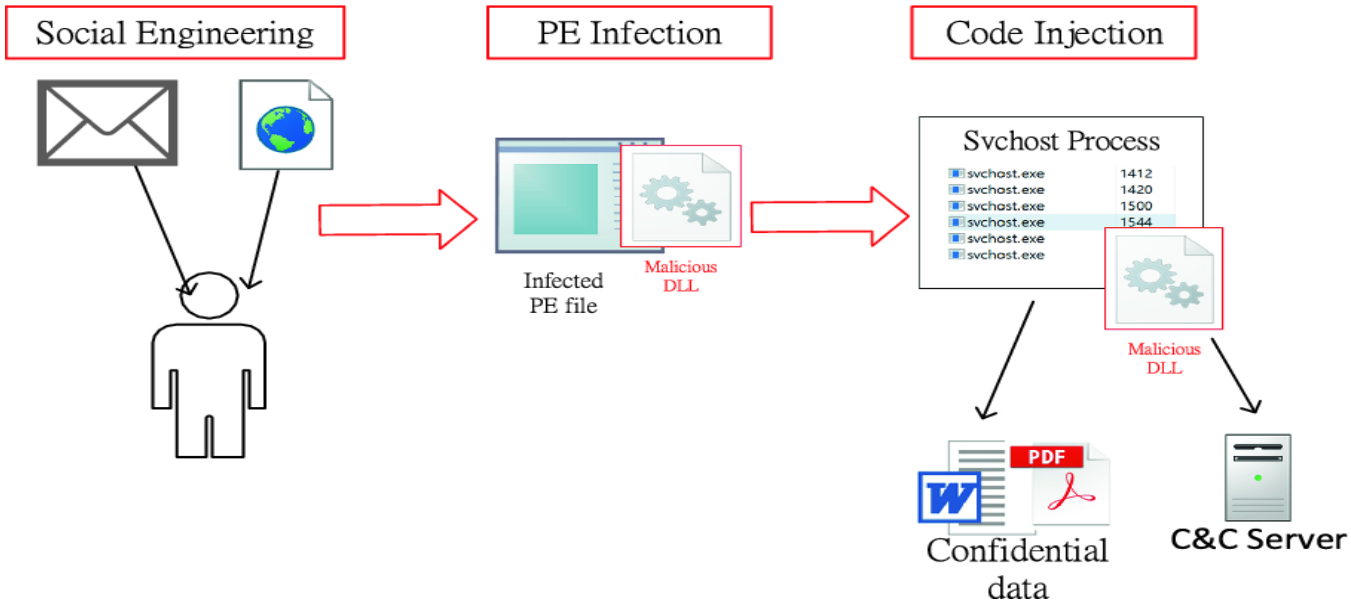
PE Infection

Portable Executable Infection

Author : Hejap Zairy

Instructor : Ali Hadi

بسم الله الرحمن الرحيم

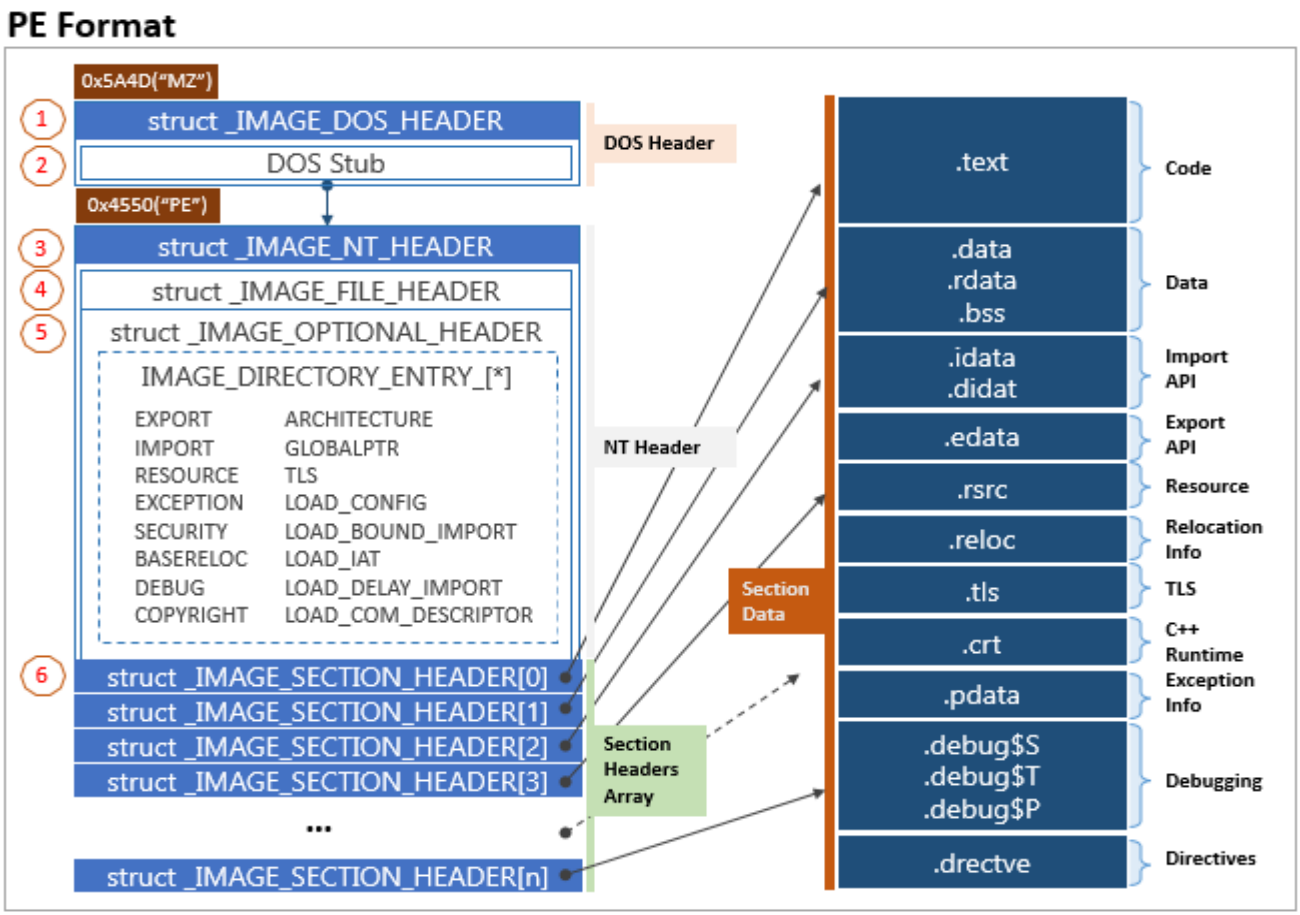


راح نتكلم اليوم عن حقن التطبيقات بأي كود خبيث بشكل يدوي وكيف نتجنبها ونحمي نفسنا منها

هذي الطريقة تعتبر من اقدم الطرق الفعالة الى الأن في نشر البرمجيات الخبيثة وذلك بحقن الأكواد الخبيثة في الملفات التنفيذية في أي نظام تشغيل كل نظام وله طريقة

ولكن راح نتكلم عن نظام **Windows** وراح نفهم **Portable Executabe** وراح نعطي أهم الحاجات اللي نفهمها عشان نعمل هذي الطريقة

ماهو Portable Executabe :

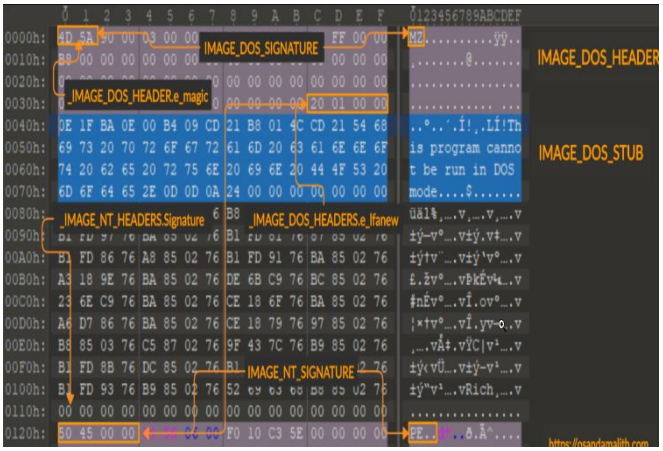


هو المهيأ الأساسي لعمل أي تطبيق تنفيذي في الويندوز بشكل تام ويتكون فيه الكثير من المعلومات الاساسية للتطبيق التنفيذي مثلا اصداره ومساحة وكم من الرام مخزن له وهذا مايسمى **VA** والهارد وهذا مايسمى بالتفصيل **RA** وراح نشرحها راح نتطرق الأن للأشياء الاساسية ونشرح **structure** الأساسي له

١. Dos Header

هو المكون الأساسي لكل ملف له امتداد signature الخاص فيه مثلا pdf او غيره مثل Dos stub وفيه الاوفسيت الخاص ب **PE Header**

```
typedef struct _IMAGE_DOS_HEADER {      // DOS .EXE header
    WORD   e_magic;                     // Magic number
    WORD   e_cblp;                      // Bytes on last page of file
    WORD   e_cp;                        // Pages in file
    WORD   e_crlc;                      // Relocations
    WORD   e_cparhdr;                   // Size of header in paragraphs
