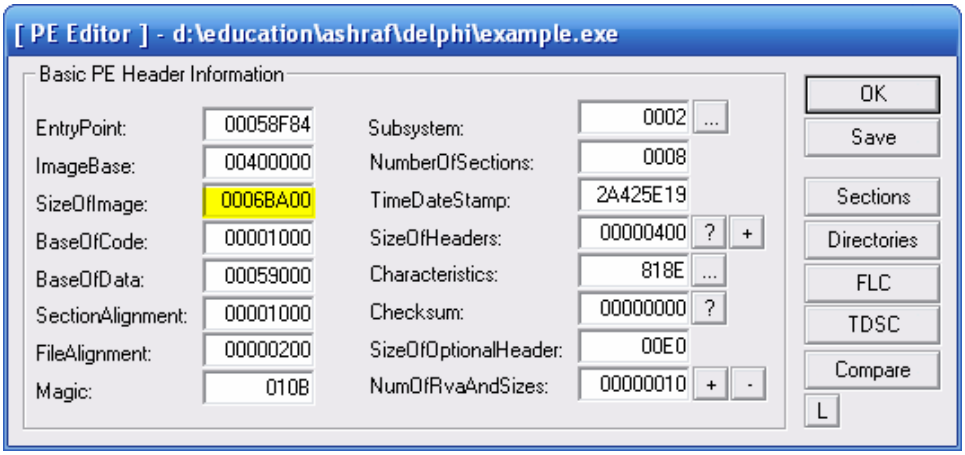
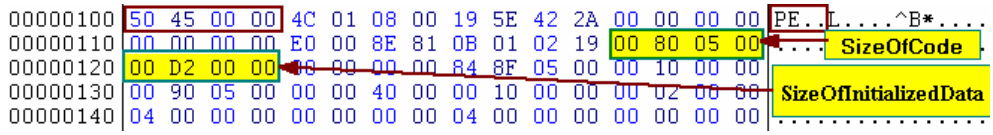
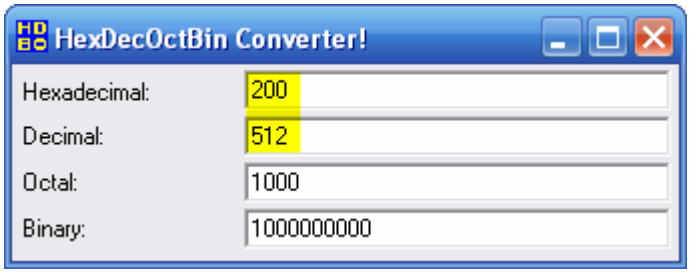
الشكل التالي هو تكملة للدرس الخامس



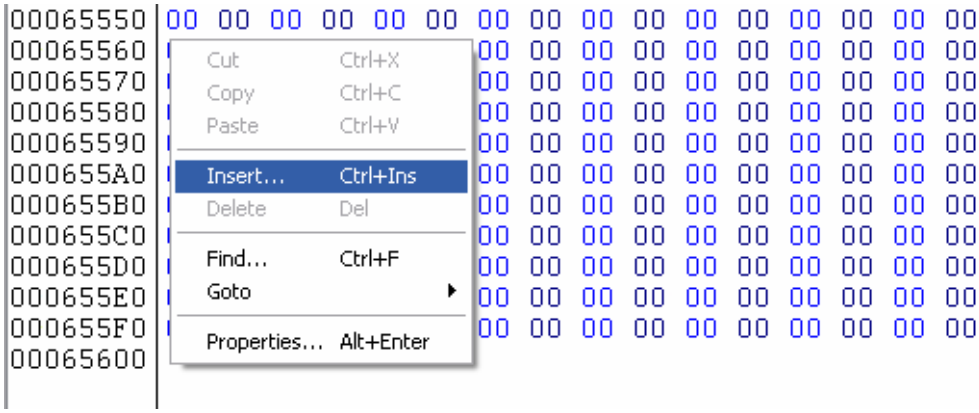
سوف نُضيف القيمة 200h إلي الدالة SizeOfImage لأنها الدالة الخاصة بحجم الملف وسوف يصبح الناتج 0006BA00h+200h = **0006BC00h** وإذا قمت بتشغيل البرنامج سوف تري نفس الرسالة وإذا إستعرضنا الجزء الخاص بـPE Header فسوف نري هذا الشكل :



