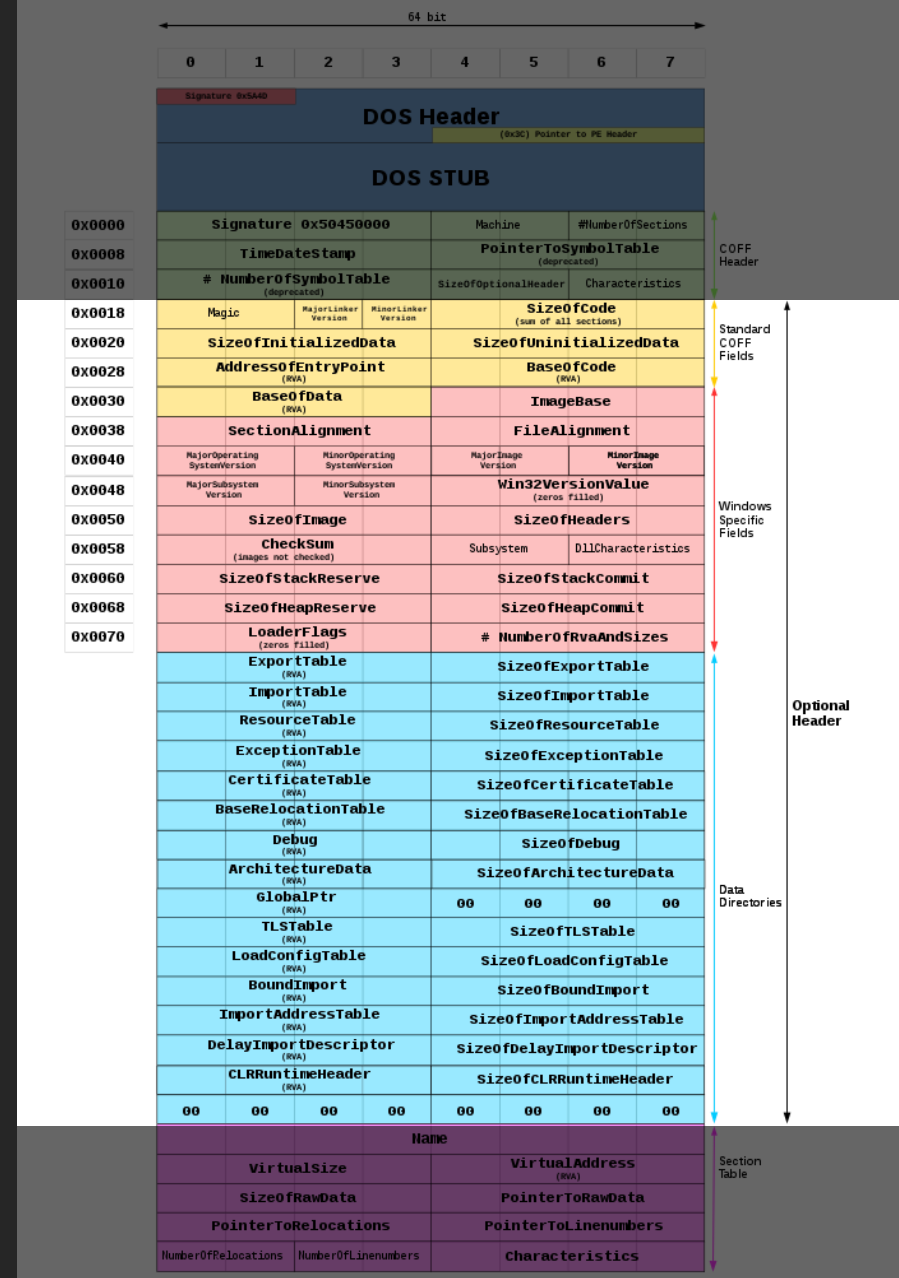
- 那麼要怎麼知道 ASLR 有沒有開啟呢?

# ASLR

Disasm | General | DOS Hdr | Rich Hdr | File Hdr | Optional Hdr | Section Hdrs | Im

| Offset | Name | Value | Value |
|---|---|---|---|
| 118 | Magic | 20B | NT64 |
| 11A | Linker Ver. (Major) | E | |
| 11B | Linker Ver. (Minor) | 1D | |
| 11C | Size of Code | 1200 | |
| 120 | Size of Initialized Data | 2000 | |
| 124 | Size of Uninitialized Data | 0 | |
| 128 | Entry Point | 1580 | |
| 12C | Base of Code | 1000 | |
| 130 | Image Base | 140000000 | |
| 138 | Section Alignment | 1000 | |
| 13C | File Alignment | 200 | |
| 140 | OS Ver. (Major) | 6 | Windows Vista / Server 2008 |
| 142 | OS Ver. (Minor) | 0 | |
| 144 | Image Ver. (Major) | 0 | |
| 146 | Image Ver. (Minor) | 0 | |
| 148 | Subsystem Ver. (Major) | 6 | |
| 14A | Subsystem Ver. Minor) | 0 | |
| 14C | Win32 Version Value | 0 | |
| 150 | Size of Image | 9000 | |
| 154 | Size of Headers | 400 | |
| 158 | Checksum | 0 | |
| 15C | Subsystem | 3 | Windows console |
| 15E | DLL Characteristics | 8160 | |
| | | 40 | DLL can move |
| | | 100 | Image is NX compatible |
| | | 8000 | TerminalServer aware |

若有 DLL can move 就是有啟用 ASLR



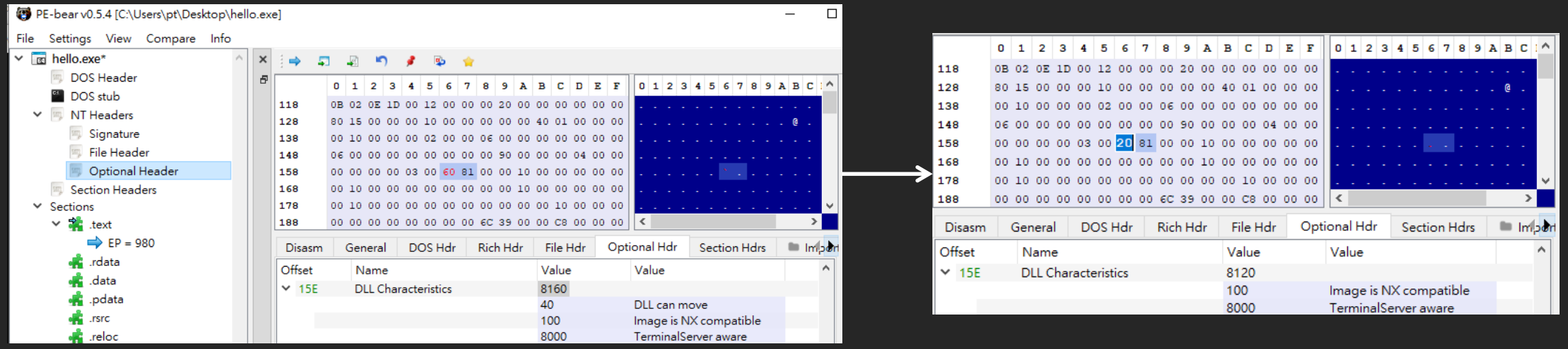
# ASLR

- 如果把程式裡的這個 bit 拔掉, 就可以關掉 ASLR 了

- 來實驗一下!

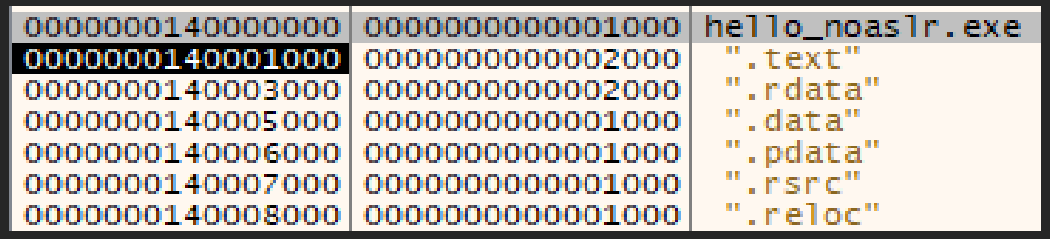
# ASLR

- DLL can move 是 0x40
- 減去 0x40 就是把它拔掉



# ASLR

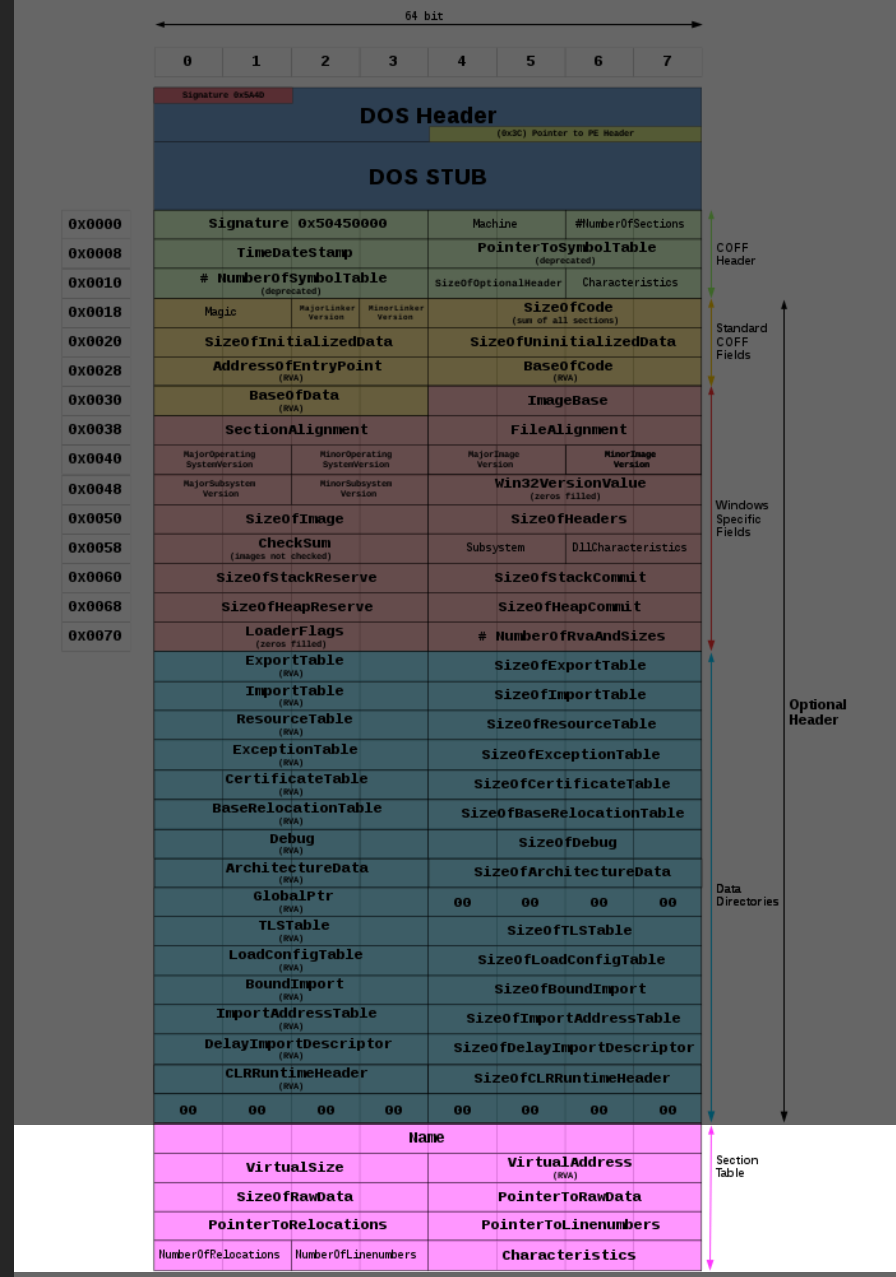
- 記得另存一個新程式
- 再度執行看看



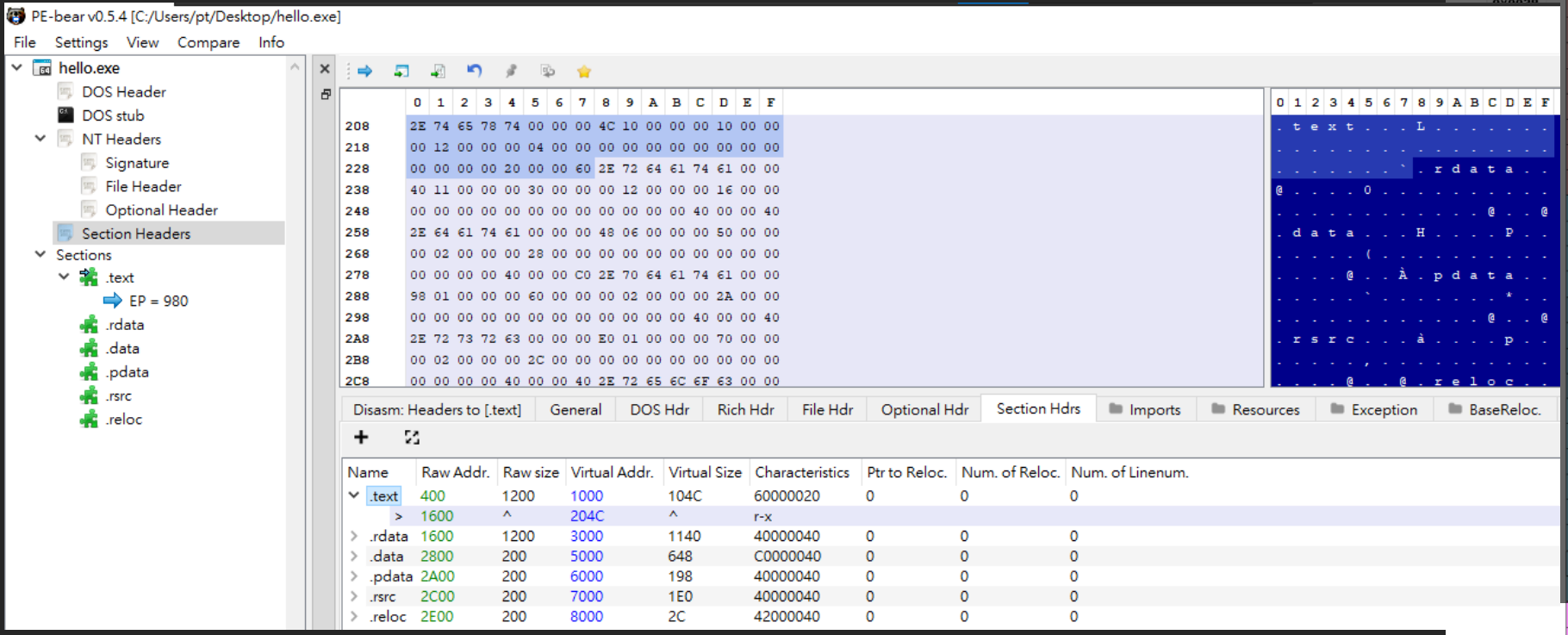
| 0000000140000000 | 0000000000001000 | hello_noaslr.exe |
|---|---|---|
| 0000000140001000 | 0000000000002000 | ".text" |
| 0000000140003000 | 0000000000002000 | ".rdata" |
| 0000000140005000 | 0000000000001000 | ".data" |
| 0000000140006000 | 0000000000001000 | ".pdata" |
| 0000000140007000 | 0000000000001000 | ".rsrc" |
| 0000000140008000 | 0000000000001000 | ".reloc" |

# PE format

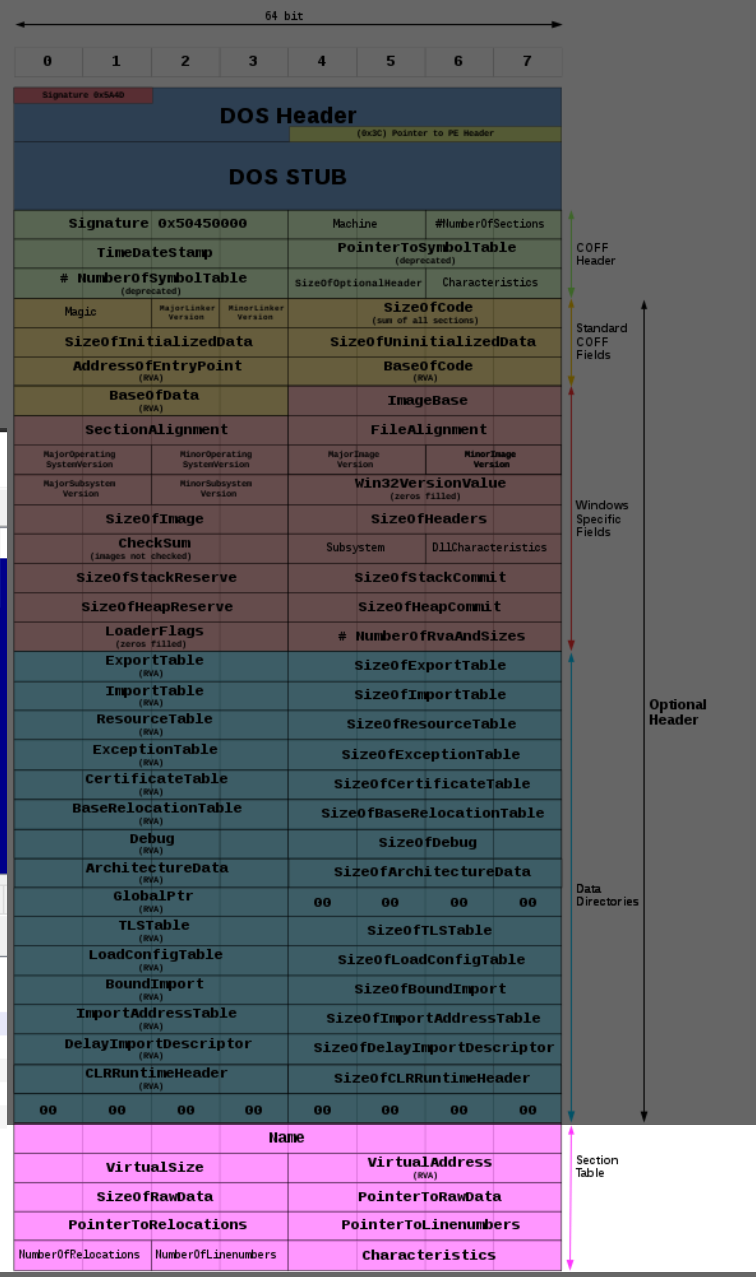
- Section Hdr
- 在右邊的圖是對應 Section Table
- 其實會有多個 Section Hdr
- 用 PE-Bear 來展示一下