    WORD   e_minalloc;                  // Minimum extra paragraphs
needed
    WORD   e_maxalloc;                  // Maximum extra paragraphs
needed
    WORD   e_ss;                        // Initial (relative) SS value
    WORD   e_sp;                        // Initial SP value
    WORD   e_csum;                      // Checksum
    WORD   e_ip;                        // Initial IP value
    WORD   e_cs;                        // Initial (relative) CS value
    WORD   e_lfarlc;                    // File address of relocation
table
    WORD   e_ovno;                      // Overlay number
    WORD   e_res[4];                    // Reserved words
    WORD   e_oemid;                     // OEM identifier (for e_oeminfo)
    WORD   e_oeminfo;                   // OEM information; e_oemid
specific
    WORD   e_res2[10];                  // Reserved words
    LONG   e_lfanew;                    // File address of new exe header
  } IMAGE_DOS_HEADER, *PIMAGE_DOS_HEADER;
```




```
typedef struct _IMAGE_NT_HEADERS64 {
    DWORD Signature;
    IMAGE_FILE_HEADER FileHeader;
    IMAGE_OPTIONAL_HEADER64 OptionalHeader;
} IMAGE_NT_HEADERS64, *PIMAGE_NT_HEADERS64;
```

٢. NT_Header

المكون **NT** مكون مهم جدا في داخله عدة **structure** مهمة في داخلها

- File Header

في داخله معلومات الملف من ناحية **architecture** الخاص في تشغيل الملف التنفيذي هل 32bit or 64bit وهل هو dll او لا ووقته انشائه معلومات أكثر واساسية في تشغيل الملف التنفيذي