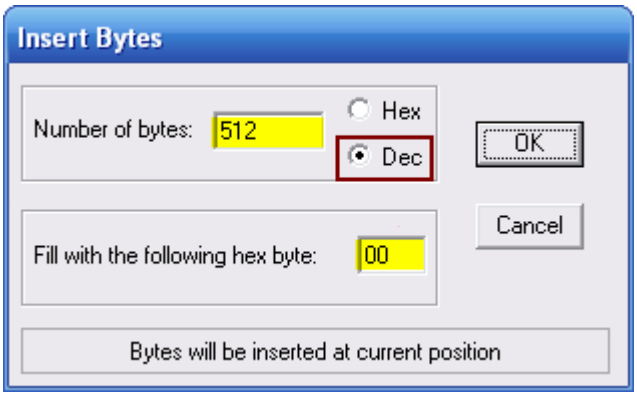
كما تري فتوجد دوال أخري لابد من تغييرها حتي يعمل البرنامج بشكل صحيح والـOffset الخاص بالدالة SizeOfCode هو 11C أما الـOffset الخاص بالدالة SizeOfInitializedData هو 120 وإذا عكسنا قيمة الدالة SizeOfCode لكي نستطيع التعامل معها سوف تصبح 00058000h وسوف نُضيف لها أيضاً القيمة 200h لتصبح 00058200h أما إذا عكسنا قيمة الدالة SizeOfInitializedData فسوف تصبح 0000D200h وإذا أضفنا لها القيمة 200h سوف تصبح 0000D400h والأن يبقي شئ واحد وهو إضافة عدد البايتات التي تساويها القيمة 200h فإذا قمنا بتحويل هذه القيمة إلي Decimal فسوف تساوي 512 بايت كما في الشكل التالي :



لذلك سوف نذهب إلي برنامج Hex Workshop ثم نذهب إلي أخر البرنامج ثم قف علي الـOffset 00065600 ثم إضغط Right Click ثم إختر هذا الإختيار :



ثم ضع هذه البيانات :



ثم شغل البرنامج وسوف تري أن البرنامج يعمل بدون أي مشاكل.

والأن سوف نتطرق إلي معرفة كيفية إضافة قسم جديد للبرنامج وقد يفيد إضافة قسم جديد للبرنامج أن تقوم بفك تشفير برنامج ما ثم تضع بيانات فك التشفير في قسم جديد وسوف نقوم بإضافة 100h من البايتات إلي البرنامج وكما عرفنا قبل ذلك أن بيانات القسم حجمها 40 بايت لذلك سوف نذهب إلي أخر قسم كما في الشكل التالي :

```
00000310 2E 72 73 72 63 00 00 00 00 3A 00 00 00 80 06 00 .rsrc....:......
00000320 00 3A 00 00 00 1C 06 00 00 00 00 00 00 00 00 00 .:..............
00000330 00 00 00 00 40 00 00 50 00 00 00 00 00 00 00 00 ....@..P........
00000340 00 00 00 00 00 C0 06 00 00 00 00 00 00 56 06 00 .............V..
00000350 00 00 00 00 00 00 00 00 00 00 00 00 40 00 00 50 ............@..P
00000360 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
```

كما تري فقد إخترنا 40 بايت بعد القسم rsrc والأن المعلومات الخاصة بقسم rsrc هي كالتالي :

VirtualAddress : 00068000h

VirtualSize : 00003A00h

RawOffset : 00061C00h

RawSize : 00003A00h

والـRVA الخاص بالـVirtualAddress للقسم rsrc هو 00068000h وهذه القيمة هي مجموع الـVirtualAddress والـVirtualSize للقسم السابق وهو ".reloc" فمثلاً لو قمنا بجمع هذه القيم للقسم السابق سوف تكون طريقة الجمع هكذا :