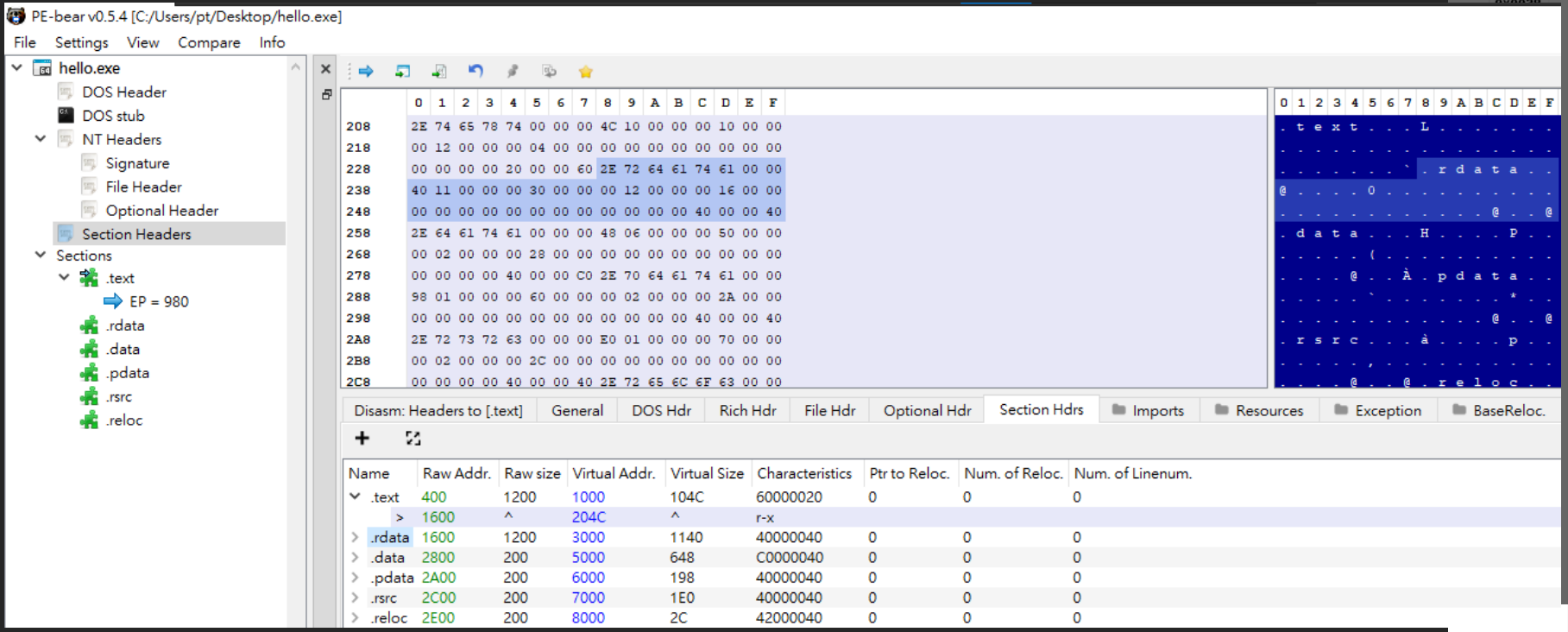
# PE format



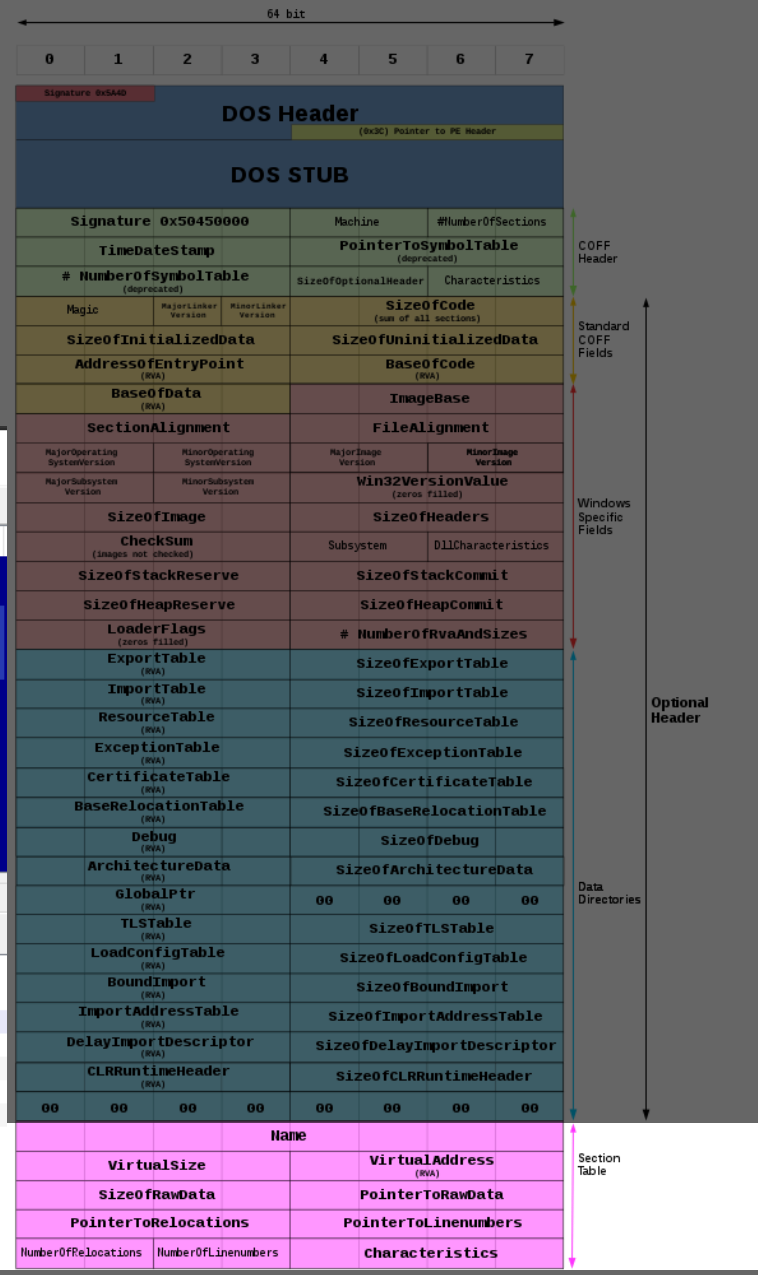

.text section hdr



# PE format




.rdata section hdr



# PE format

- 解釋一下 Section Hdr

- 程式檔案的 Raw Addr 開始的 Raw size 個 Bytes 會映射到 Virtual Addr 開始的 Virtual Size 個 Bytes
  - Virtual Addr 是 RVA

- Characteristics 設定了該區記憶體位址的權限

| Name | Raw Addr. | Raw size | Virtual Addr. | Virtual Size | Characteristics | Ptr to Reloc. | Num. of Reloc. | Num. of Linenum. |
|---|---|---|---|---|---|---|---|---|
| .text | 400 | 1200 | 1000 | 104C | 60000020 | 0 | 0 | 0 |
| > | 1600 | ^ | 204C | ^ | r-x | | | |
| .rdata | 1600 | 1200 | 3000 | 1140 | 40000040 | 0 | 0 | 0 |
| .data | 2800 | 200 | 5000 | 648 | C0000040 | 0 | 0 | 0 |
| .pdata | 2A00 | 200 | 6000 | 198 | 40000040 | 0 | 0 | 0 |
| .rsrc | 2C00 | 200 | 7000 | 1E0 | 40000040 | 0 | 0 | 0 |
| .reloc | 2E00 | 200 | 8000 | 2C | 42000040 | 0 | 0 | 0 |

# PE format

| Name | Raw Addr. | Raw size | Virtual Addr. | Virtual Size | Characteristics | Ptr to Reloc. | Num. of Reloc. | Num. of Linenum. |
|---|---|---|---|---|---|---|---|---|
| .text | 400 | 1200 | 1000 | 104C | 60000020 | 0 | 0 | 0 |
| > | 1600 | ^ | 204C | ^ | r-x | | | |
| .rdata | 1600 | 1200 | 3000 | 1140 | 40000040 | 0 | 0 | 0 |
| .data | 2800 | 200 | 5000 | 648 | C0000040 | 0 | 0 | 0 |
| .pdata | 2A00 | 200 | 6000 | 198 | 40000040 | 0 | 0 | 0 |
| .rsrc | 2C00 | 200 | 7000 | 1E0 | 40000040 | 0 | 0 | 0 |
| .reloc | 2E00 | 200 | 8000 | 2C | 42000040 | 0 | 0 | 0 |

hello.exe ×

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | 0123456789ABCDEF |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0400h: | 48 | 8D | 05 | 29 | 46 | 00 | 00 | C3 | CC | CC | CC | CC | CC | CC | CC | CC | H..)F..ÃÌÌÌÌÌÌÌÌ |
| 0410h: | 48 | 8D | 05 | 11 | 46 | 00 | 00 | C3 | CC | CC | CC | CC | CC | CC | CC | CC | H...F..ÃÌÌÌÌÌÌÌÌ |
| 0420h: | 48 | 89 | 4C | 24 | 08 | 48 | 89 | 54 | 24 | 10 | 4C | 89 | 44 | 24 | 18 | 4C | H‰L$.H‰T$.L‰D$.L |
| 0430h: | 89 | 4C | 24 | 20 | 53 | 56 | 57 | 48 | 83 | EC | 30 | 48 | 8B | F9 | 48 | 8D | ‰L$ SVWHƒì0H‹ùH. |

| 位址 | 十六進位 | | | | | | | | | | | | | | | | ASCII |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0000000140001000 | 48 | 8D | 05 | 29 | 46 | 00 | 00 | C3 | CC | CC | CC | CC | CC | CC | CC | CC | H..)F..ÃÌÌÌÌÌÌÌÌ |
| 0000000140001010 | 48 | 8D | 05 | 11 | 46 | 00 | 00 | C3 | CC | CC | CC | CC | CC | CC | CC | CC | H...F..ÃÌÌÌÌÌÌÌÌ |
| 0000000140001020 | 48 | 89 | 4C | 24 | 08 | 48 | 89 | 54 | 24 | 10 | 4C | 89 | 44 | 24 | 18 | 4C | H.L$.H.T$.L.D$.L |
| 0000000140001030 | 89 | 4C | 24 | 20 | 53 | 56 | 57 | 48 | 83 | EC | 30 | 48 | 8B | F9 | 48 | 8D | .L$ SVWH.ì0H.ùH. |

# PE format

- 但看了一下 .rdata，好像不是這麼回事?

# PE format

| Name | Raw Addr. | Raw size | Virtual Addr. | Virtual Size | Characteristics | Ptr to Reloc. | Num. of Reloc. | Num. of Linenum. |
|---|---|---|---|---|---|---|---|---|
| .text | 400 | 1200 | 1000 | 104C | 60000020 | 0 | 0 | 0 |
| > | 1600 | ^ | 204C | ^ | r-x | | | |
| .rdata | 1600 | 1200 | 3000 | 1140 | 40000040 | 0 | 0 | 0 |
| .data | 2800 | 200 | 5000 | 648 | C0000040 | 0 | 0 | 0 |
| .pdata | 2A00 | 200 | 6000 | 198 | 40000040 | 0 | 0 | 0 |
| .rsrc | 2C00 | 200 | 7000 | 1E0 | 40000040 | 0 | 0 | 0 |
| .reloc | 2E00 | 200 | 8000 | 2C | 42000040 | 0 | 0 | 0 |

hello.exe ×

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1600h: | 28 | 3C | 00 | 00 | 00 | 00 | 00 | 00 | 34 | 3C | 00 | 00 | 00 | 00 | 00 | 00 |
| 1610h: | 44 | 3C | 00 | 00 | 00 | 00 | 00 | 00 | 52 | 3C | 00 | 00 | 00 | 00 | 00 | 00 |
| 1620h: | 60 | 3C | 00 | 00 | 00 | 00 | 00 | 00 | 18 | 41 | 00 | 00 | 00 | 00 | 00 | 00 |
| 1630h: | 02 | 41 | 00 | 00 | 00 | 00 | 00 | 00 | E8 | 40 | 00 | 00 | 00 | 00 | 00 | 00 |

| 位址 | 十六進位 | | | | ASCII |
|---|---|---|---|---|---|
| 0000000140003000 | D0 4F 0F 38 FB 7F 00 00 | 30 4E 0F 38 FB 7F 00 00 | ÐO.8û...0N.8û... |
| 0000000140003010 | 50 4B 0F 38 FB 7F 00 00 | E0 48 0F 38 FB 7F 00 00 | PK.8û...àH.8û... |
| 0000000140003020 | 10 4E 0F 38 FB 7F 00 00 | B0 01 0F 38 FB 7F 00 00 | .N.8û...°..8û... |
| 0000000140003030 | A0 EA 51 39 FB 7F 00 00 | 80 7B 0E 38 FB 7F 00 00 | êQ9û....{.8û... |

# PE format

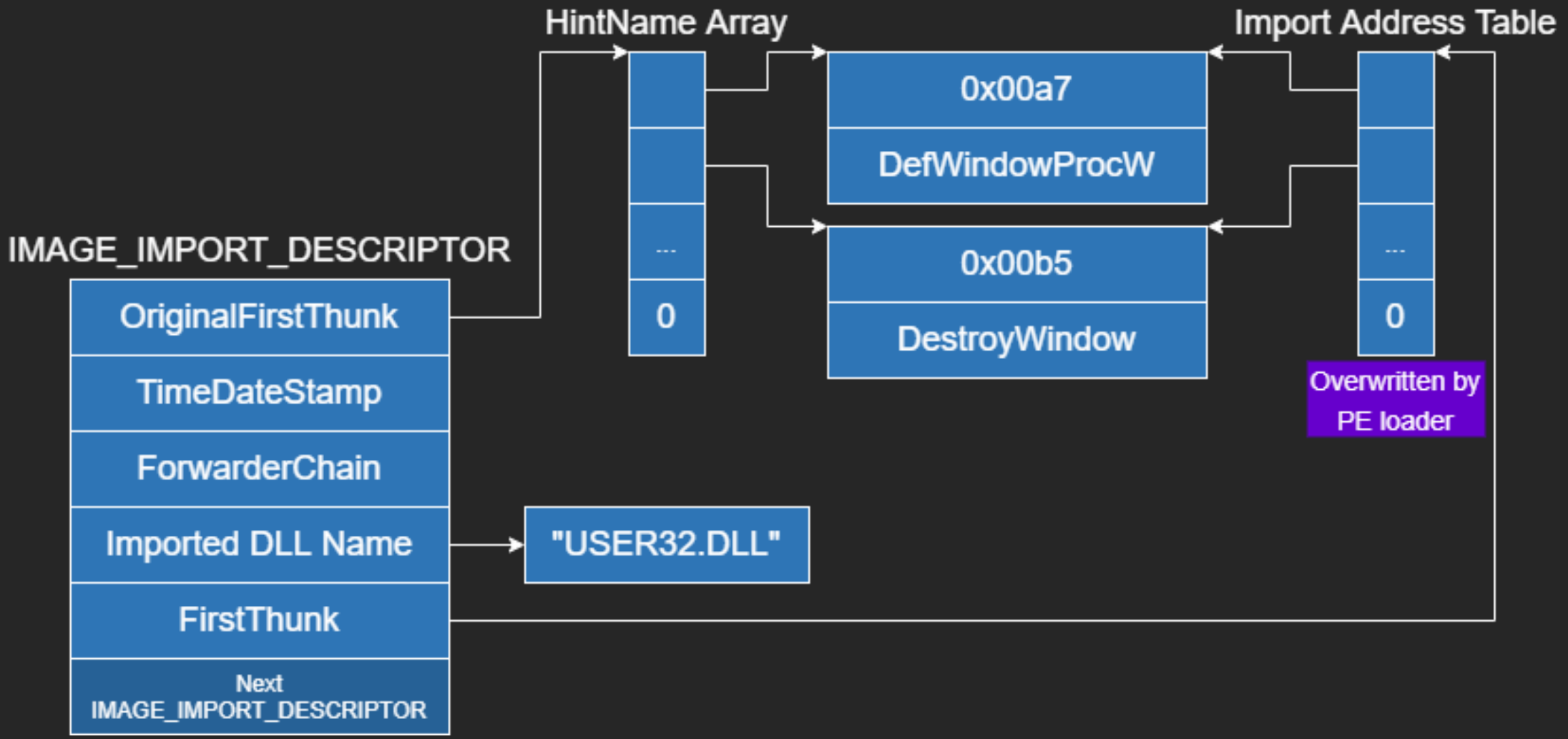
- 但看了一下 .rdata, 好像不是這麼回事?