- **optional header**

في داخل هذا المكون اشياء أساسية مهمة مثل هل الملف فيه **nx(Dep)** ووقته انشائه وهل الملف بواجهة ولا بدون واهم معلومة هي يعطيك **AddressOfEntryPoint** المكان المراد تنفيذه في بداية تشغيل البرنامج وهذي راح نحتاجه وقت نتطرق pe Infection

AddressOfEntryPoint	00000128	Dword	00001784	.text

```
typedef struct _IMAGE_OPTIONAL_HEADER {
WORD                 Magic;
BYTE                 MajorLinkerVersion;
BYTE                 MinorLinkerVersion;
DWORD                SizeOfCode;
DWORD                SizeOfInitializedData;
DWORD                SizeOfUninitializedData;
DWORD                AddressOfEntryPoint;
} IMAGE_OPTIONAL_HEADER32,
*PIMAGE_OPTIONAL_HEADER32;
```

٣. Section Header

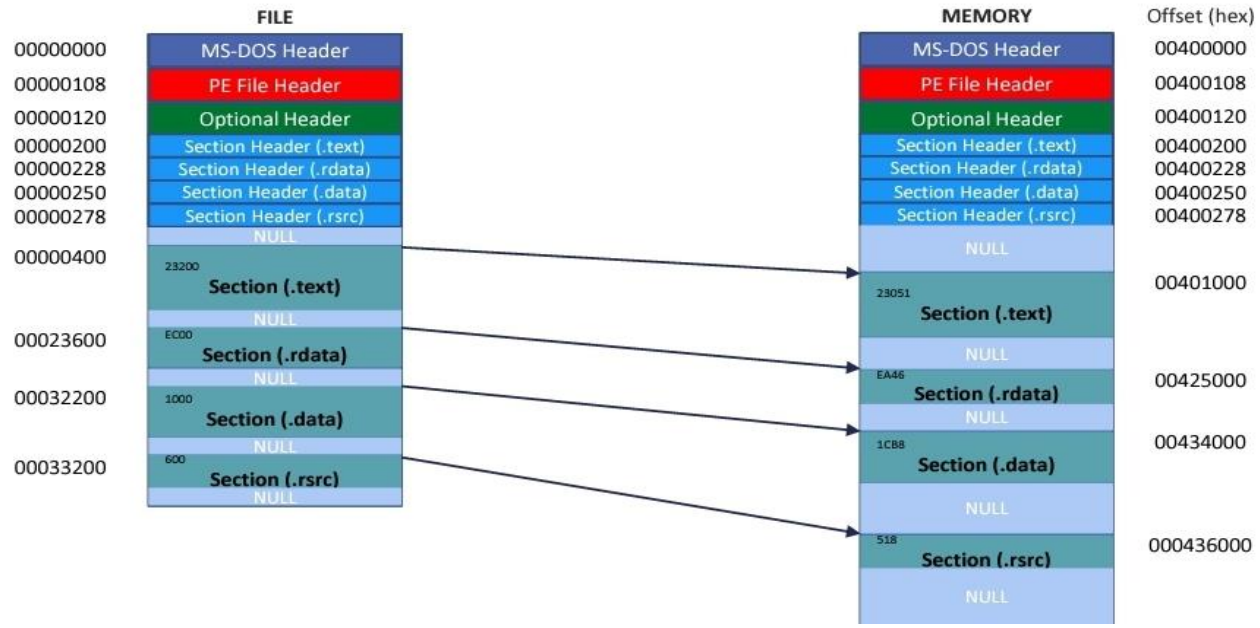
المكون هذا موجود في الاكواد المعرفة والاشياء الاساسية لتشغيل البرنامج بشكل كامل مثال الكود الاساسي والقيم المعرفة وايضا الايقونة وموضحة في الصورة فوق بشكل كبير وايضا المكان اللي يتوجهله **AddressOfEntryPoint** كما هو موضح فوق **وايضا نستطيع الإضافة فيه**

Name	Virtual Size	Virtual Address	Raw Size	Raw Address	Reloc Address	Linenumbers	Relocations N...	Linenumbers ...	Characteristics
Byte[8]	Dword	Dword	Dword	Dword	Dword	Dword	Word	Word	Dword
.text	00336F9C	00001000	00337000	00000400	00000000	00000000	0000	0000	60000020
.rdata	000F8E52	00338000	000F9000	00337400	00000000	00000000	0000	0000	40000040
.data	000251AC	00431000	00011800	00430400	00000000	00000000	0000	0000	C0000040
.pdata	0001A424	00457000	0001A600	00441C00	00000000	00000000	0000	0000	40000040
.rsrc	00163CD0	00472000	00163E00	0045C200	00000000	00000000	0000	0000	40000040
.reloc	00005238	005D6000	00005400	005C0000	00000000	00000000	0000	0000	42000040

راح نركز على **VA** وهو الخاص بالميموري وايضا **RA** وهو الخاص بالهارد طبعا يصير فيها عمليات حسابية عشان تحسب بهذي الطريقة مثال وبدرج صورة تختصر الموضوع ومقالة للتعمق فيها

RVA = VA – ImageBase
or
VA = RA + ImageBase

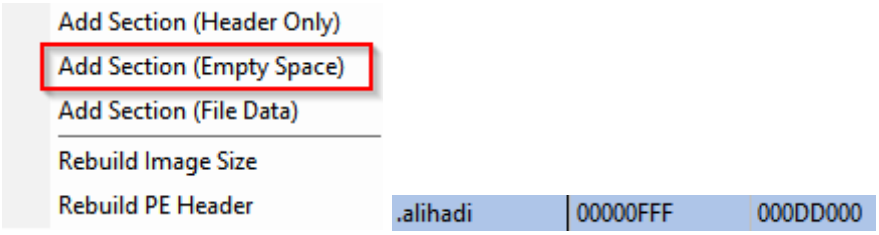
Assume we have the following example, which shows how the file is presented on disk and in memory