00061000h + 00006108h = 00067108h

وبما أن قيمة الدالة SectionAlignment = 1000h عندئذ سوف يتم الجمع لأقرب 1000h وسوف يكون الناتج 00068108h ومع التقريب لأقرب 1000h سوف يُصبح الناتج النهائي 00068000h وهو الـVirtualAddress الخاص بالقسم rsrc ولذلك سوف نجمع الـVirtualAddress والـVirtualSize الخاص بالقسم rsrc ليصبح الناتج كما يلي :

00068000h + 00003A00h = 0006BA00h + 1000h = 0006CA00h

وبعد التقريب سوف يُصبح الناتج النهائي 0006C000h وإذا لاحظت أيضاً سوف تجد أن الـRawOffset هو مجموع الـRawOffset والـRawSize للقسم السابق وعندئذ سوف تصبح بيانات القسم الجديد كالتالي :

VirtualAddress : 0006C000h ---→ 00 C0 06 00

VirtualSize : 100h ---→ 00 01 00 00

RawOffset : 00065600h ---→ 00 56 06 00

RawSize : 100h ---→ 00 01 00 00

Characteristics : E00000060 ---→ 60 00 00 E0

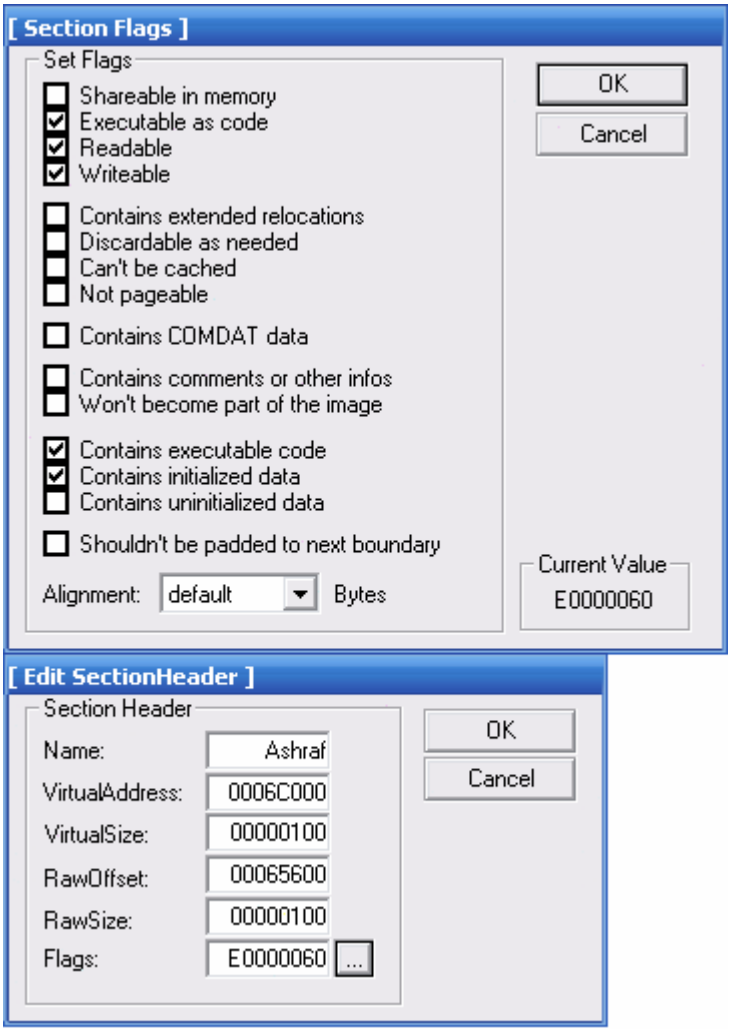
وسوف نضع هذه البيانات في برنامج Hex Workshop كما يلي :

```
00000330 00 00 00 00 40 00 00 50 41 73 68 72 61 66 00 00 ....@..PAshraf..
00000340 00 01 00 00 00 C0 06 00 00 01 00 00 00 56 06 00 .............V..
00000350 00 00 00 00 00 00 00 00 00 00 00 00 60 00 00 E0 ............`...
00000360 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
```