- 實際上是因為另一個機制, 改掉了這邊的資料

# IAT

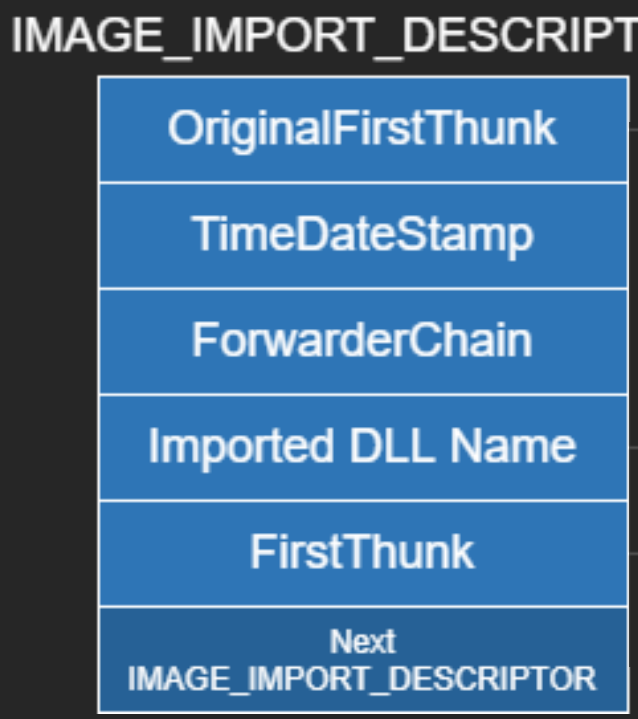
Import Address Table

# IAT




Ref: https://docs.microsoft.com/en-us/archive/msdn-magazine/2002/march/inside-windows-an-in-depth-look-into-the-win32-portable-executable-file-format-part-2

# IAT



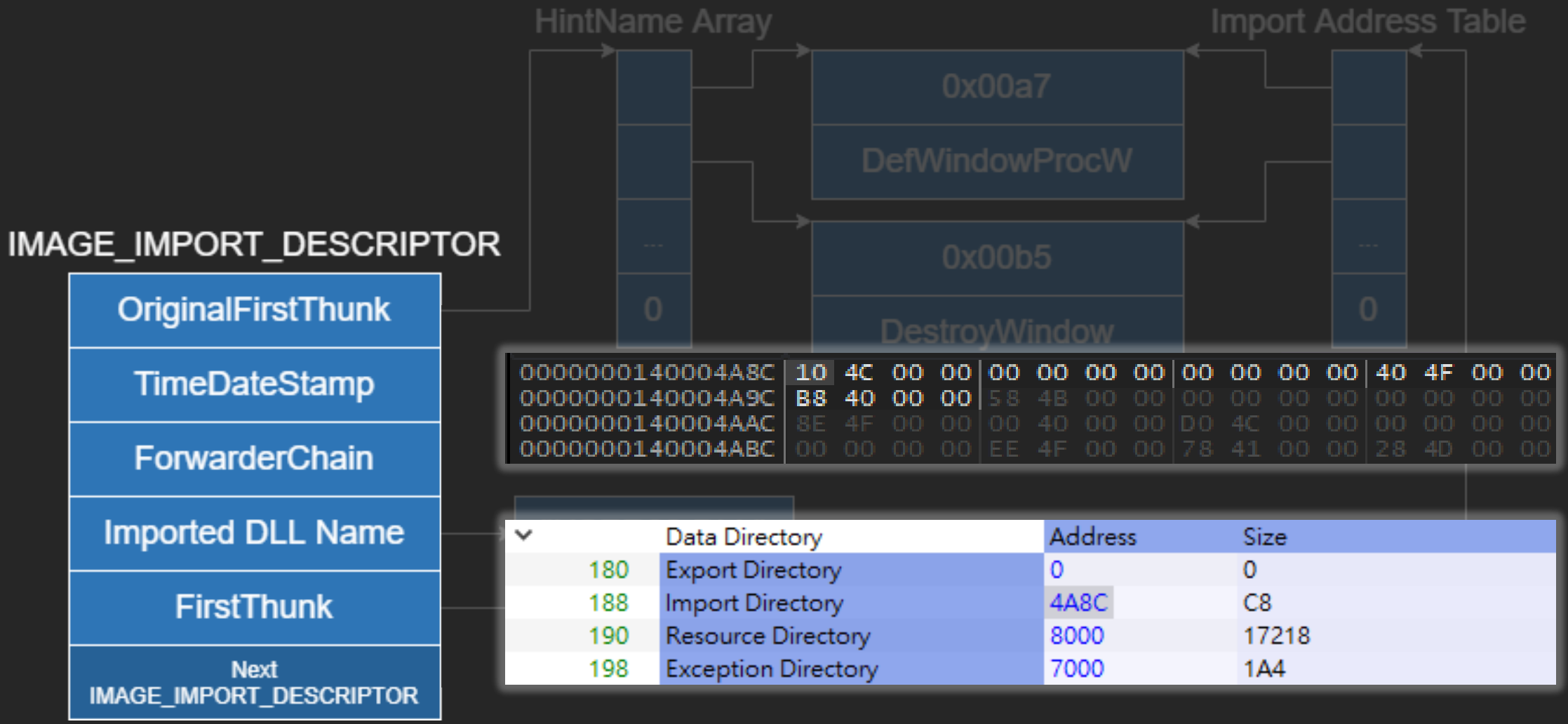

Disasm: .rdata | General | DOS Hdr | Rich Hdr | File Hdr | Optional Hdr | Section H

| Offset | Name | Value | Value |
|---|---|---|---|
| 134 | File Alignment | 200 | |
| 138 | OS Ver. (Major) | 6 | Windows Vista / Server 2008 |
| 13A | OS Ver. (Minor) | 0 | |
| 13C | Image Ver. (Major) | 0 | |
| 13E | Image Ver. (Minor) | 0 | |
| 140 | Subsystem Ver. (Major) | 6 | |
| 142 | Subsystem Ver. Minor) | 0 | |
| 144 | Win32 Version Value | 0 | |
| 148 | Size of Image | 21000 | |
| 14C | Size of Headers | 400 | |
| 150 | Checksum | 0 | |
| 154 | Subsystem | 2 | Windows GUI |
| 156 | DLL Characteristics | 8120 | |
| | | 100 | Image is NX compatible |
| | | 8000 | TerminalServer aware |
| 158 | Size of Stack Reserve | 100000 | |
| 160 | Size of Stack Commit | 1000 | |
| 168 | Size of Heap Reserve | 100000 | |
| 170 | Size of Heap Commit | 1000 | |
| 178 | Loader Flags | 0 | |
| 17C | Number of RVAs and Sizes | 10 | |

| | Data Directory | Address | Size |
|---|---|---|---|
| 180 | Export Directory | 0 | 0 |
| 188 | Import Directory | 4A8C | C8 |
| 190 | Resource Directory | 8000 | 17218 |
| 198 | Exception Directory | 7000 | 1A4 |

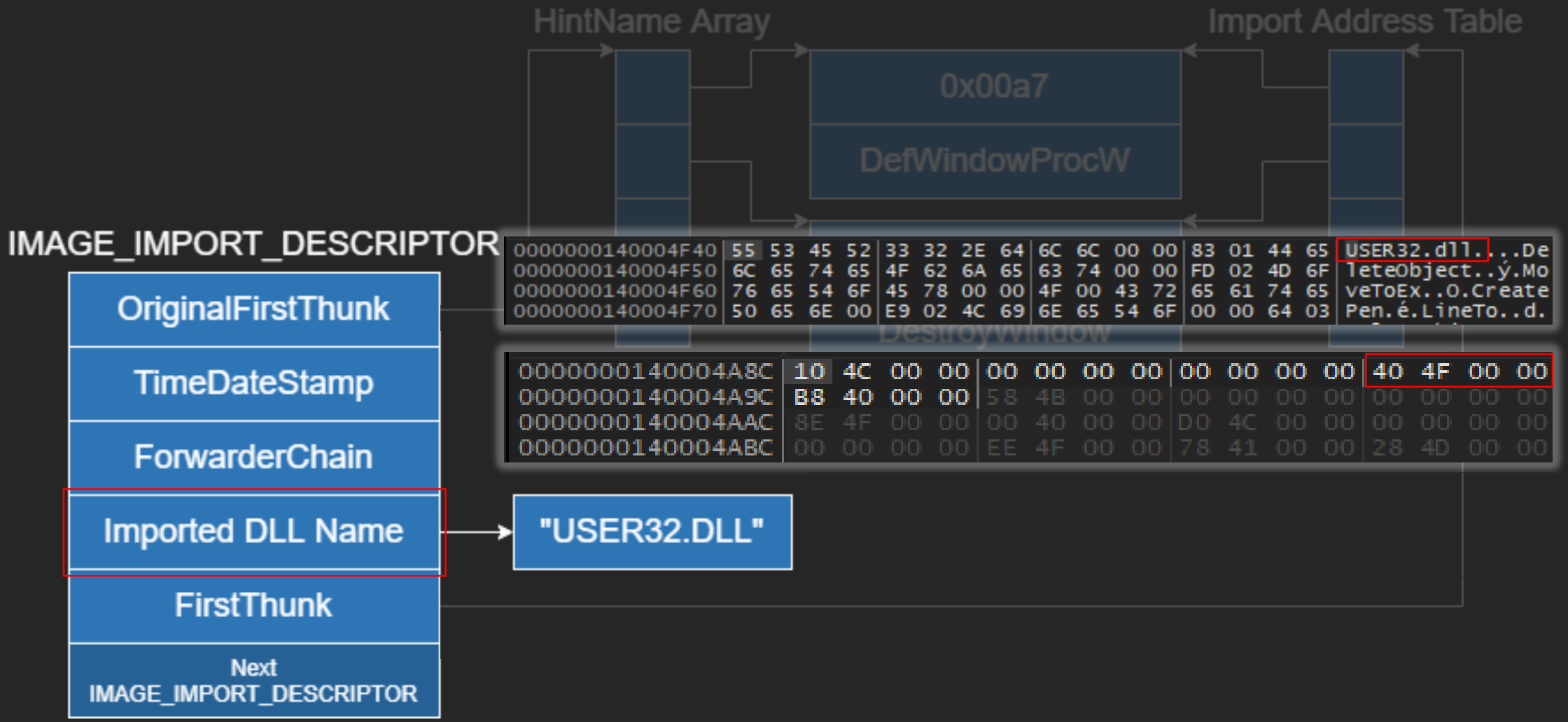
Ref: https://docs.microsoft.com/en-us/archive/msdn-magazine/2002/march/inside-windows-an-in-depth-look-into-the-win32-portable-executable-file-format-part-2

# IAT



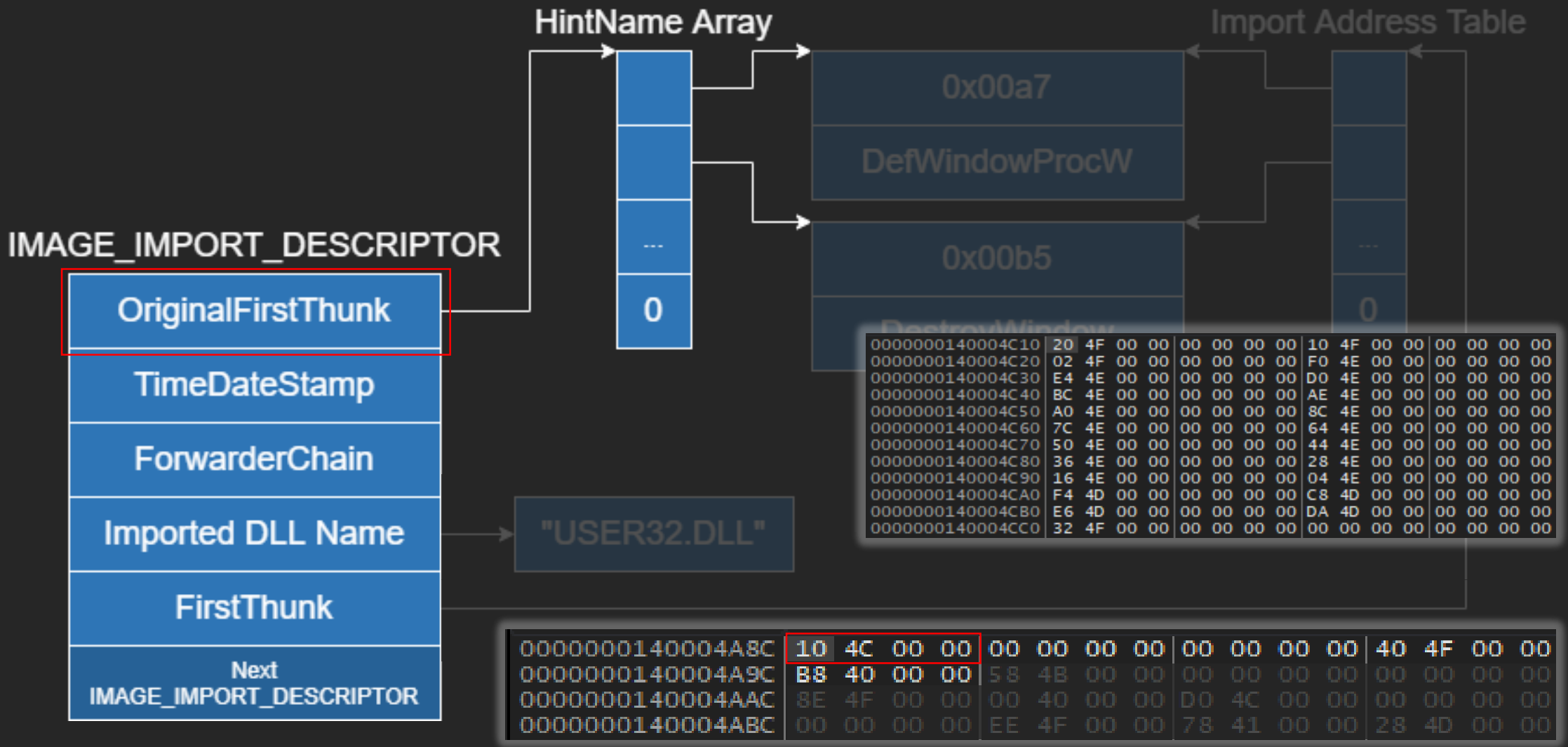

Ref: https://docs.microsoft.com/en-us/archive/msdn-magazine/2002/march/inside-windows-an-in-depth-look-into-the-win32-portable-executable-file-format-part-2

# IAT



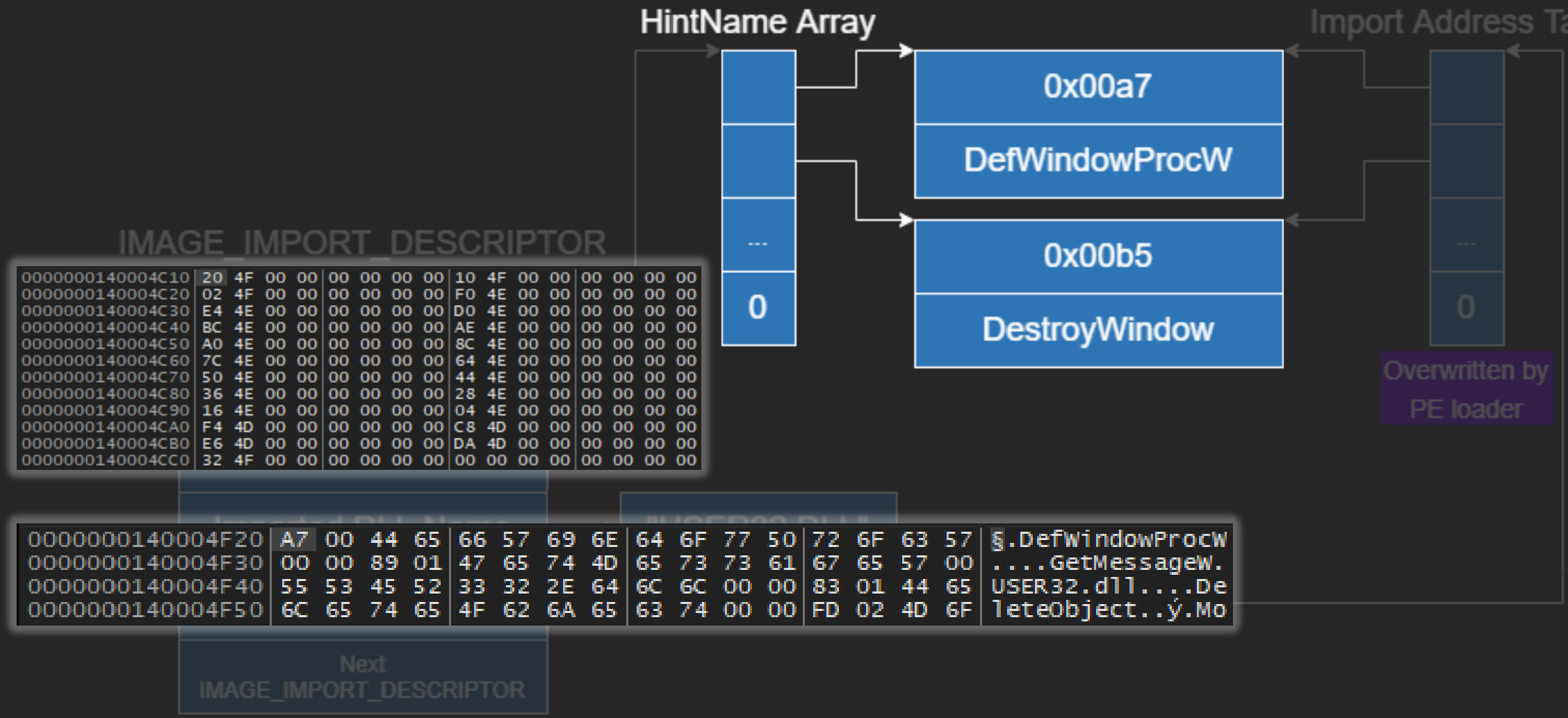

Ref: https://docs.microsoft.com/en-us/archive/msdn-magazine/2002/march/inside-windows-an-in-depth-look-into-the-win32-portable-executable-file-format-part-2

# IAT




Ref: https://docs.microsoft.com/en-us/archive/msdn-magazine/2002/march/inside-windows-an-in-depth-look-into-the-win32-portable-executable-file-format-part-2

# IAT

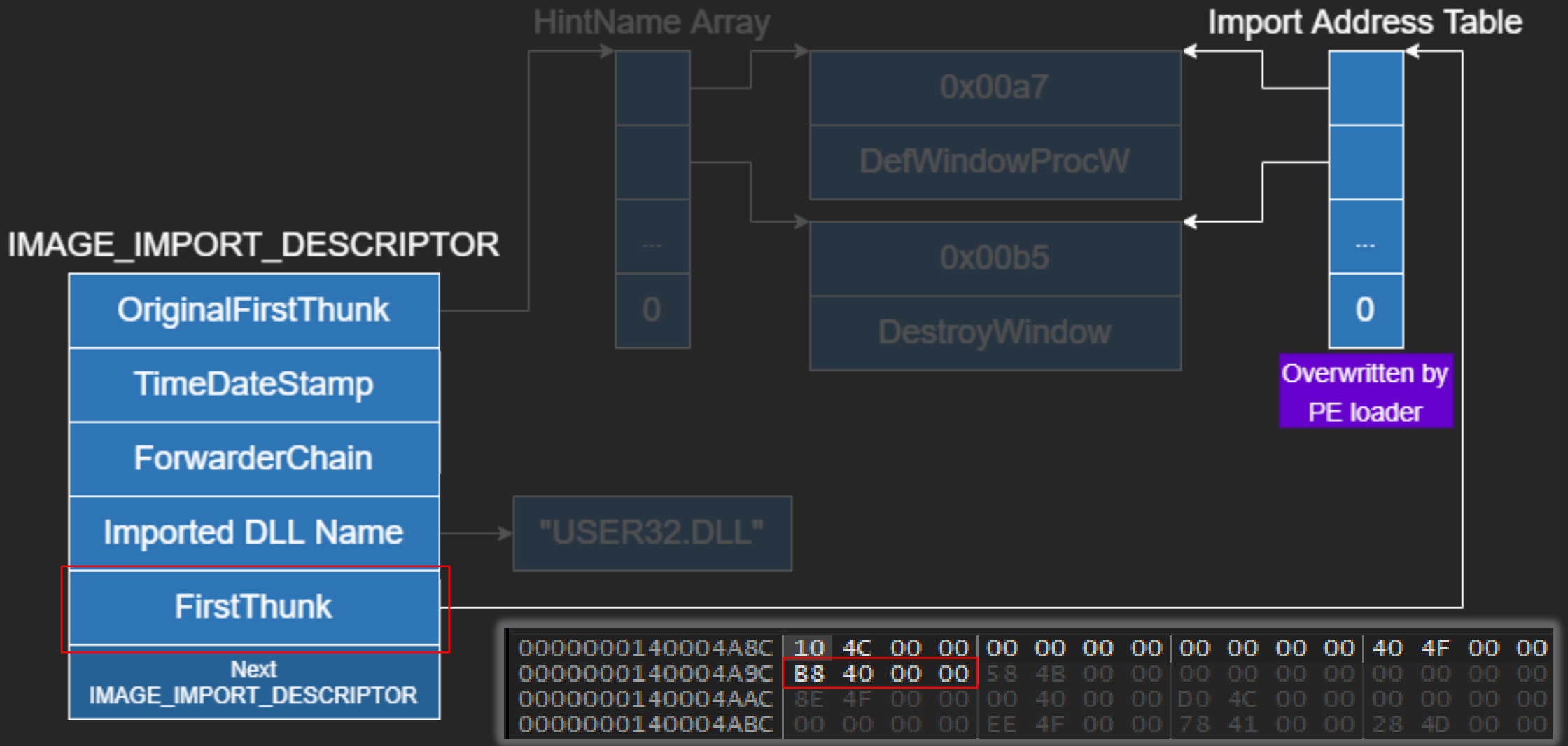



```
0000000140004C10  20 4F 00 00 00 00 00 00 10 4F 00 00 00 00 00 00
0000000140004C20  02 4F 00 00 00 00 00 00 F0 4E 00 00 00 00 00 00
0000000140004C30  E4 4E 00 00 00 00 00 00 D0 4E 00 00 00 00 00 00
0000000140004C40  BC 4E 00 00 00 00 00 00 AE 4E 00 00 00 00 00 00
0000000140004C50  A0 4E 00 00 00 00 00 00 8C 4E 00 00 00 00 00 00
0000000140004C60  7C 4E 00 00 00 00 00 00 64 4E 00 00 00 00 00 00
0000000140004C70  50 4E 00 00 00 00 00 00 44 4E 00 00 00 00 00 00
0000000140004C80  36 4E 00 00 00 00 00 00 28 4E 00 00 00 00 00 00
0000000140004C90  16 4E 00 00 00 00 00 00 04 4E 00 00 00 00 00 00
0000000140004CA0  F4 4D 00 00 00 00 00 00 C8 4D 00 00 00 00 00 00
0000000140004CB0  E6 4D 00 00 00 00 00 00 DA 4D 00 00 00 00 00 00
0000000140004CC0  32 4F 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