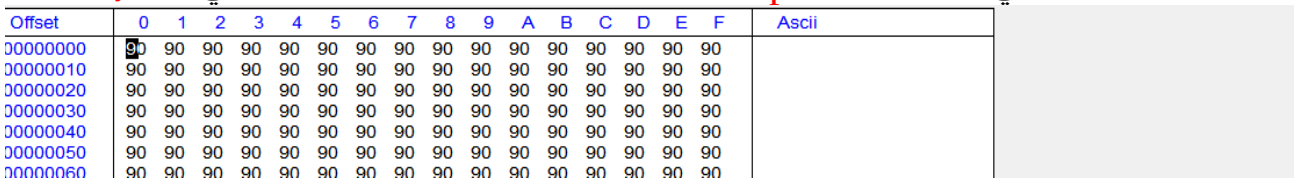
Pe Infection

بعد مافهمنا كيف يعمل الملف التنفيذي وكيف ينشئ الأن نبدا موضوعنا الأساسي وعليكم بربط الأحداث المهمة والنصوص المميزة باللون الأحمر

الأن عشان نعمل حقن كود خبيث راح نعمل في البداية **Section Header** خاص فينا وراح نسميها **.alihadi**
وراح نخلي حجم size (FFF) الحجم غير مهم الاهم حجم عشان تقدر تحقن فيه كود خبيث

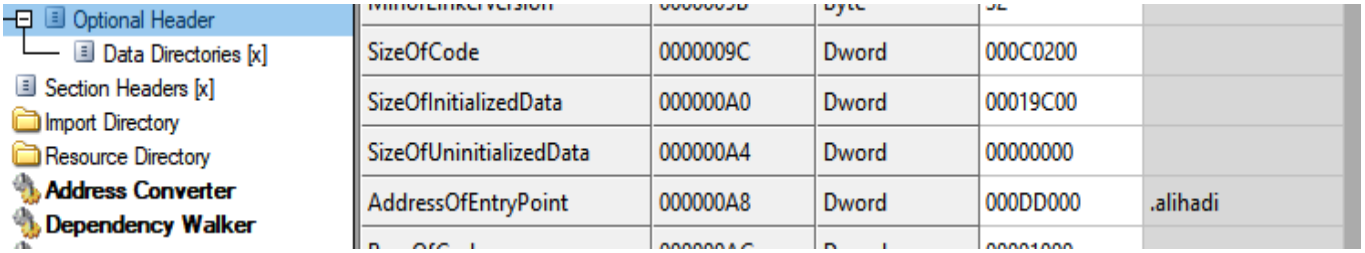


تم الإنشاء الأن لنضمن نعمل في كامل المساحة nop عشان نتاكد من وصولنا بعد التعديل في **AddressOfEntryPoint**

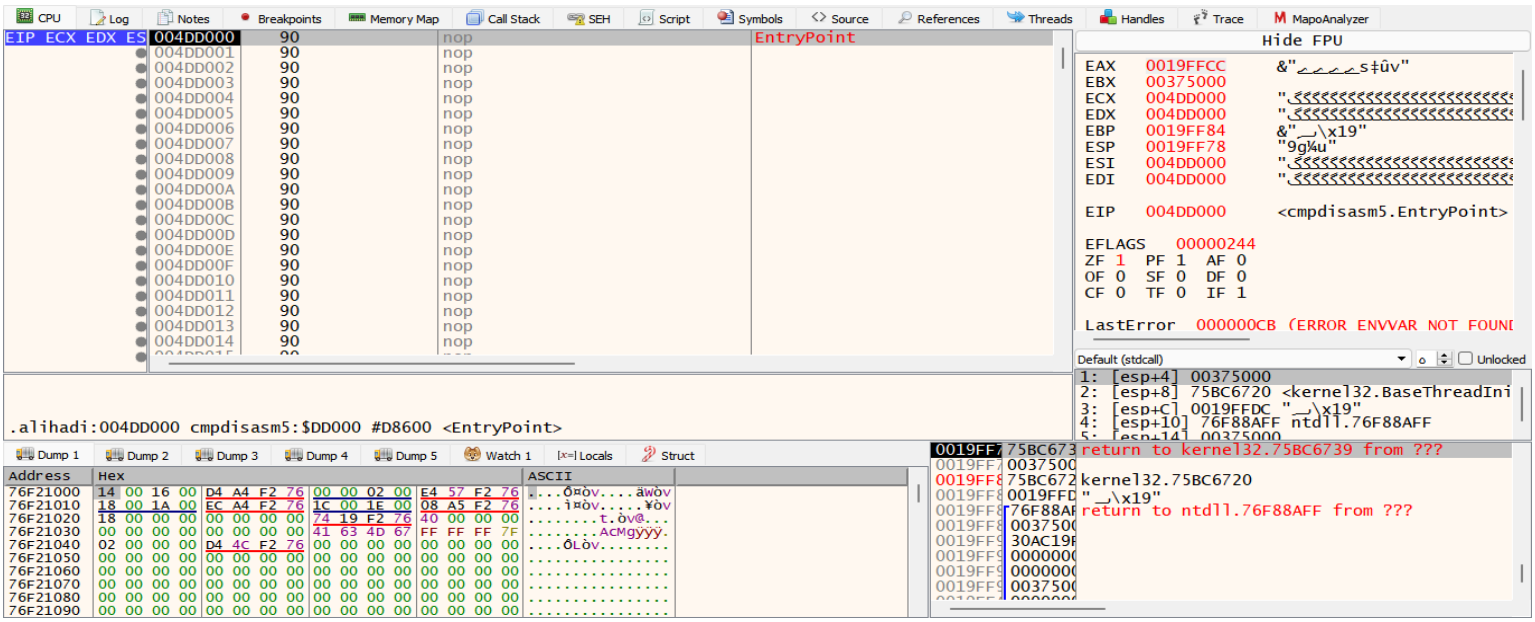


الان راح نأخذ **VA** الخاص بالسيكشن علي هادي وهي **0xDD000**

وراح نخليه في **AddressOfEntryPoint** ولكن قبل هذا لابد نحفظ نقطة البداية السابقة وهي **(0x401000)**