ثم نقوم بإلصاق 256 بايت في أخر البرنامج ثم نغير قيمة الدالة NumberOfSections إلي العدد ٩ وهو عدد الأقسام وإذا قمنا بإستعراض أقسام البرنامج سوف نري هذا الشكل :

Name	VOffset	VSize	ROffset	RSize	Flags
CODE	00001000	00057FCC	00000400	00058000	60000020
DATA	00059000	0000111C	00058400	00001200	C0000040
BSS	0005B000	00000C51	00059600	00000000	C0000000
.idata	0005C000	000021BC	00059600	00002200	C0000040
.tls	0005F000	00000010	0005B800	00000000	C0000000
.rdata	00060000	00000018	0005B800	00000200	50000040
.reloc	00061000	00006108	0005BA00	00006200	50000040
.rsrc	00068000	00003A00	00061C00	00003A00	50000040
Ashraf	0006C000	00000100	00065600	00000100	E0000060

كما تري فالشكل السابق يشير إلي القسم الجديد الذي أنشأناه وإذا إستعرضنا البيانات الخاصة بهذا القسم سوف تري الشكل التالي :



وإذا لاحظت سوف تجد أن الدالة SizeOfImage هي مجموع الـVirtualAddress والـVirtualSize لأخر قسم وسوف يتم توضيحها في الشكل التالي :

[PE Editor] - d:\education\ashraf\delphi\example.exe

Basic PE Header Information

EntryPoint:	00058F84	Subsystem:	0002
ImageBase:	00400000	NumberOfSections:	0009
SizeOfImage:	0006C100	TimeDateStamp:	2A425E19
BaseOfCode:	00001000	SizeOfHeaders:	00000400
BaseOfData:	00059000	Characteristics:	818E
SectionAlignment:	00001000	Checksum:	00000000
FileAlignment:	00000200	SizeOfOptionalHeader:	00E0
Magic:	010B	NumOfRvaAndSizes:	00000010

OK | Save | Sections | Directories | FLC | TDSC | Compare | L

Name	VOffset	VSize	ROffset	RSize	Flags
BSS	0005B000	00000C51	00059600	00000000	C0000000
.idata	0005C000	000021BC	00059600	00002200	C0000040
.tls	0005F000	00000010	0005B800	00000000	C0000000
.rdata	00060000	00000018	0005B800	00000200	50000040
.reloc	00061000	00006108	0005BA00	00006200	50000040
.rsrc	00068000	00003A00	00061C00	00003A00	50000040
Ashraf	0006C000	00000100	00065600	00000100	E0000060