```
0000000140004F20  A7 00 44 65 66 57 69 6E 64 6F 77 50 72 6F 63 57  §.DefWindowProcW
0000000140004F30  00 00 89 01 47 65 74 4D 65 73 73 61 67 65 57 00  ....GetMessageW.
0000000140004F40  55 53 45 52 33 32 2E 64 6C 6C 00 00 83 01 44 65  USER32.dll....De
0000000140004F50  6C 65 74 65 4F 62 6A 65 63 74 00 00 FD 02 4D 6F  leteObject..ý.Mo
```

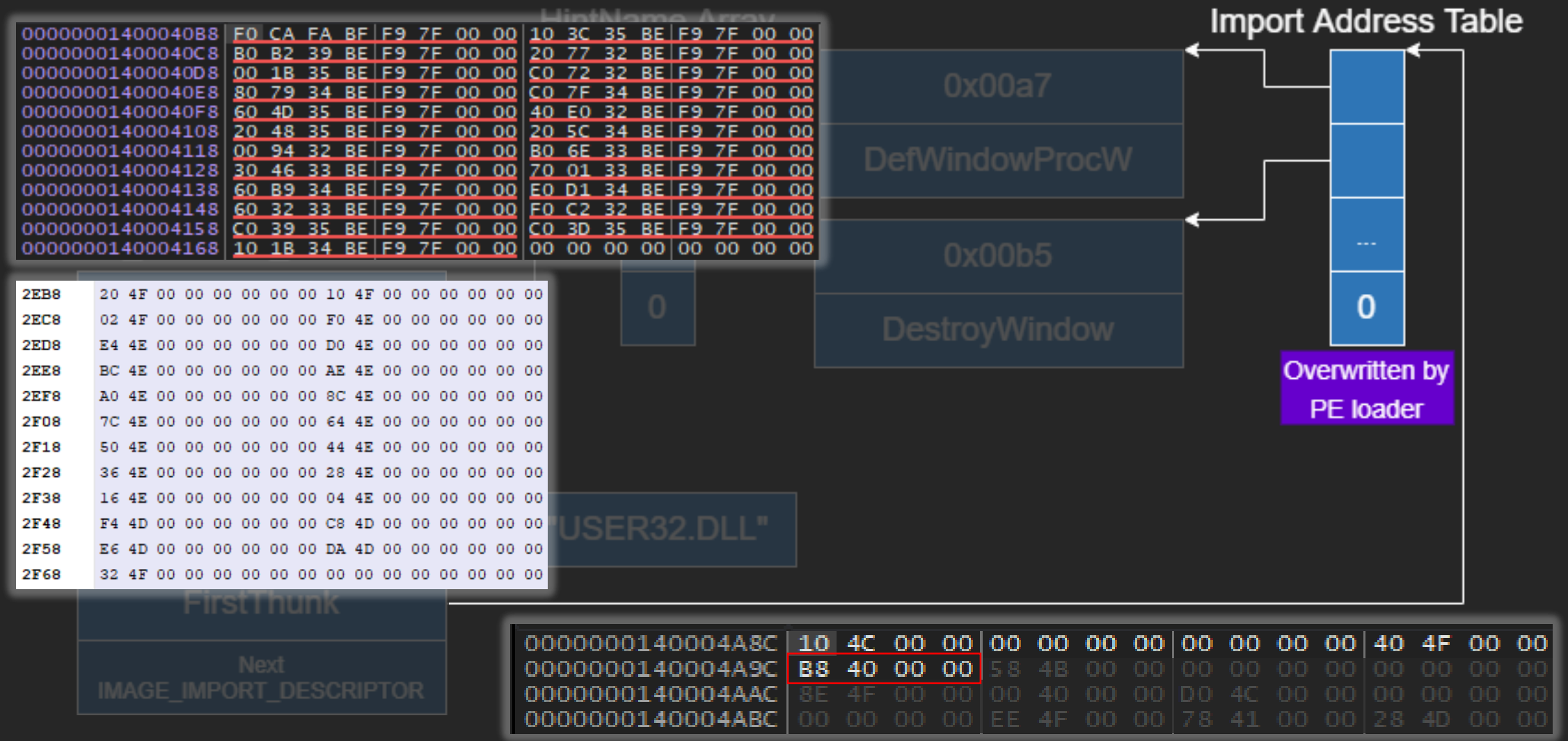
Ref: https://docs.microsoft.com/en-us/archive/msdn-magazine/2002/march/inside-windows-an-in-depth-look-into-the-win32-portable-executable-file-format-part-2

# IAT




Ref: https://docs.microsoft.com/en-us/archive/msdn-magazine/2002/march/inside-windows-an-in-depth-look-into-the-win32-portable-executable-file-format-part-2

# IAT




| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0000000140004088 | F0 | CA | FA | BF | F9 | 7F | 00 | 00 | 10 | 3C | 35 | BE | F9 | 7F | 00 | 00 |
| 0000000140004098 | B0 | B2 | 39 | BE | F9 | 7F | 00 | 00 | 20 | 77 | 32 | BE | F9 | 7F | 00 | 00 |
| 00000001400040A8 | 00 | 1B | 35 | BE | F9 | 7F | 00 | 00 | C0 | 72 | 32 | BE | F9 | 7F | 00 | 00 |
| 00000001400040B8 | 80 | 79 | 34 | BE | F9 | 7F | 00 | 00 | C0 | 7F | 34 | BE | F9 | 7F | 00 | 00 |
| 00000001400040C8 | 60 | 4D | 35 | BE | F9 | 7F | 00 | 00 | 40 | E0 | 32 | BE | F9 | 7F | 00 | 00 |
| 00000001400040D8 | 20 | 48 | 35 | BE | F9 | 7F | 00 | 00 | 20 | 5C | 34 | BE | F9 | 7F | 00 | 00 |
| 00000001400040E8 | 00 | 94 | 32 | BE | F9 | 7F | 00 | 00 | B0 | 6E | 33 | BE | F9 | 7F | 00 | 00 |
| 00000001400040F8 | 30 | 46 | 33 | BE | F9 | 7F | 00 | 00 | 70 | 01 | 33 | BE | F9 | 7F | 00 | 00 |
| 0000000140004108 | 60 | B9 | 34 | BE | F9 | 7F | 00 | 00 | E0 | D1 | 34 | BE | F9 | 7F | 00 | 00 |
| 0000000140004118 | 60 | 32 | 33 | BE | F9 | 7F | 00 | 00 | F0 | C2 | 32 | BE | F9 | 7F | 00 | 00 |
| 0000000140004128 | C0 | 39 | 35 | BE | F9 | 7F | 00 | 00 | C0 | 3D | 35 | BE | F9 | 7F | 00 | 00 |
| 0000000140004138 | 10 | 1B | 34 | BE | F9 | 7F | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2EB8 | 20 | 4F | 00 | 00 | 00 | 00 | 00 | 00 | 10 | 4F | 00 | 00 | 00 | 00 | 00 | 00 |
| 2EC8 | 02 | 4F | 00 | 00 | 00 | 00 | 00 | 00 | F0 | 4E | 00 | 00 | 00 | 00 | 00 | 00 |
| 2ED8 | E4 | 4E | 00 | 00 | 00 | 00 | 00 | 00 | D0 | 4E | 00 | 00 | 00 | 00 | 00 | 00 |
| 2EE8 | BC | 4E | 00 | 00 | 00 | 00 | 00 | 00 | AE | 4E | 00 | 00 | 00 | 00 | 00 | 00 |
| 2EF8 | A0 | 4E | 00 | 00 | 00 | 00 | 00 | 00 | 8C | 4E | 00 | 00 | 00 | 00 | 00 | 00 |
| 2F08 | 7C | 4E | 00 | 00 | 00 | 00 | 00 | 00 | 64 | 4E | 00 | 00 | 00 | 00 | 00 | 00 |
| 2F18 | 50 | 4E | 00 | 00 | 00 | 00 | 00 | 00 | 44 | 4E | 00 | 00 | 00 | 00 | 00 | 00 |
| 2F28 | 36 | 4E | 00 | 00 | 00 | 00 | 00 | 00 | 28 | 4E | 00 | 00 | 00 | 00 | 00 | 00 |
| 2F38 | 16 | 4E | 00 | 00 | 00 | 00 | 00 | 00 | 04 | 4E | 00 | 00 | 00 | 00 | 00 | 00 |
| 2F48 | F4 | 4D | 00 | 00 | 00 | 00 | 00 | 00 | C8 | 4D | 00 | 00 | 00 | 00 | 00 | 00 |
| 2F58 | E6 | 4D | 00 | 00 | 00 | 00 | 00 | 00 | DA | 4D | 00 | 00 | 00 | 00 | 00 | 00 |
| 2F68 | 32 | 4F | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |

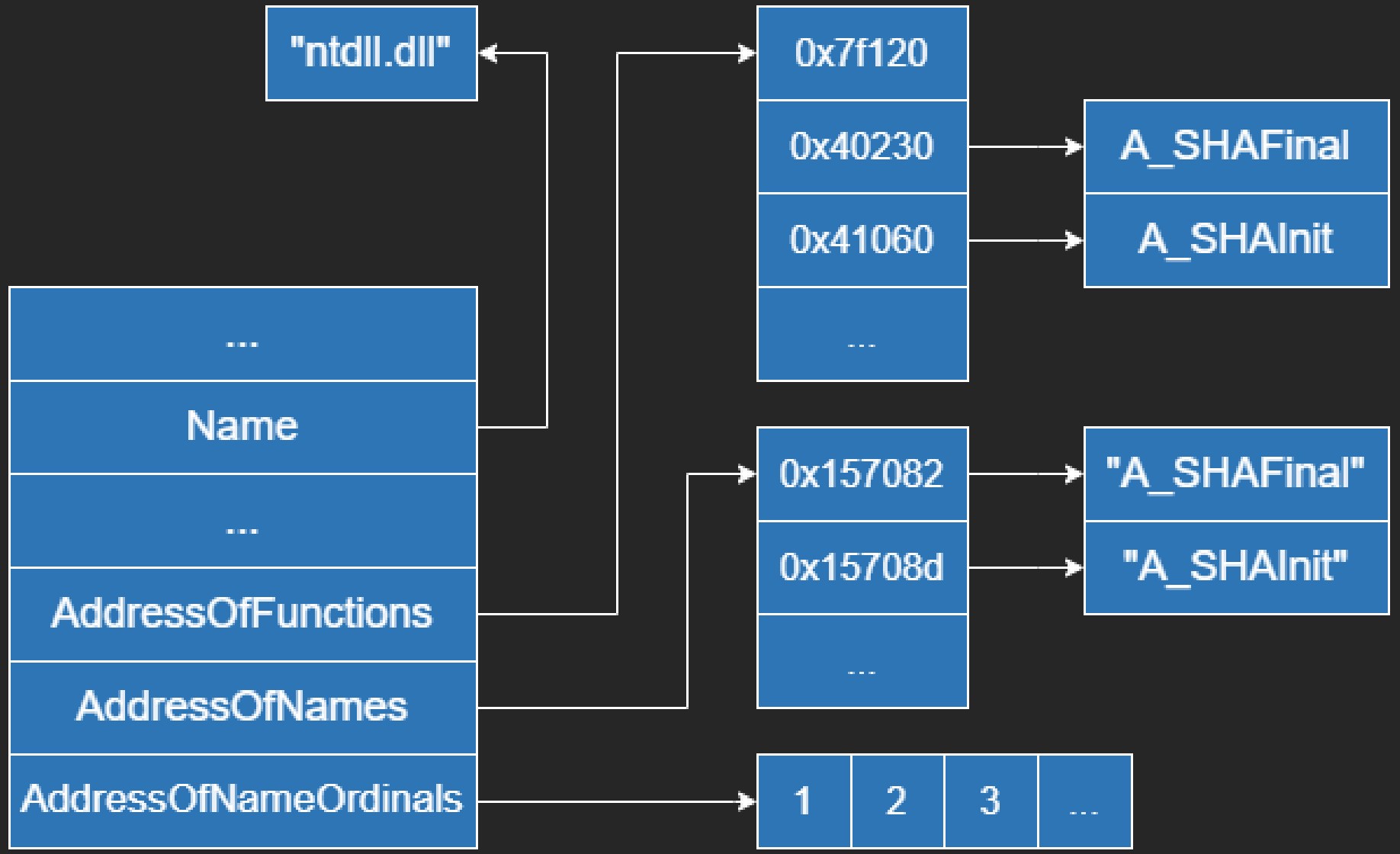
| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0000000140004A8C | 10 | 4C | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 40 | 4F | 00 | 00 |
| 0000000140004A9C | B8 | 40 | 00 | 00 | 58 | 4B | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0000000140004AAC | 8E | 4F | 00 | 00 | 00 | 40 | 00 | 00 | D0 | 4C | 00 | 00 | 00 | 00 | 00 | 00 |
| 0000000140004ABC | 00 | 00 | 00 | 00 | EE | 4F | 00 | 00 | 78 | 41 | 00 | 00 | 28 | 4D | 00 | 00 |

Ref: https://docs.microsoft.com/en-us/archive/msdn-magazine/2002/march/inside-windows-an-in-depth-look-into-the-win32-portable-executable-file-format-part-2

# EAT

Export Address Table

# EAT


"ntdll.dll"
...
Name
...
AddressOfFunctions
AddressOfNames
AddressOfNameOrdinals
0x7f120
0x40230
0x41060
...
A_SHAFinal
A_SHAInit
0x157082
0x15708d
...
"A_SHAFinal"
"A_SHAInit"
1
2
3
...

# EAT

"ntdll.dll"

| | Data Directory | Address | Size |
|---|---|---|---|
| 170 | Export Directory | 151160 | 12E71 |

Disasm: .rdata | General | DOS Hdr | Rich Hdr | File Hdr | Optional Hdr | Section Hdrs

| Name | Raw Addr. | Raw size | Virtual Addr. | Virtual Size | Characteristics | Ptr to Reloc. | Num. of |
|---|---|---|---|---|---|---|---|
| .text | 400 | 119000 | 1000 | 118F1E | 60000020 | 0 | 0 |
| PAGE | 119400 | 600 | 11A000 | 592 | 60000020 | 0 | 0 |
| RT | 119A00 | 200 | 11B000 | 1F9 | 60000020 | 0 | 0 |
| .rdata | 119C00 | 48000 | 11C000 | 47FD1 | 40000040 | 0 | 0 |

- ...
- Name
- ...
- AddressOfFunctions
- AddressOfNames
- AddressOfNameOrdinals

```
00007FFE1ED90000 0000000000001000 ntdll.dll
00007FFE1ED91000 0000000000119000  ".text"
00007FFE1EEAA000 0000000000001000  "PAGE"
00007FFE1EEAB000 0000000000001000  "RT"
00007FFE1EEAC000 0000000000048000  ".rdata"
```

```
>>> rdata = 0x11C000
>>> rdata_base = 0x0007FFE1EEAC000
>>> export_dir = 0x151160
>>> hex(export_dir - rdata + rdata_base)
'0x7ffe1eee1160'
```

# EAT