كما نلاحظ انها صارت نقطة البداية علي هادي اذا لاحظت وربطت الاحداث كفو عليك الأن لنتوجه إلى **x32dbg**



جميل جدا الأن لو نلاحظ وصل لنقطة البداية حق سيكشن علي هادي وبنفس العنوان اللي أنا أبيه الأن ماذا سنعمل ياترى ؟

الأن سنعمل كالتالي نضيف الكود الخبيث (**shellcode**) وراح يعمل الكود مسج بوكس بهذا الشكل وبعد الإستغلال راح نخليه يعمل **jmp** ل **EntryPoint** القديمة

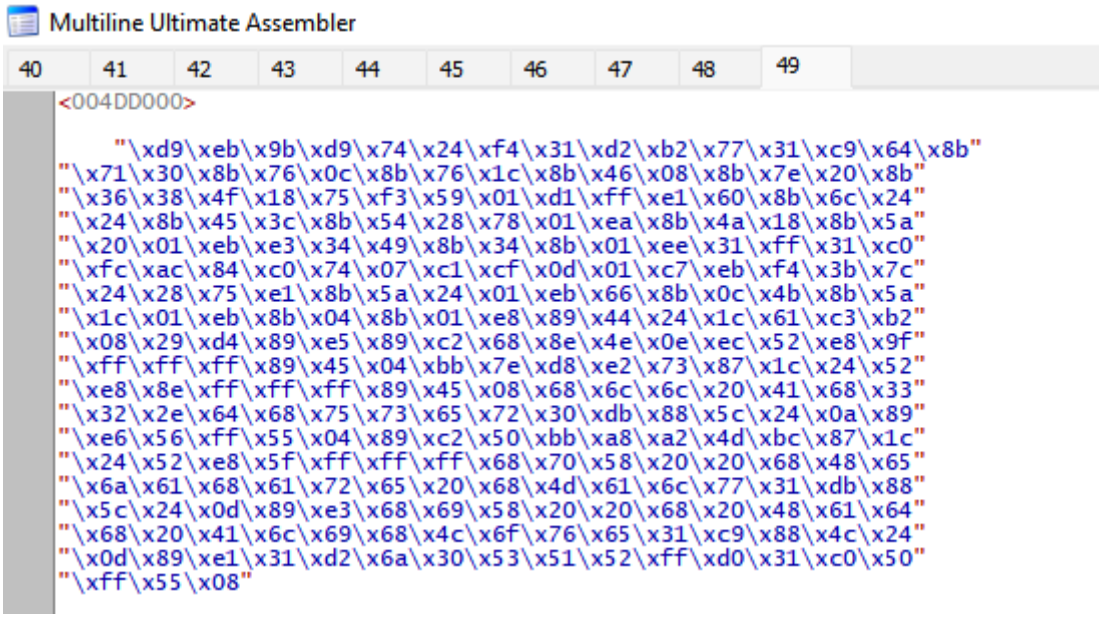
راح نعمل الشل كود كالتالي نستطيع عملها يدوي وجلب العنوان حق المسج بوكس ولكن اداة**Msfvenom** موفرتلي الشغل كله

راح نعمل الأتي

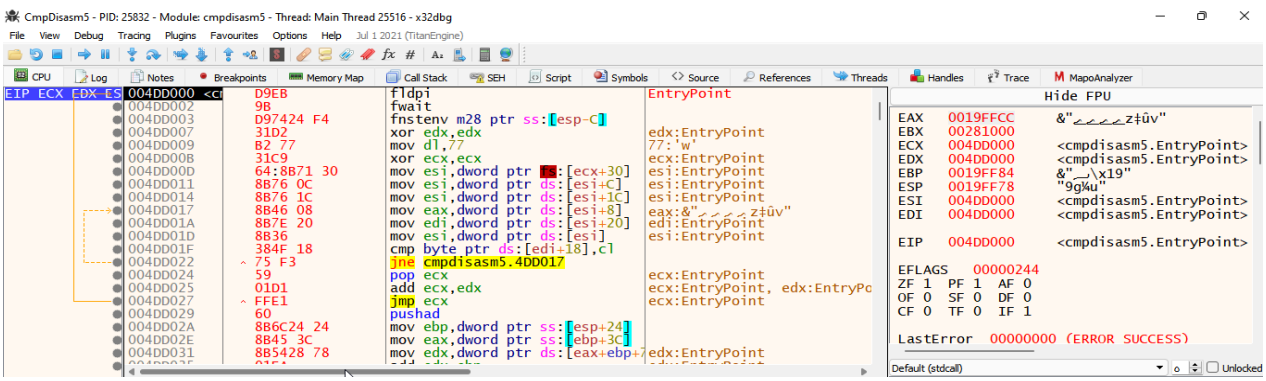
```
┌──(kali㉿kali)-[~/Desktop]
└─$ msfvenom -p windows/messagebox ICON="WARNING" TEXT="Love Ali Hadi" TITLE="Malware Hejap" -f c -a x86 --platform win

No encoder specified, outputting raw payload
Payload size: 273 bytes
Final size of c file: 1173 bytes
unsigned char buf[] =
"\xd9\xeb\x9b\xd9\x74\x24\xf4\x31\xd2\xb2\x77\x31\xc9\x64\x8b"
"\x71\x30\x8b\x76\x0c\x8b\x76\x1c\x8b\x46\x08\x8b\x7e\x20\x8b"
"\x36\x38\x4f\x18\x75\xf3\x59\x01\xd1\xff\xe1\x60\x8b\x6c\x24"
"\x24\x8b\x45\x3c\x8b\x54\x28\x78\x01\xea\x8b\x4a\x18\x8b\x5a"
"\x20\x01\xeb\xe3\x34\x49\x8b\x34\x8b\x01\xee\x31\xff\x31\xc0"
"\xfc\xac\x84\xc0\x74\x07\xc1\xcf\x0d\x01\xc7\xeb\xf4\x3b\x7c"
"\x24\x28\x75\xe1\x8b\x5a\x24\x01\xeb\x66\x8b\x0c\x4b\x8b\x5a"
"\x1c\x01\xeb\x8b\x04\x8b\x01\xe8\x89\x44\x24\x1c\x61\xc3\xb2"
"\x08\x29\xd4\x89\xe5\x89\xc2\x68\x8e\x4e\x0e\xec\x52\xe8\x9f"
"\xff\xff\xff\x89\x45\x04\xbb\x7e\xd8\xe2\x73\x87\x1c\x24\x52"
"\xe8\x8e\xff\xff\xff\x89\x45\x08\x68\x6c\x6c\x20\x41\x68\x33"
"\x32\x2e\x64\x68\x75\x73\x65\x72\x30\xdb\x88\x5c\x24\x0a\x89"
"\xe6\x56\xff\x55\x04\x89\xc2\x50\xbb\xa8\xa2\x4d\xbc\x87\x1c"
"\x24\x52\xe8\x5f\xff\xff\xff\x68\x70\x58\x20\x20\x68\x48\x65"
"\x6a\x61\x68\x61\x72\x65\x20\x68\x4d\x61\x6c\x77\x31\xdb\x88"
"\x5c\x24\x0d\x89\xe3\x68\x69\x58\x20\x20\x68\x20\x48\x61\x64"
"\x68\x20\x41\x6c\x69\x68\x4c\x6f\x76\x65\x31\xc9\x88\x4c\x24"
"\x0d\x89\xe1\x31\xd2\x6a\x30\x53\x51\x52\xff\xd0\x31\xc0\x50"
"\xff\x55\x08";
```

```
msfvenom -p windows/messagebox ICON="WARNING" TEXT="Love Ali Hadi" TITLE="Malware Hejap" -f c -a x86 --platform win
```

راح ننسخ الشل كود ونلصقه في عنوان السيكشن **EntryPoint** حق علي هادي ونلصق الشل كود من مميزات الاسمبلر في x32dbg يقدر يفهم opcode ويسوي disassembler عليه ويكتب فيه