```
00007FFE1EEE1160  00 00 00 00 23 3C 2C BD 00 00 00 00 78 70 15 00
00007FFE1EEE1170  08 00 00 00 7F 09 00 00 7E 09 00 00 88 11 15 00
00007FFE1EEE1180  84 37 15 00 7C 5D 15 00 20 F1 07 00 30 02 04 00
```

Disasm: .rdata | General | DOS Hdr | Rich Hdr | File Hdr | Optional Hdr | Section Hdrs | Exports

"ntdll.dll"

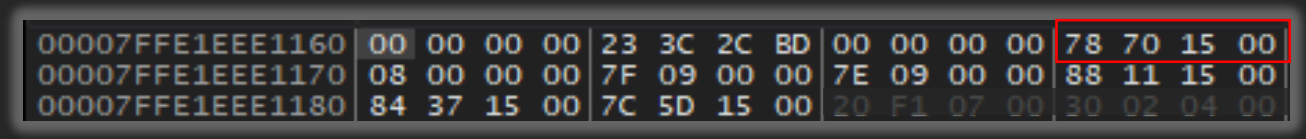
| Offset | Name | Value | Meaning |
|---|---|---|---|
| 14ED60 | Characteristics | 0 | |
| 14ED64 | TimeDateStamp | BD2C3C23 | 星期一, 28.07.2070 17:06:43 UTC |
| 14ED68 | MajorVersion | 0 | |
| 14ED6A | MinorVersion | 0 | |
| 14ED6C | Name | 157078 | ntdll.dll |
| 14ED70 | Base | 8 | |
| 14ED74 | NumberOfFunctions | 97F | |
| 14ED78 | NumberOfNames | 97E | |
| 14ED7C | AddressOfFunctions | 151188 | |
| 14ED80 | AddressOfNames | 153784 | |
| 14ED84 | AddressOfNameOrdinals | 155D7C | |

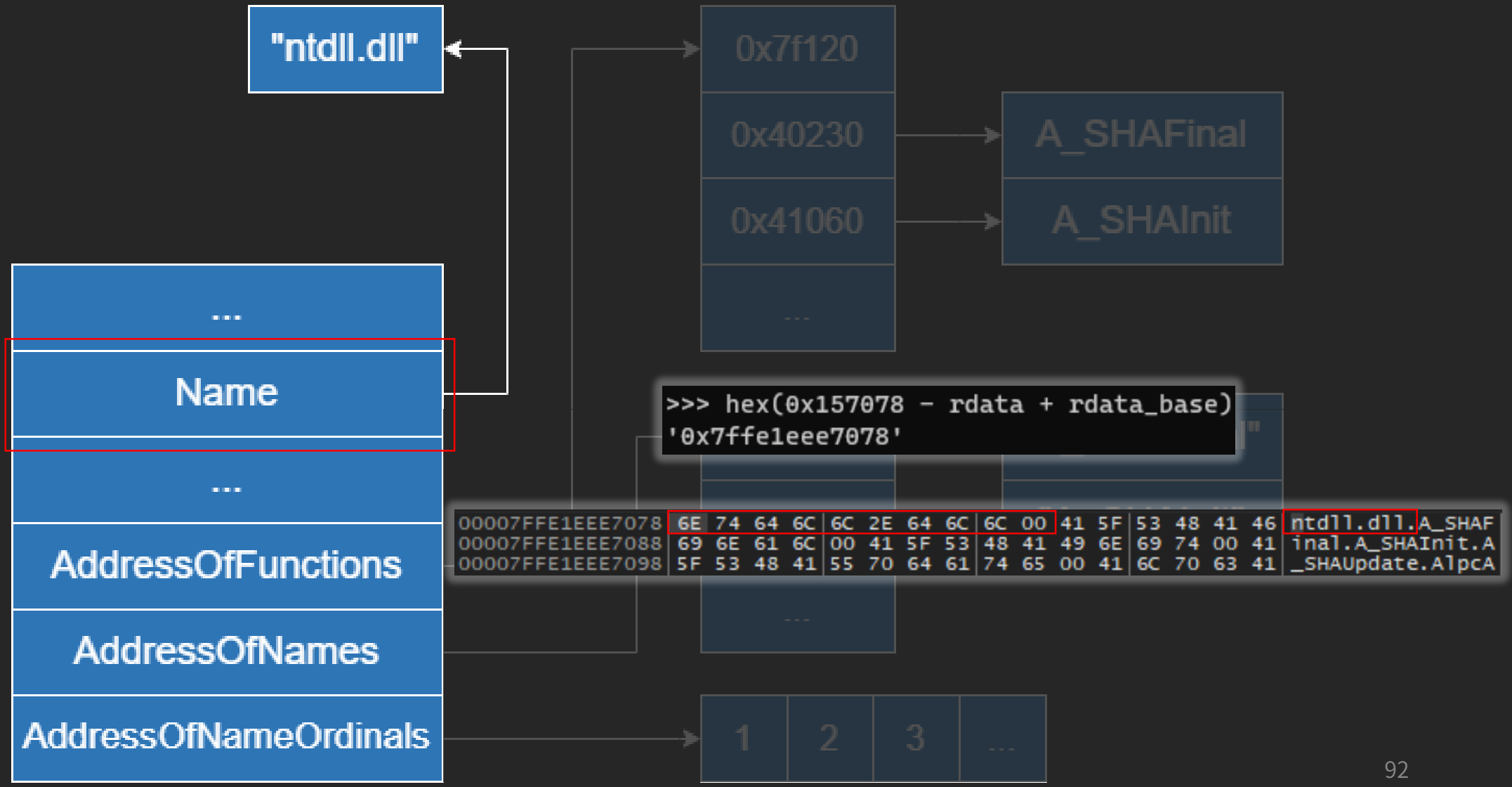
- ...
- Name
- ...
- AddressOfFunctions
- AddressOfNames
- AddressOfNameOrdinals

0x157082 → "A_SHAFinal"

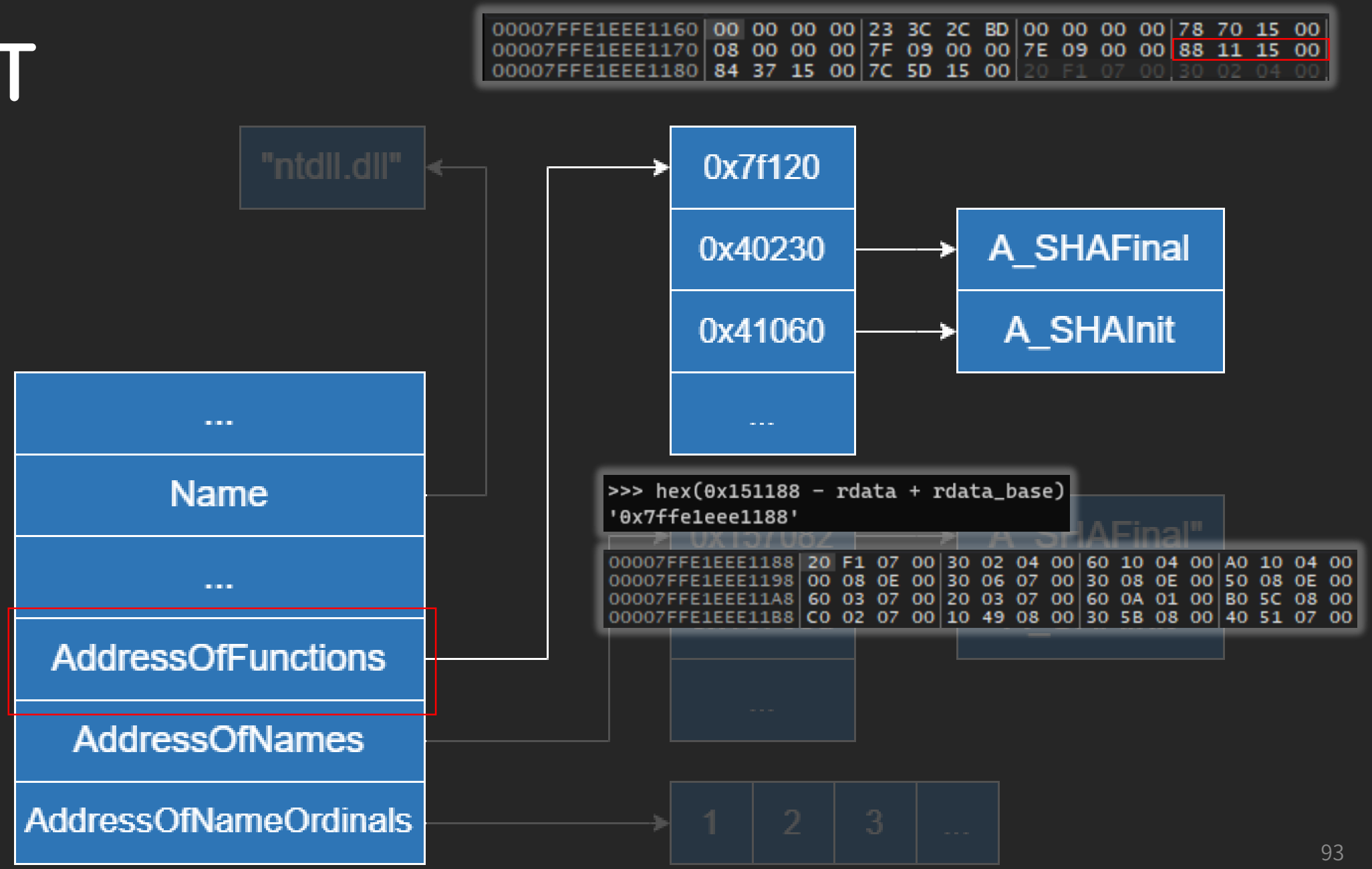
```
>>> rdata = 0x11C000
>>> rdata_base = 0x0007FFE1EEAC000
>>> export_dir = 0x151160
>>> hex(export_dir - rdata + rdata_base)
'0x7ffe1eee1160'
```

# EAT

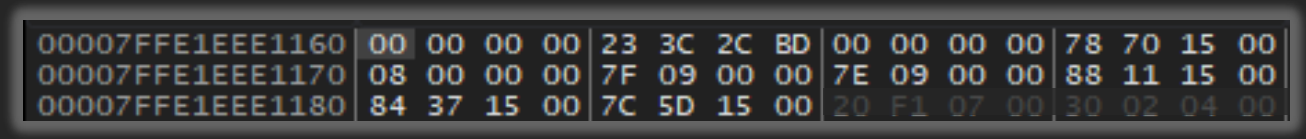

00007FFE1EEE1160 00 00 00 00 23 3C 2C BD 00 00 00 00 78 70 15 00
00007FFE1EEE1170 08 00 00 00 7F 09 00 00 7E 09 00 00 88 11 15 00
00007FFE1EEE1180 84 37 15 00 7C 5D 15 00 20 F1 07 00 30 02 04 00


"ntdll.dll"
0x7f120
0x40230
0x41060
...
A_SHAFinal
A_SHAInit
...
Name
...
AddressOfFunctions
AddressOfNames
AddressOfNameOrdinals
>>> hex(0x157078 - rdata + rdata_base)
'0x7ffe1eee7078'
00007FFE1EEE7078 6E 74 64 6C 6C 2E 64 6C 6C 00 41 5F 53 48 41 46 ntdll.dll.A_SHAF
00007FFE1EEE7088 69 6E 61 6C 00 41 5F 53 48 41 49 6E 69 74 00 41 inal.A_SHAInit.A
00007FFE1EEE7098 5F 53 48 41 55 70 64 61 74 65 00 41 6C 70 63 41 _SHAUpdate.AlpcA
...
1
2
3
...

# EAT


00007FFE1EEE1160 00 00 00 00 23 3C 2C BD 00 00 00 00 78 70 15 00
00007FFE1EEE1170 08 00 00 00 7F 09 00 00 7E 09 00 00 88 11 15 00
00007FFE1EEE1180 84 37 15 00 7C 5D 15 00 20 F1 07 00 30 02 04 00
"ntdll.dll"
0x7f120
0x40230
0x41060
...
A_SHAFinal
A_SHAInit
...
Name
...
AddressOfFunctions
AddressOfNames
AddressOfNameOrdinals
>>> hex(0x151188 - rdata + rdata_base)
'0x7ffe1eee1188'
00007FFE1EEE1188 20 F1 07 00 30 02 04 00 60 10 04 00 A0 10 04 00
00007FFE1EEE1198 00 08 0E 00 30 06 07 00 30 08 0E 00 50 08 0E 00
00007FFE1EEE11A8 60 03 07 00 20 03 07 00 60 0A 01 00 B0 5C 08 00
00007FFE1EEE11B8 C0 02 07 00 10 49 08 00 30 5B 08 00 40 51 07 00
1
2
3
...

# EAT




"ntdll.dll"

0x7f120

0x40230

0x41060

...

A_SHAFinal

A_SHAInit

...

Name

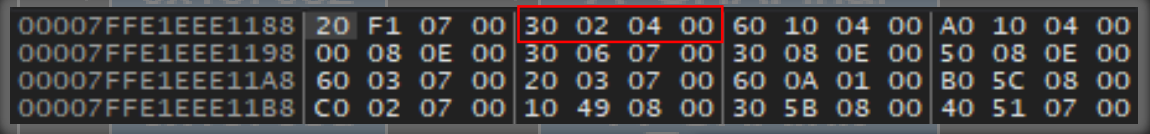
...

AddressOfFunctions

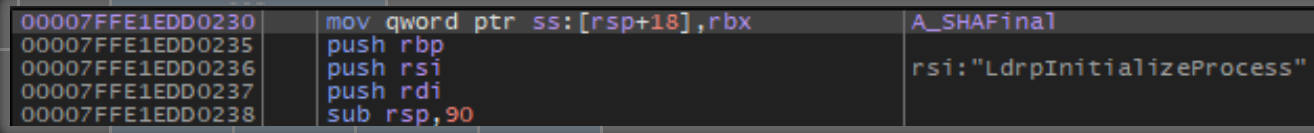
AddressOfNames

AddressOfNameOrdinals

```
>>> hex(0x151188 - rdata + rdata_base)
'0x7ffe1eee1188'
```

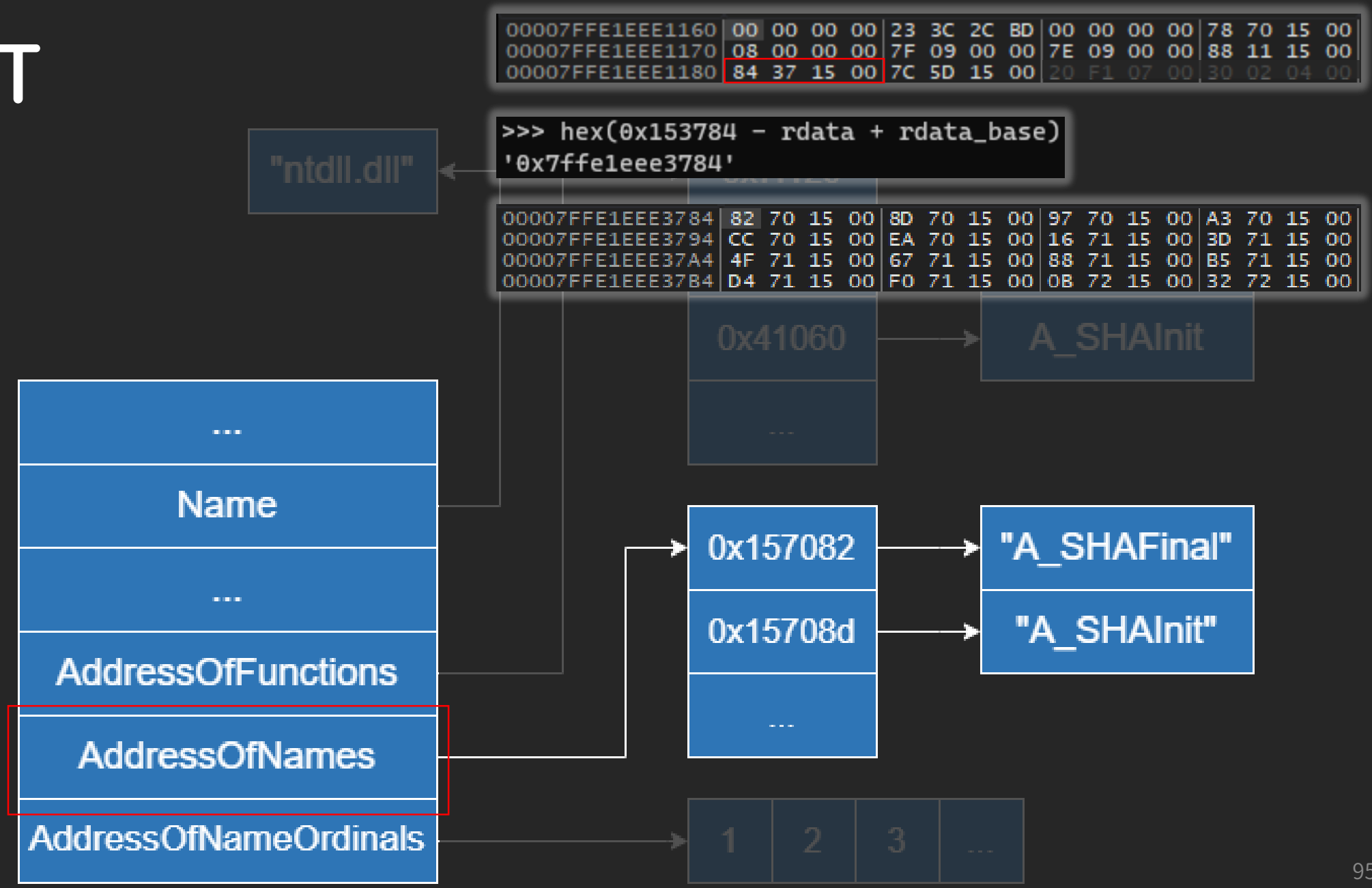



```
>>> hex(0x40230 - rdata + rdata_base)
'0x7ffe1edd0230'
```

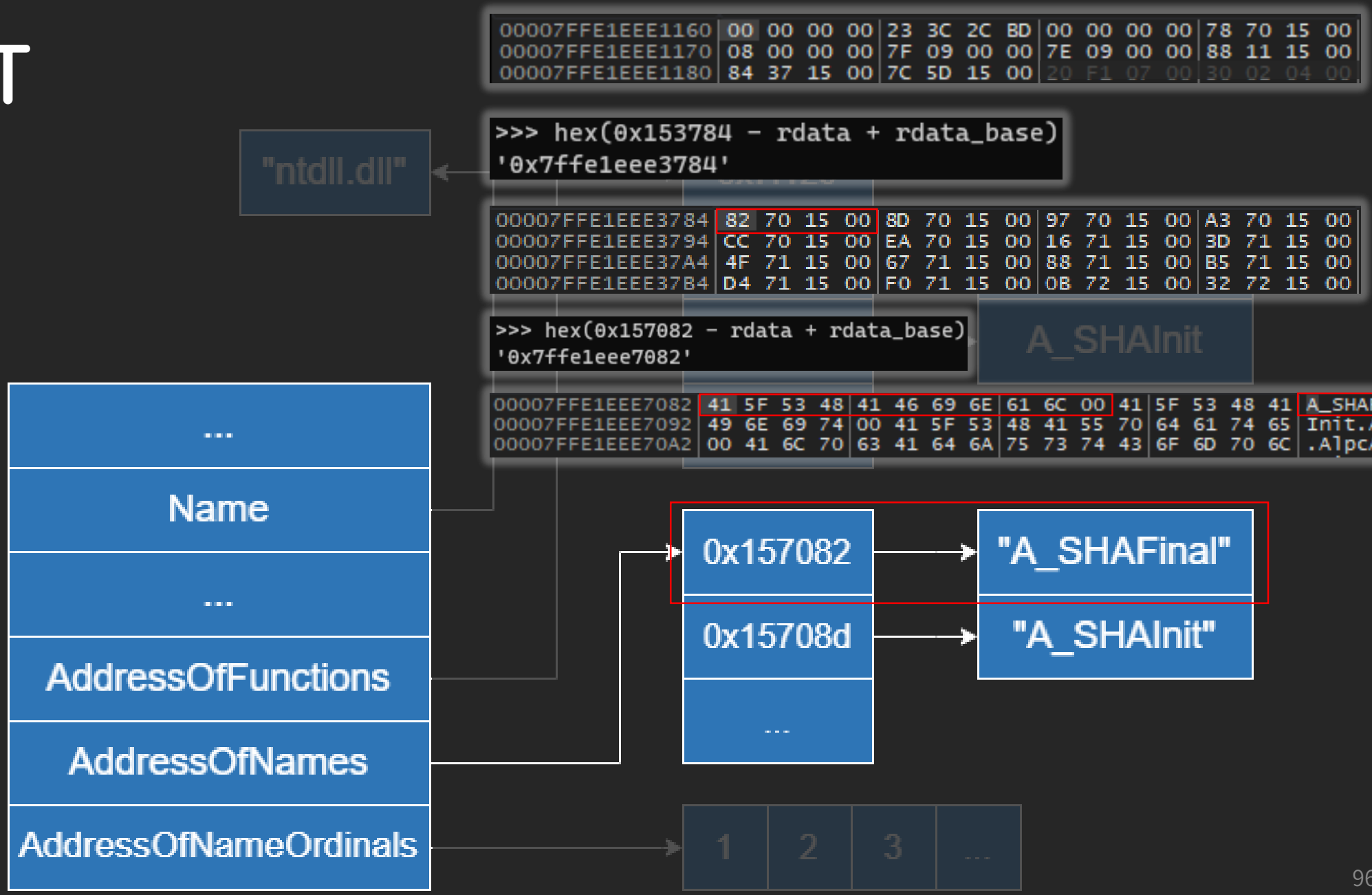



1 2 3 ...

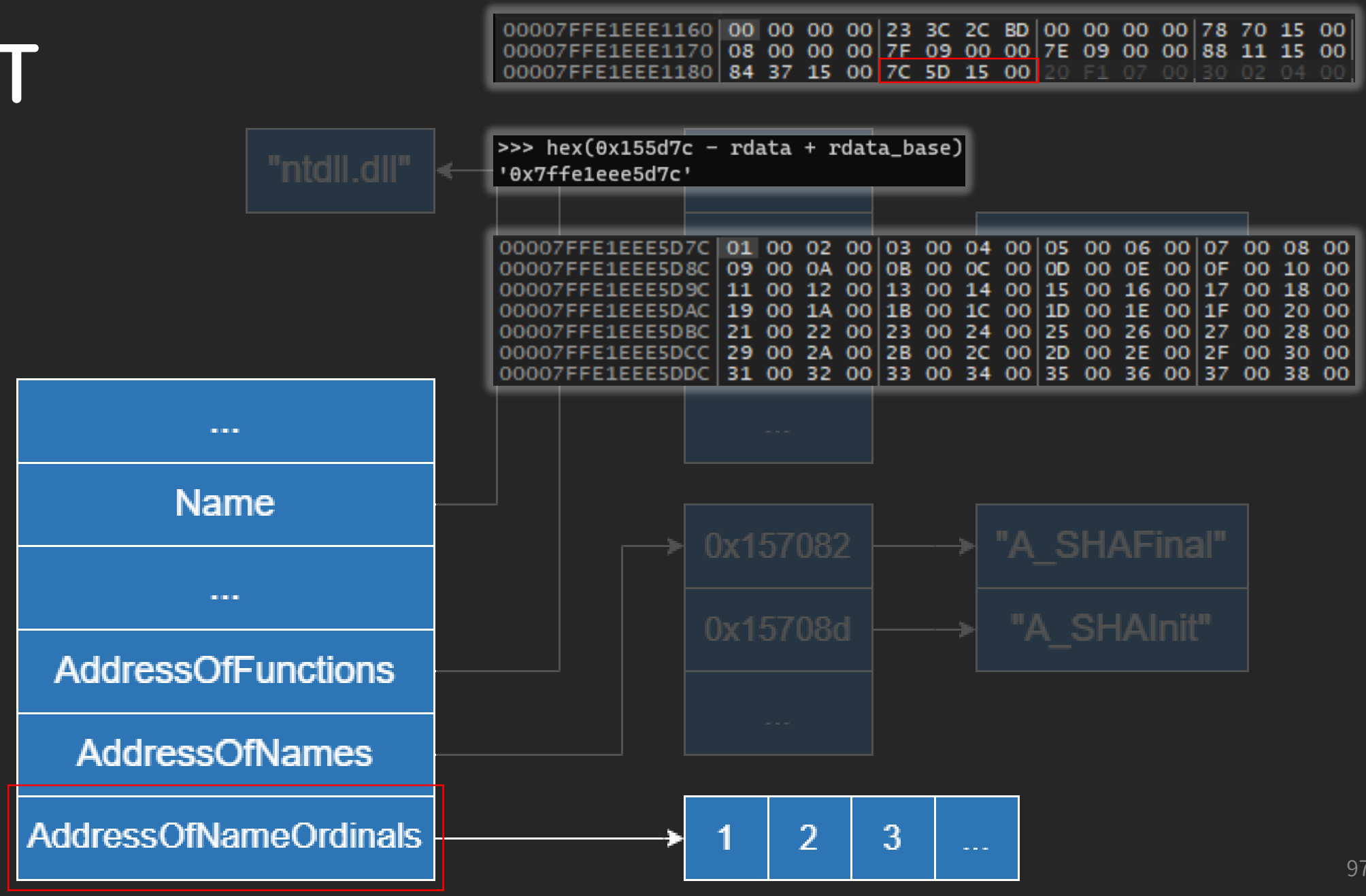
# EAT


00007FFE1EEE1160 00 00 00 00 23 3C 2C BD 00 00 00 00 78 70 15 00
00007FFE1EEE1170 08 00 00 00 7F 09 00 00 7E 09 00 00 88 11 15 00
00007FFE1EEE1180 84 37 15 00 7C 5D 15 00
>>> hex(0x153784 - rdata + rdata_base)
'0x7ffe1eee3784'
"ntdll.dll"
00007FFE1EEE3784 82 70 15 00 8D 70 15 00 97 70 15 00 A3 70 15 00
00007FFE1EEE3794 CC 70 15 00 EA 70 15 00 16 71 15 00 3D 71 15 00
00007FFE1EEE37A4 4F 71 15 00 67 71 15 00 88 71 15 00 B5 71 15 00
00007FFE1EEE37B4 D4 71 15 00 F0 71 15 00 0B 72 15 00 32 72 15 00
0x41060
A_SHAInit
...
Name
...
AddressOfFunctions
AddressOfNames
AddressOfNameOrdinals
0x157082
0x15708d
...
"A_SHAFinal"
"A_SHAInit"
1
2
3
...

# EAT


00007FFE1EEE1160 00 00 00 00 23 3C 2C BD 00 00 00 00 78 70 15 00
00007FFE1EEE1170 08 00 00 00 7F 09 00 00 7E 09 00 00 88 11 15 00
00007FFE1EEE1180 84 37 15 00 7C 5D 15 00
"ntdll.dll"
>>> hex(0x153784 - rdata + rdata_base)
'0x7ffe1eee3784'
00007FFE1EEE3784 82 70 15 00 8D 70 15 00 97 70 15 00 A3 70 15 00
00007FFE1EEE3794 CC 70 15 00 EA 70 15 00 16 71 15 00 3D 71 15 00
00007FFE1EEE37A4 4F 71 15 00 67 71 15 00 88 71 15 00 B5 71 15 00
00007FFE1EEE37B4 D4 71 15 00 F0 71 15 00 0B 72 15 00 32 72 15 00
>>> hex(0x157082 - rdata + rdata_base)
'0x7ffe1eee7082'
A_SHAInit
00007FFE1EEE7082 41 5F 53 48 41 46 69 6E 61 6C 00 41 5F 53 48 41 A_SHAFinal.A_SHA
00007FFE1EEE7092 49 6E 69 74 00 41 5F 53 48 41 55 70 64 61 74 65 Init.A_SHAUpdate
00007FFE1EEE70A2 00 41 6C 70 63 41 64 6A 75 73 74 43 6F 6D 70 6C .AlpcAdjustCompl
...
Name
...
AddressOfFunctions
AddressOfNames
AddressOfNameOrdinals
0x157082
0x15708d
...
"A_SHAFinal"
"A_SHAInit"
1
2
3
...

# EAT


00007FFE1EEE1160 00 00 00 00 23 3C 2C BD 00 00 00 00 78 70 15 00
00007FFE1EEE1170 08 00 00 00 7F 09 00 00 7E 09 00 00 88 11 15 00
00007FFE1EEE1180 84 37 15 00 7C 5D 15 00 20 F1 07 00 30 02 04 00
"ntdll.dll"
>>> hex(0x155d7c - rdata + rdata_base)
'0x7ffe1eee5d7c'
00007FFE1EEE5D7C 01 00 02 00 03 00 04 00 05 00 06 00 07 00 08 00
00007FFE1EEE5D8C 09 00 0A 00 0B 00 0C 00 0D 00 0E 00 0F 00 10 00
00007FFE1EEE5D9C 11 00 12 00 13 00 14 00 15 00 16 00 17 00 18 00
00007FFE1EEE5DAC 19 00 1A 00 1B 00 1C 00 1D 00 1E 00 1F 00 20 00
00007FFE1EEE5DBC 21 00 22 00 23 00 24 00 25 00 26 00 27 00 28 00
00007FFE1EEE5DCC 29 00 2A 00 2B 00 2C 00 2D 00 2E 00 2F 00 30 00
00007FFE1EEE5DDC 31 00 32 00 33 00 34 00 35 00 36 00 37 00 38 00
...
...
Name
...
AddressOfFunctions
AddressOfNames
AddressOfNameOrdinals
0x157082
0x15708d
...
"A_SHAFinal"
"A_SHAInit"
1
2
3
...

# PEB

Process Environment Block

# PEB

- MSDN 裡面是這樣寫的 …


```
typedef struct _PEB {
  BYTE                          Reserved1[2];
  BYTE                          BeingDebugged;
  BYTE                          Reserved2[1];
  PVOID                         Reserved3[2];
  PPEB_LDR_DATA                 Ldr;
  PRTL_USER_PROCESS_PARAMETERS  ProcessParameters;
  PVOID                         Reserved4[3];
  PVOID                         AtlThunkSListPtr;
  PVOID                         Reserved5;
  ULONG                         Reserved6;
  PVOID                         Reserved7;
  ULONG                         Reserved8;
  ULONG                         AtlThunkSListPtr32;
  PVOID                         Reserved9[45];
  BYTE                          Reserved10[96];
  PPS_POST_PROCESS_INIT_ROUTINE PostProcessInitRoutine;
  BYTE                          Reserved11[128];
  PVOID                         Reserved12[1];
  ULONG                         SessionId;
} PEB, *PPEB;
```

# PEB

- 改用 windbg 來查看…