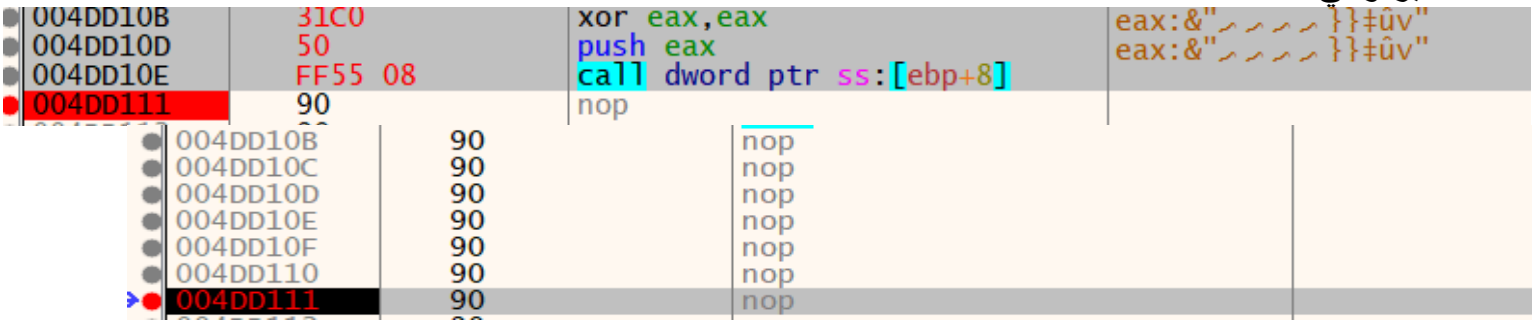


نشوف الأن كتب فعلا

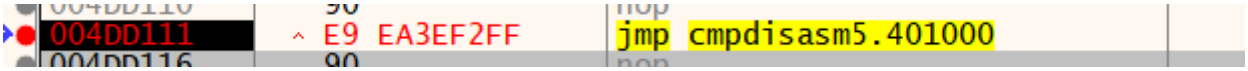


الأن نعمل **breakboint** عند اخر نقطة في الشل كود ونشوف هل فعلا يتنفذ الشل كود

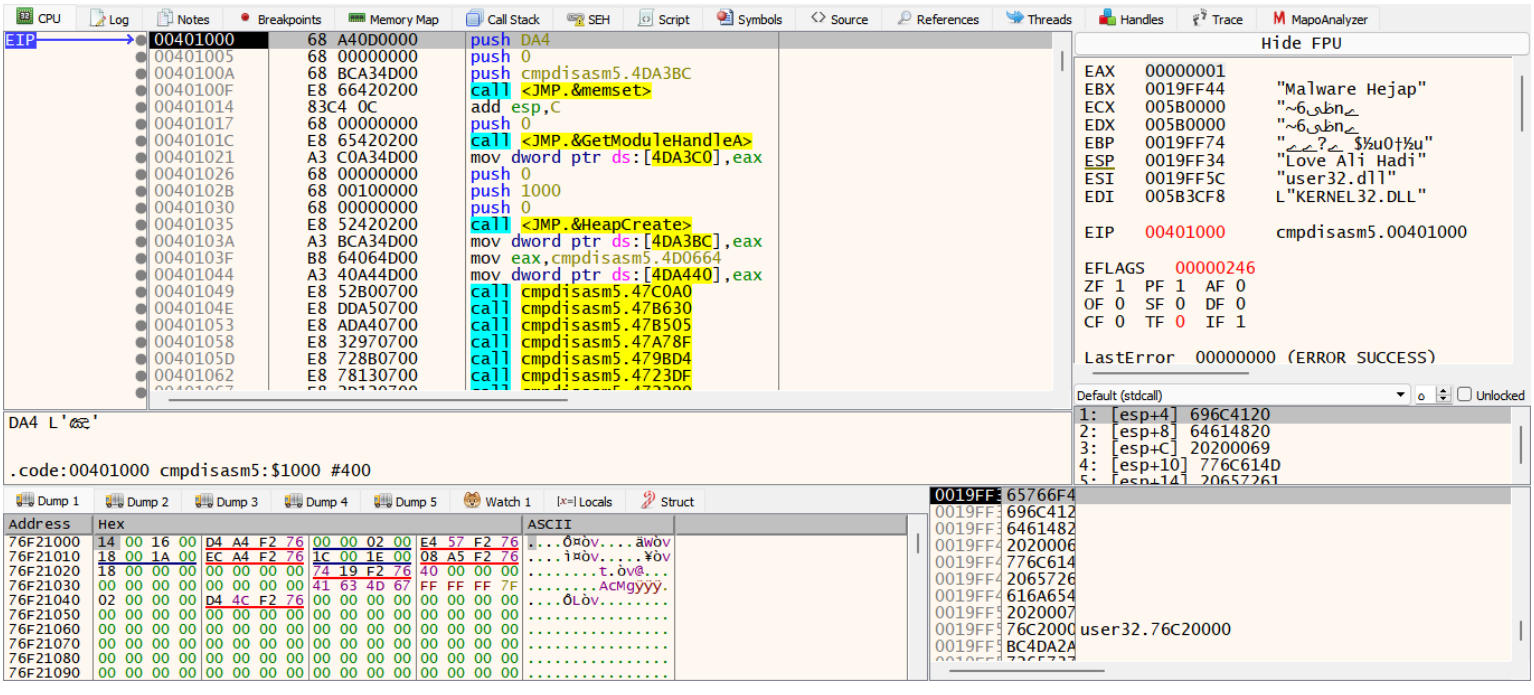
ولكن نلاحظ بانه يتوقف دايركت ومايتوقف في نفس المكان اللي ابيه الحل بسيط نعمل نوب على الثلاث تعليمات قبل الاخير وهي