```
||1:1:001> dt _peb @$peb
ntdll!_PEB
   +0x000 InheritedAddressSpace : 0 ''
   +0x001 ReadImageFileExecOptions : 0 ''
   +0x002 BeingDebugged    : 0x1 ''
   +0x003 BitField         : 0 ''
   +0x003 ImageUsesLargePages : 0y0
   +0x003 IsProtectedProcess : 0y0
   +0x003 IsImageDynamicallyRelocated : 0y0
   +0x003 SkipPatchingUser32Forwarders : 0y0
   +0x003 IsPackagedProcess : 0y0
   +0x003 IsAppContainer   : 0y0
   +0x003 IsProtectedProcessLight : 0y0
   +0x003 IsLongPathAwareProcess : 0y0
   +0x004 Padding0         : [4]  ""
   +0x008 Mutant           : 0xffffffff`ffffffff Void
   +0x010 ImageBaseAddress : 0x00000000`00400000 Void
   +0x018 Ldr              : 0x00007fff`dd0c53c0 _PEB_LDR_DATA
   +0x020 ProcessParameters : 0x00000000`00772190 _RTL_USER_PROCESS_PARAMETERS
   +0x028 SubSystemData    : (null)
   +0x030 ProcessHeap      : 0x00000000`00770000 Void
   +0x038 FastPebLock      : 0x00007fff`dd0c4fe0 _RTL_CRITICAL_SECTION
   +0x040 AtlThunkSListPtr : (null)
   +0x048 IFEOKey          : (null)
   +0x050 CrossProcessFlags : 2
   +0x050 ProcessInJob     : 0y0
   +0x050 ProcessInitializing : 0y1
   +0x050 ProcessUsingVEH  : 0y0
   +0x050 ProcessUsingVCH  : 0y0
   +0x050 ProcessUsingFTH  : 0y0
   +0x050 ProcessPreviouslyThrottled : 0y0
   +0x050 ProcessCurrentlyThrottled : 0y0
   +0x050 ProcessImagesHotPatched : 0y0
   +0x050 ReservedBits0    : 0y00000000000000000000000000 (0)
   +0x054 Padding1         : [4]  ""
   +0x058 KernelCallbackTable : (null)
   +0x058 UserSharedInfoPtr : (null)
   +0x060 SystemReserved   : 0
   +0x064 AtlThunkSListPtr32 : 0
```


MSDN
騙你的

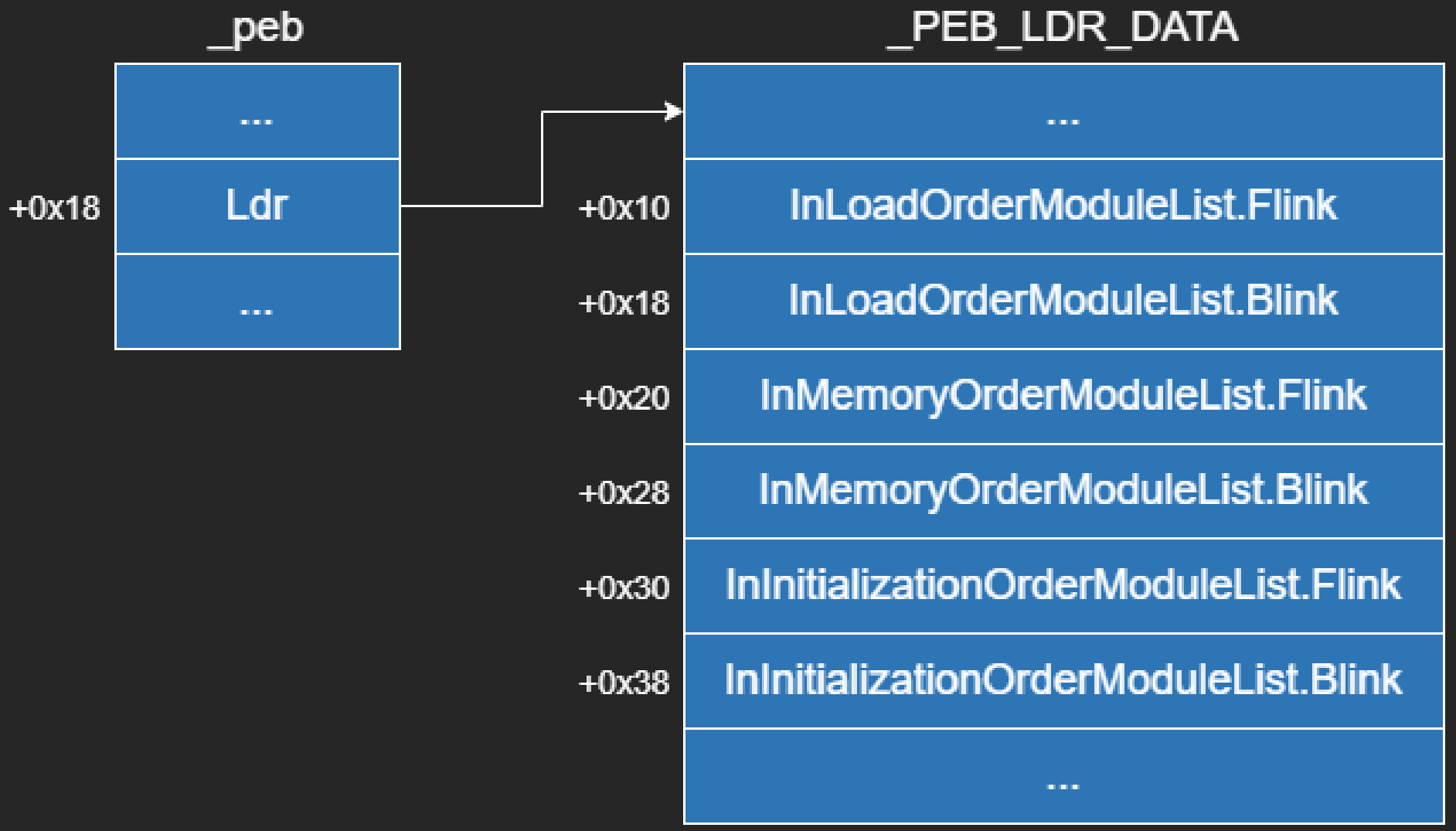
# PEB

- 取得 PEB
- 32bit
  - FS:[0x30]
- 64bit
  - GS:[0x60]

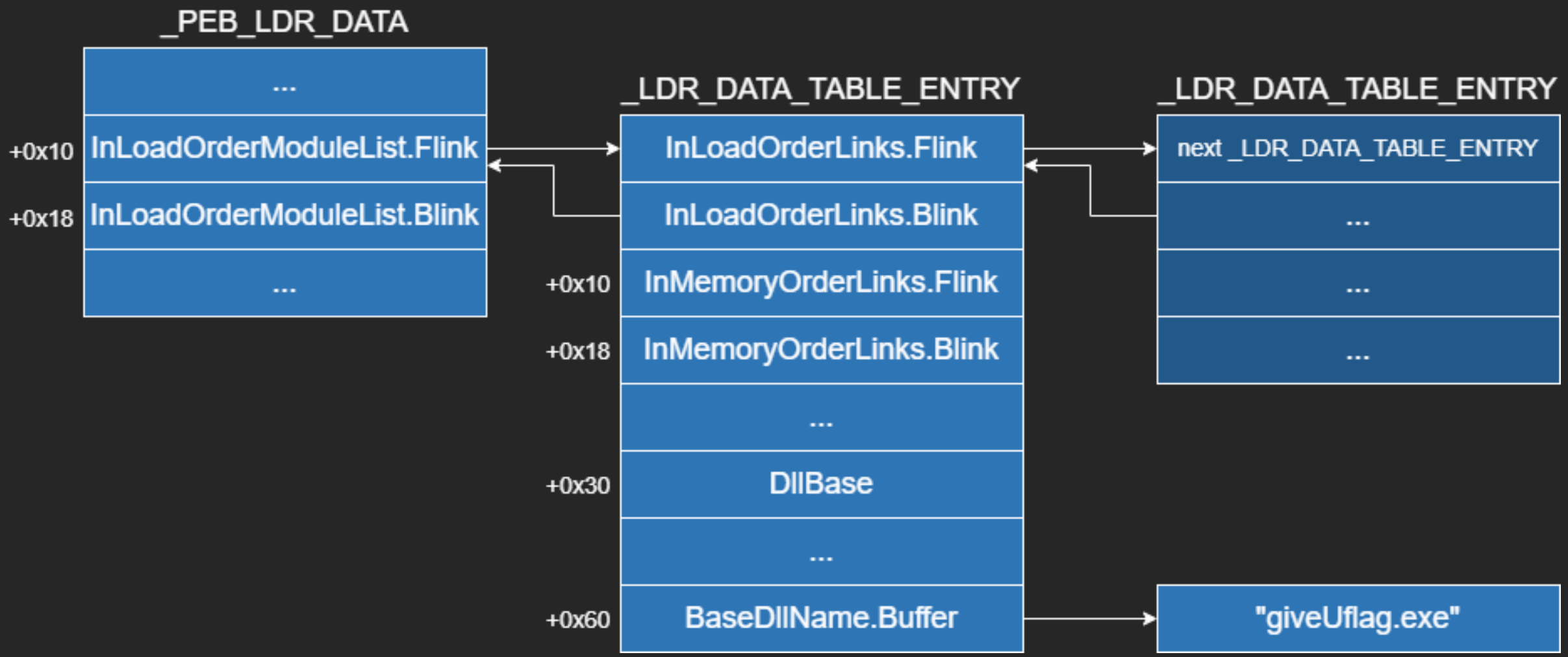
```
mov     rax, gs:60h     ; get PEB
mov     [rbp+PEB], rax
```

Ref: https://en.wikipedia.org/wiki/Win32_Thread_Information_Block

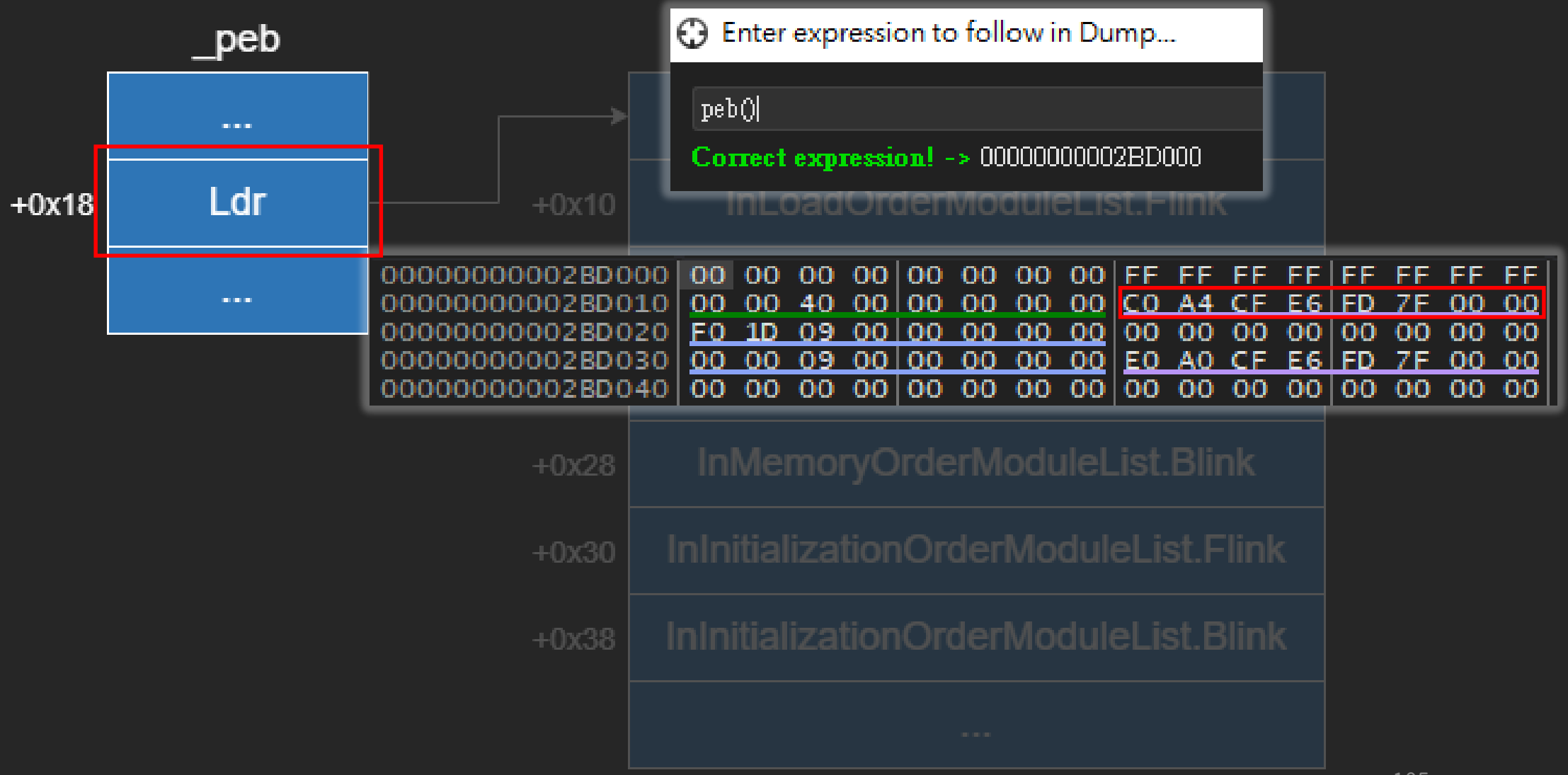
# PEB


_peb
...
+0x18
Ldr
...
_PEB_LDR_DATA
...
+0x10
InLoadOrderModuleList.Flink
+0x18
InLoadOrderModuleList.Blink
+0x20
InMemoryOrderModuleList.Flink
+0x28
InMemoryOrderModuleList.Blink
+0x30
InInitializationOrderModuleList.Flink
+0x38
InInitializationOrderModuleList.Blink
...

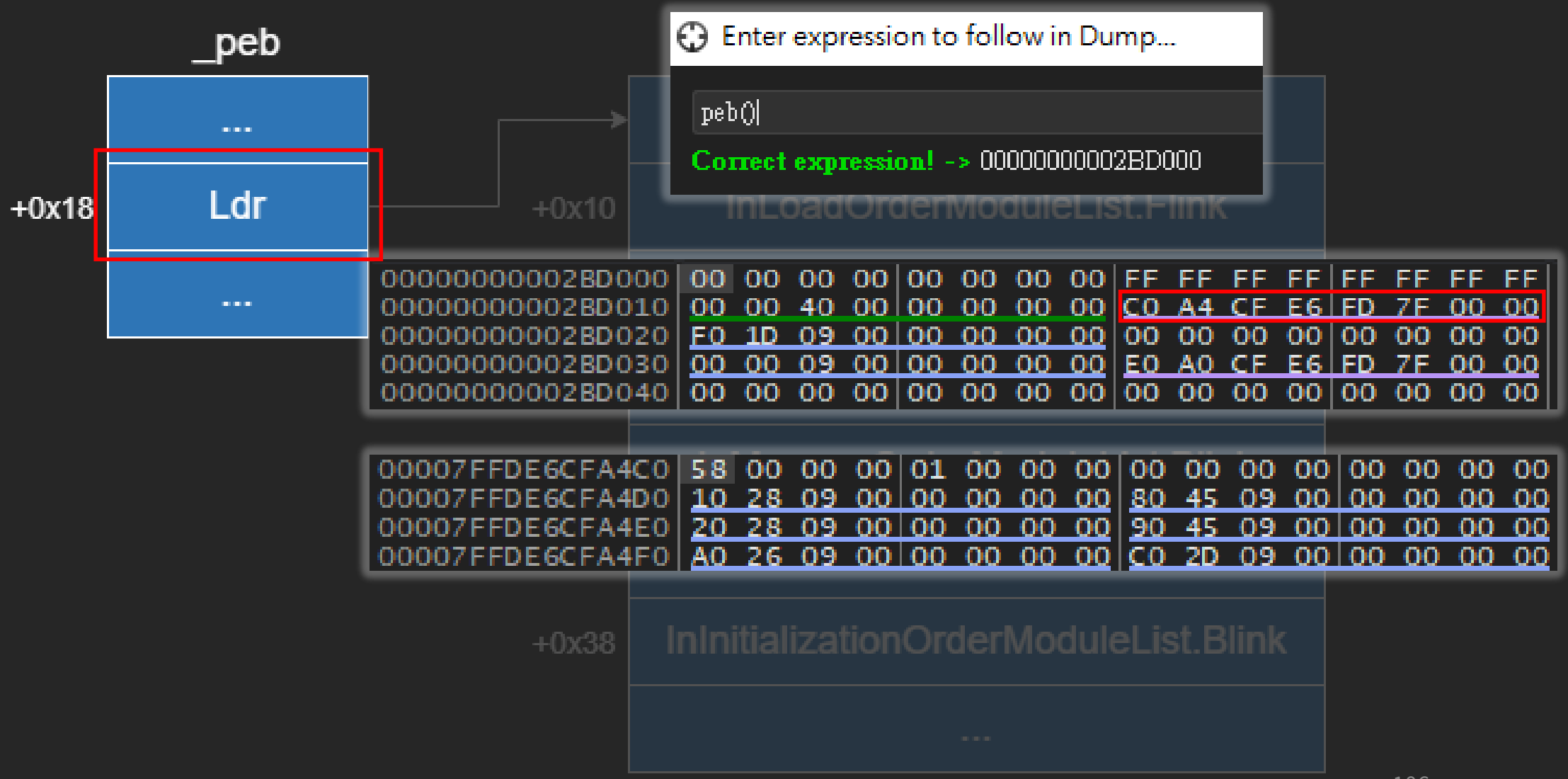
# PEB


_PEB_LDR_DATA
...
+0x10
InLoadOrderModuleList.Flink
+0x18
InLoadOrderModuleList.Blink
...
_LDR_DATA_TABLE_ENTRY
InLoadOrderLinks.Flink
InLoadOrderLinks.Blink
+0x10
InMemoryOrderLinks.Flink
+0x18
InMemoryOrderLinks.Blink
...
+0x30
DllBase
...
+0x60
BaseDllName.Buffer
_LDR_DATA_TABLE_ENTRY
next _LDR_DATA_TABLE_ENTRY
...
...
...
"giveUflag.exe"

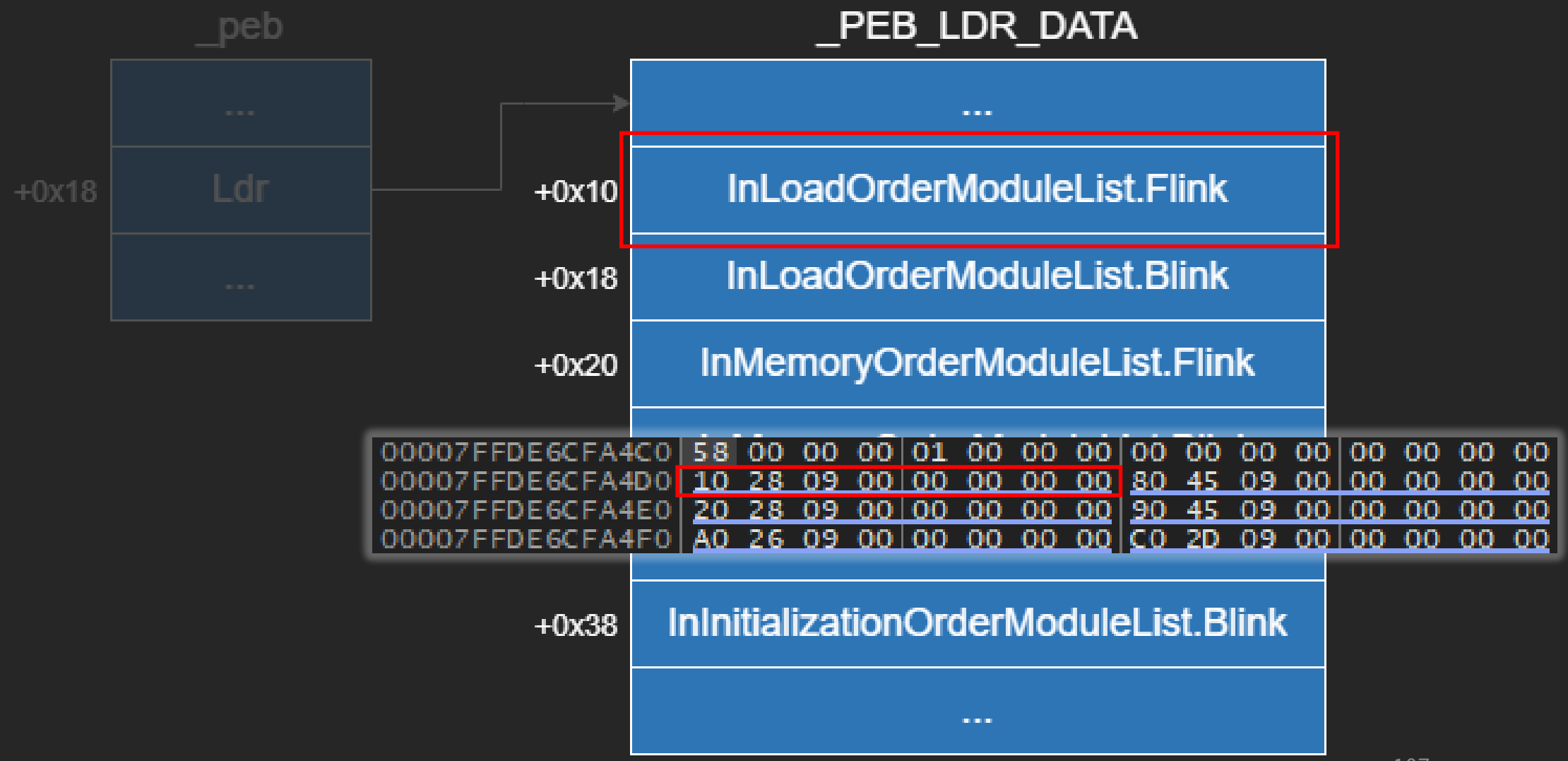
# PEB


_peb
...
+0x18
Ldr
...
Enter expression to follow in Dump...
peb()
Correct expression! -> 00000000002BD000
+0x10
InLoadOrderModuleList.Flink
00000000002BD000 00 00 00 00 00 00 00 00 FF FF FF FF FF FF FF FF
00000000002BD010 00 00 40 00 00 00 00 00 C0 A4 CF E6 FD 7F 00 00
00000000002BD020 F0 1D 09 00 00 00 00 00 00 00 00 00 00 00 00 00
00000000002BD030 00 00 09 00 00 00 00 00 E0 A0 CF E6 FD 7F 00 00
00000000002BD040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
+0x28
InMemoryOrderModuleList.Blink
+0x30
InInitializationOrderModuleList.Flink
+0x38
InInitializationOrderModuleList.Blink
...

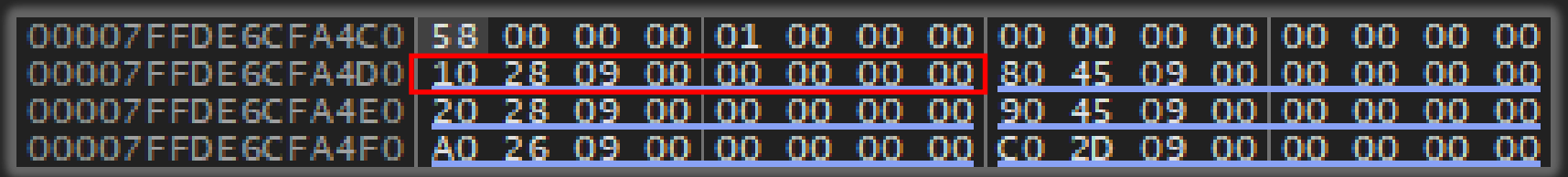
# PEB

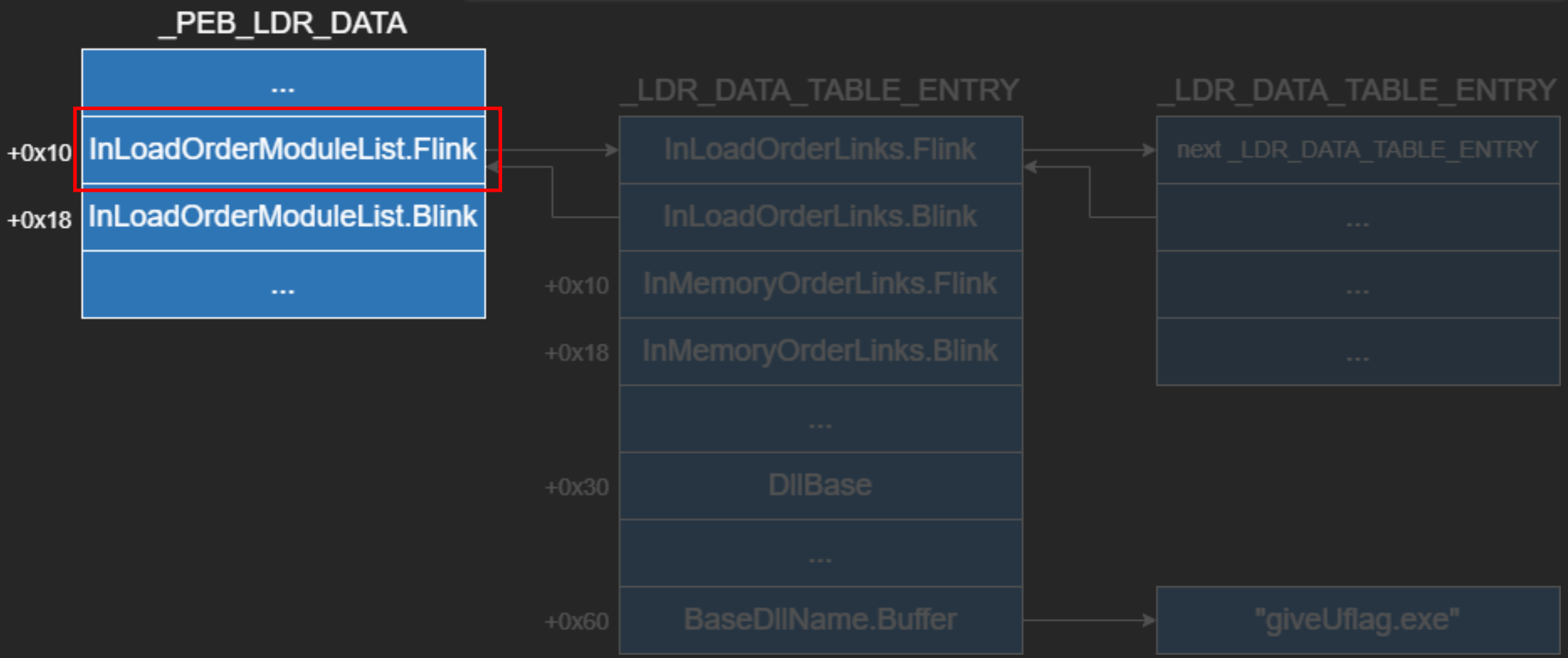

_peb
...
+0x18
Ldr
...
Enter expression to follow in Dump...
peb()
Correct expression! -> 00000000002BD000
+0x10
InLoadOrderModuleList.Flink
00000000002BD000 00 00 00 00 00 00 00 00 FF FF FF FF FF FF FF FF
00000000002BD010 00 00 40 00 00 00 00 00 C0 A4 CF E6 FD 7F 00 00
00000000002BD020 F0 1D 09 00 00 00 00 00 00 00 00 00 00 00 00 00
00000000002BD030 00 00 09 00 00 00 00 00 E0 A0 CF E6 FD 7F 00 00
00000000002BD040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00007FFDE6CFA4C0 58 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00
00007FFDE6CFA4D0 10 28 09 00 00 00 00 00 80 45 09 00 00 00 00 00
00007FFDE6CFA4E0 20 28 09 00 00 00 00 00 90 45 09 00 00 00 00 00
00007FFDE6CFA4F0 A0 26 09 00 00 00 00 00 C0 2D 09 00 00 00 00 00
+0x38
InInitializationOrderModuleList.Blink
...

# PEB

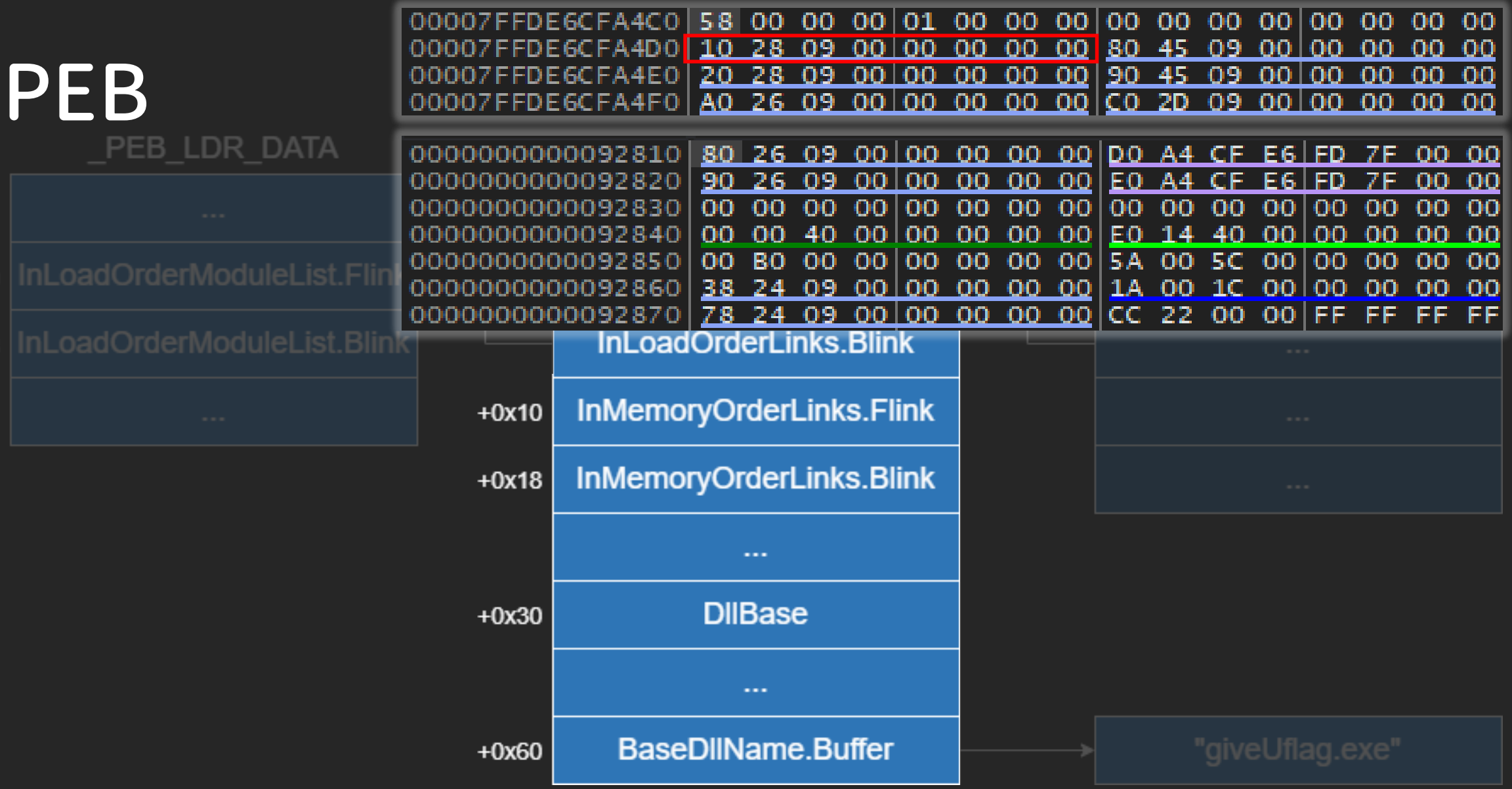

_peb
...
+0x18
Ldr
...
_PEB_LDR_DATA
...
+0x10
InLoadOrderModuleList.Flink
+0x18
InLoadOrderModuleList.Blink
+0x20
InMemoryOrderModuleList.Flink
00007FFDE6CFA4C0 58 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00
00007FFDE6CFA4D0 10 28 09 00 00 00 00 00 80 45 09 00 00 00 00 00
00007FFDE6CFA4E0 20 28 09 00 00 00 00 00 90 45 09 00 00 00 00 00
00007FFDE6CFA4F0 A0 26 09 00 00 00 00 00 C0 2D 09 00 00 00 00 00
+0x38
InInitializationOrderModuleList.Blink
...

# PEB

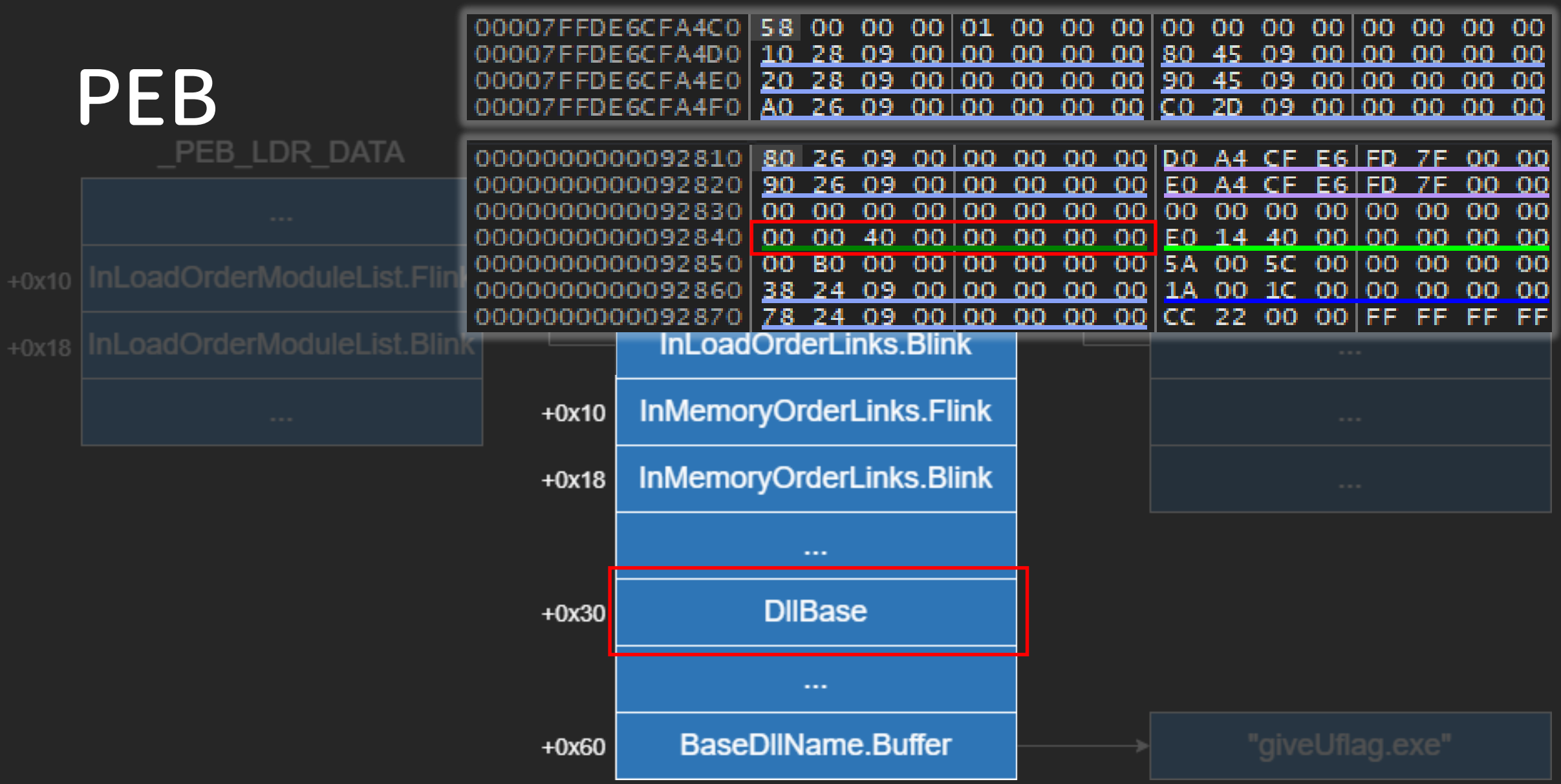

00007FFDE6CFA4C0 58 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00
00007FFDE6CFA4D0 10 28 09 00 00 00 00 00 80 45 09 00 00 00 00 00
00007FFDE6CFA4E0 20 28 09 00 00 00 00 00 90 45 09 00 00 00 00 00
00007FFDE6CFA4F0 A0 26 09 00 00 00 00 00 C0 2D 09 00 00 00 00 00


_PEB_LDR_DATA
...
+0x10
InLoadOrderModuleList.Flink
+0x18
InLoadOrderModuleList.Blink
...
_LDR_DATA_TABLE_ENTRY
InLoadOrderLinks.Flink
InLoadOrderLinks.Blink
+0x10
InMemoryOrderLinks.Flink
+0x18
InMemoryOrderLinks.Blink
...
+0x30
DllBase
...
+0x60
BaseDllName.Buffer
_LDR_DATA_TABLE_ENTRY
next _LDR_DATA_TABLE_ENTRY
...
...
...
"giveUflag.exe"

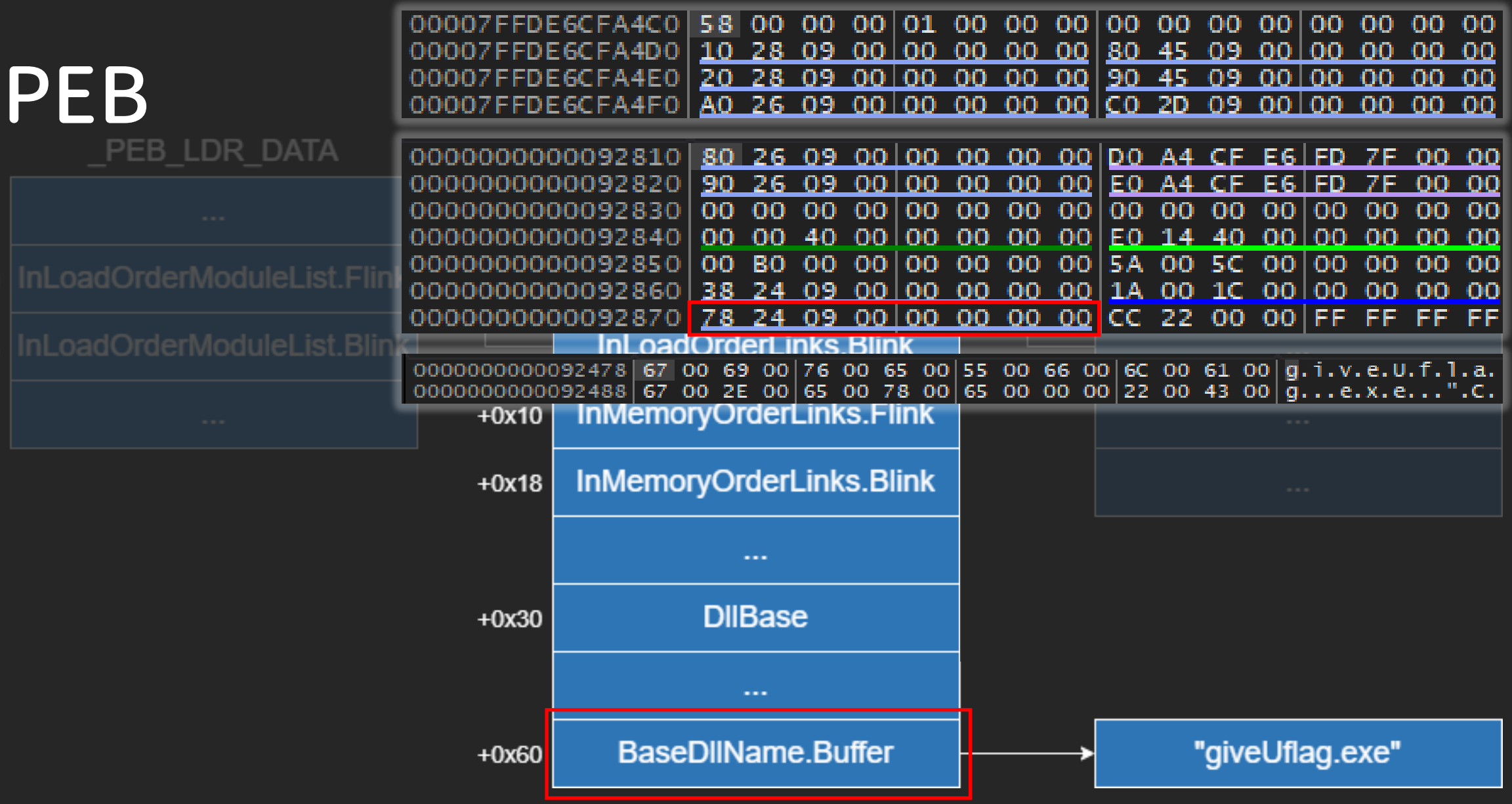
# PEB


00007FFDE6CFA4C0 58 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00
00007FFDE6CFA4D0 10 28 09 00 00 00 00 00 80 45 09 00 00 00 00 00
00007FFDE6CFA4E0 20 28 09 00 00 00 00 00 90 45 09 00 00 00 00 00
00007FFDE6CFA4F0 A0 26 09 00 00 00 00 00 C0 2D 09 00 00 00 00 00
0000000000092810 80 26 09 00 00 00 00 00 D0 A4 CF E6 FD 7F 00 00
0000000000092820 90 26 09 00 00 00 00 00 E0 A4 CF E6 FD 7F 00 00
0000000000092830 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000000000092840 00 00 40 00 00 00 00 00 E0 14 40 00 00 00 00 00
0000000000092850 00 B0 00 00 00 00 00 00 5A 00 5C 00 00 00 00 00
0000000000092860 38 24 09 00 00 00 00 00 1A 00 1C 00 00 00 00 00
0000000000092870 78 24 09 00 00 00 00 00 CC 22 00 00 FF FF FF FF
_PEB_LDR_DATA
...
+0x10
InLoadOrderModuleList.Flink
+0x18
InLoadOrderModuleList.Blink
...
InLoadOrderLinks.Blink
+0x10
InMemoryOrderLinks.Flink
+0x18
InMemoryOrderLinks.Blink
...
+0x30
DllBase
...
+0x60
BaseDllName.Buffer
...
...
...
"giveUflag.exe"

# PEB


00007FFDE6CFA4C0 58 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00
00007FFDE6CFA4D0 10 28 09 00 00 00 00 00 80 45 09 00 00 00 00 00
00007FFDE6CFA4E0 20 28 09 00 00 00 00 00 90 45 09 00 00 00 00 00
00007FFDE6CFA4F0 A0 26 09 00 00 00 00 00 C0 2D 09 00 00 00 00 00
00000000092810 80 26 09 00 00 00 00 00 D0 A4 CF E6 FD 7F 00 00
00000000092820 90 26 09 00 00 00 00 00 E0 A4 CF E6 FD 7F 00 00
00000000092830 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000000092840 00 00 40 00 00 00 00 00 E0 14 40 00 00 00 00 00
00000000092850 00 B0 00 00 00 00 00 00 5A 00 5C 00 00 00 00 00
00000000092860 38 24 09 00 00 00 00 00 1A 00 1C 00 00 00 00 00
00000000092870 78 24 09 00 00 00 00 00 CC 22 00 00 FF FF FF FF
_PEB_LDR_DATA
...
+0x10
InLoadOrderModuleList.Flink
+0x18
InLoadOrderModuleList.Blink
...
InLoadOrderLinks.Blink
+0x10
InMemoryOrderLinks.Flink
+0x18
InMemoryOrderLinks.Blink
...
+0x30
DllBase
...
+0x60
BaseDllName.Buffer
...
...
...
"giveUflag.exe"

# PEB


00007FFDE6CFA4C0 58 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00
00007FFDE6CFA4D0 10 28 09 00 00 00 00 00 80 45 09 00 00 00 00 00
00007FFDE6CFA4E0 20 28 09 00 00 00 00 00 90 45 09 00 00 00 00 00
00007FFDE6CFA4F0 A0 26 09 00 00 00 00 00 C0 2D 09 00 00 00 00 00
00000000092810 80 26 09 00 00 00 00 00 D0 A4 CF E6 FD 7F 00 00
00000000092820 90 26 09 00 00 00 00 00 E0 A4 CF E6 FD 7F 00 00
00000000092830 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000000092840 00 00 40 00 00 00 00 00 E0 14 40 00 00 00 00 00
00000000092850 00 B0 00 00 00 00 00 00 5A 00 5C 00 00 00 00 00
00000000092860 38 24 09 00 00 00 00 00 1A 00 1C 00 00 00 00 00
00000000092870 78 24 09 00 00 00 00 00 CC 22 00 00 FF FF FF FF
00000000092478 67 00 69 00 76 00 65 00 55 00 66 00 6C 00 61 00 g.i.v.e.U.f.l.a.
00000000092488 67 00 2E 00 65 00 78 00 65 00 00 00 22 00 43 00 g...e.x.e..."C.
_PEB_LDR_DATA
+0x10
InLoadOrderModuleList.Flink
+0x18
InLoadOrderModuleList.Blink
InLoadOrderLinks.Blink
+0x10
InMemoryOrderLinks.Flink
+0x18
InMemoryOrderLinks.Blink
+0x30
DllBase
+0x60
BaseDllName.Buffer
"giveUflag.exe"

# PEB

- 一直順著 linked list 爬下去
- 檢查 BaseDllName 是否為想用的模組 (dll / exe)
- 通過 DllBase 取得該模組的 base address
- 解析在 base address 的 PE format, 從 EAT 爬到想用的 API
- Manual Symbol Resolution!

# Q & A

# 下課囉 \(. _ .)>