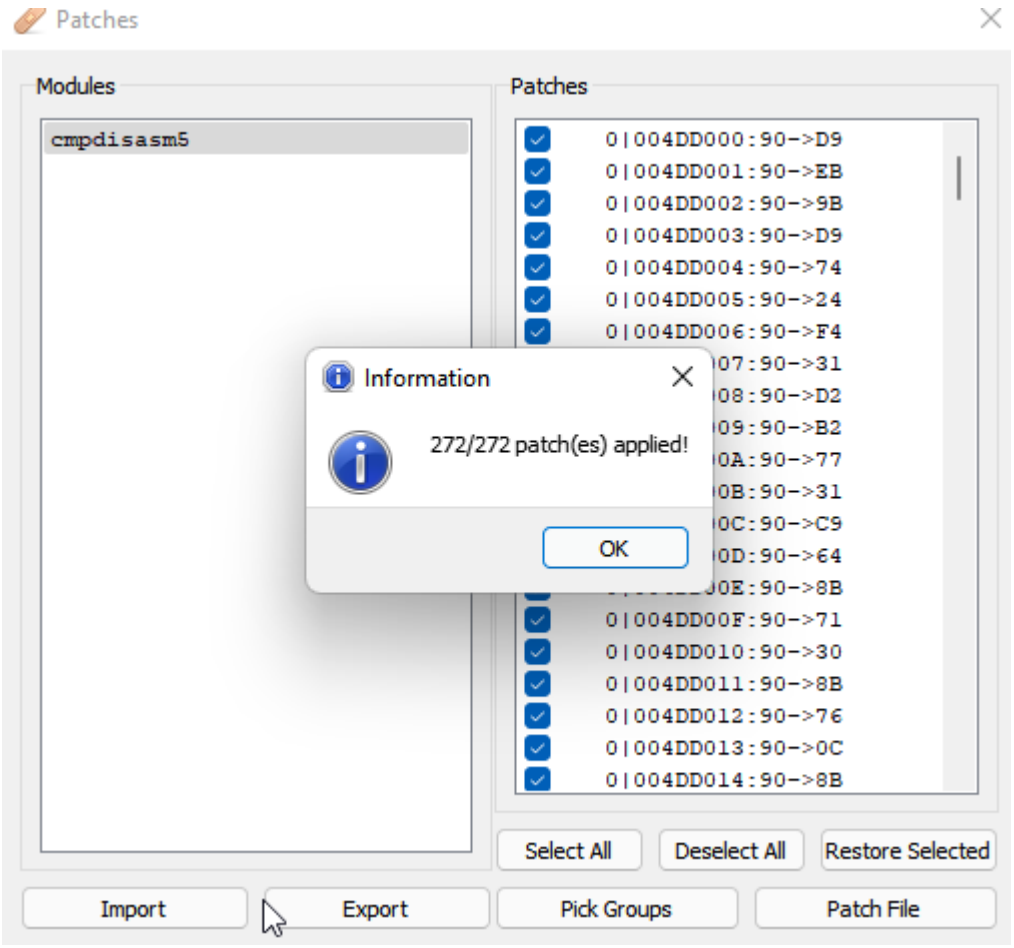


الان لو تلاحظ توقف عند نفس العنوان اللي ابيه
الان اعمل كالتالي اعمل **jmp** ل **AddressOfEntryPoint** وهي (**0x00401000**) عشان ينفذ الشل كود ويشغل البرنامج طبيعي ولا كانه حصل شيء أبدا

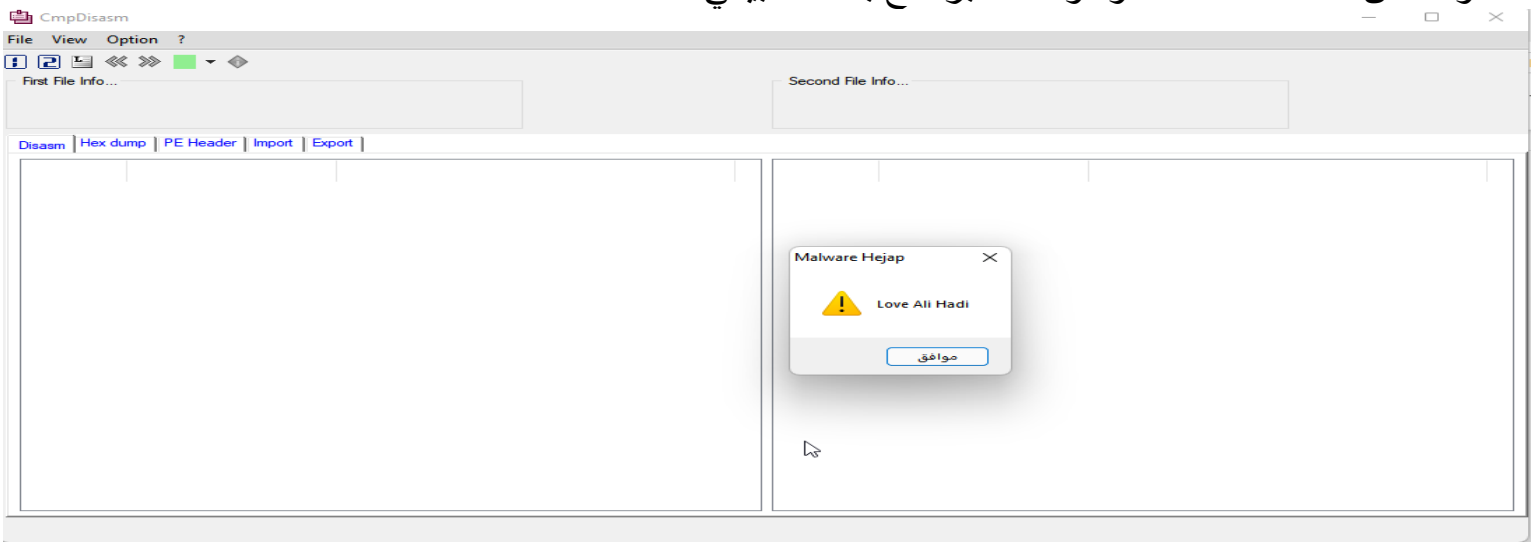


الأن نلاحظ أنه وصلني للنقطة القديمة حلو جدا الان نحفظ التعديل ونشغله طبيعي





نشوفه الأن فعلا شغل الشل كود وشغل البرنامج بشكل طبيعي



وكذا نقول خلصنا تقدر تعمل الشل الكود اللي تبيه انت من اساليب الحماية **File verification** وهو عمل مقارات مع الملف الاصلي من هاش او غيره او البحث عن أي شل مود وهو مايسمى **Pattern recognition** وتقدر تعمل الشغل كله بشكل اوتميشن توجه للمقالة وايضا هناك طريقة اخرى وهي ب Code cave توجه للمقالة لتعرف كيف عملها

الشكر لله ثم للدكتور علي هادي
والمدرب مصطفى في توصيلهم للمعلومة بشكل دقيق وبسيط

المصادر :

https://www.youtube.com/c/AliHadiC5W
https://my.ine.com/INE/courses/f7be49bd/malware-analysis
https://my.ine.com/search?Instructors=AliHadi
/https://wes4m.io/PeInfection
/https://tech-zealots.com/malware-analysis/understanding-concepts-of-va-rva-and-offset
/https://osandamalith.com/2020/07/19/exploring-the-ms-dos-stub
/https://axcheron.github.io/code-injection-with-python

حسابي للتواصل :

https://twitter.com/matrix0700

الكاتب : حجاب زائري
المدرب : علي هادي
المشرف : مصطفى

الحمد لله