كمـــا تـــري فالـــشكل الـــسابق يوضـــح أن الدالـــة SizeOfImage مجمـــوع الدالـــة VirtualAddress والدالة VirtualSize. ومن الممكن أن نجعل البرنامج يبدأ بعنوان من القسم الجديد فمثلاً سوف نذهب إلي هذا العنوان :

```
000655C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000655D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000655E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000655F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00065600 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00065610 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00065620 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00065630 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00065640 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00065650 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

ثم نُضيف له هذه القيم :

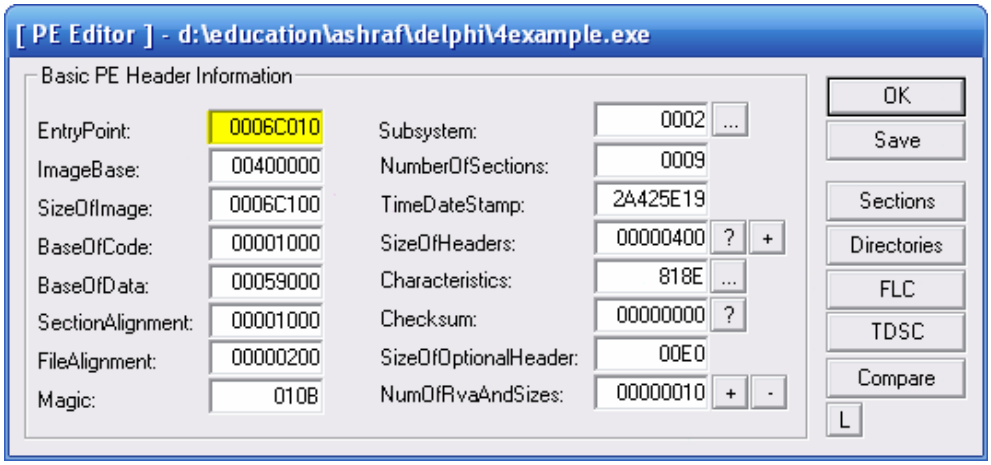
```
00065600 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00065610 B8 84 8F 45 00 FF E0 90 00 00 00 00 00 00 00 00 ...E............
00065620 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
```

وهذه الأكواد هي الخاصة بهذه الأوامر :

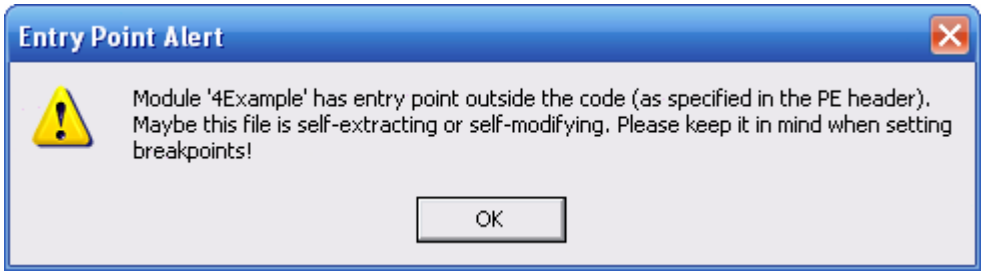
MOV EAX, 00458F84

JMP EAX

ثم قم بحفظ هذه الأكواد والعنوان الذي ألصقنا به هذه الأكواد بعد تحويلة سوف يصبح 0046C010 والـRVA الخاص به هو 0006C010 لذلك سوف نغير عنوان الـEP إلي العنوان 0006C010 كما في الشكل التالي :



وعند تحميل البرنامج في برنامج Ollydbg سوف تري هذه الرسالة :



وهذه الرسالة تخبرك بأن الـEntry Point ليس مكانه في قسم الـCODE وإنما في جزء أخر وعند الضغط علي زر OK سوف يتم تحميل البرنامج وسوف تري هذا الشكل :

```
Address   Hex dump     Disassembly
0046C010  B8 848F4500  MOV EAX,4Example.00458F84
0046C015  FFE0         JMP EAX
0046C017  90           NOP
0046C018  0000         ADD BYTE PTR DS:[EAX],AL
0046C01A  0000         ADD BYTE PTR DS:[EAX],AL
0046C01C  0000         ADD BYTE PTR DS:[EAX],AL
0046C01E  0000         ADD BYTE PTR DS:[EAX],AL
0046C020  0000         ADD BYTE PTR DS:[EAX],AL
0046C022  0000         ADD BYTE PTR DS:[EAX],AL
0046C024  0000         ADD BYTE PTR DS:[EAX],AL
0046C026  0000         ADD BYTE PTR DS:[EAX],AL
0046C028  0000         ADD BYTE PTR DS:[EAX],AL
0046C02A  0000         ADD BYTE PTR DS:[EAX],AL
0046C02C  0000         ADD BYTE PTR DS:[EAX],AL
00458F84=4Example.00458F84
EAX=00000000
```

كما تري فقد تم الذهاب إلي العنوان الذي حددناه كبداية تشغيل البرنامج وإذا ضغطت F8 سوف يعمل البرنامج بشكل طبيعي وسوف يذهب إلي الـOEP الحقيقي للبرنامج.

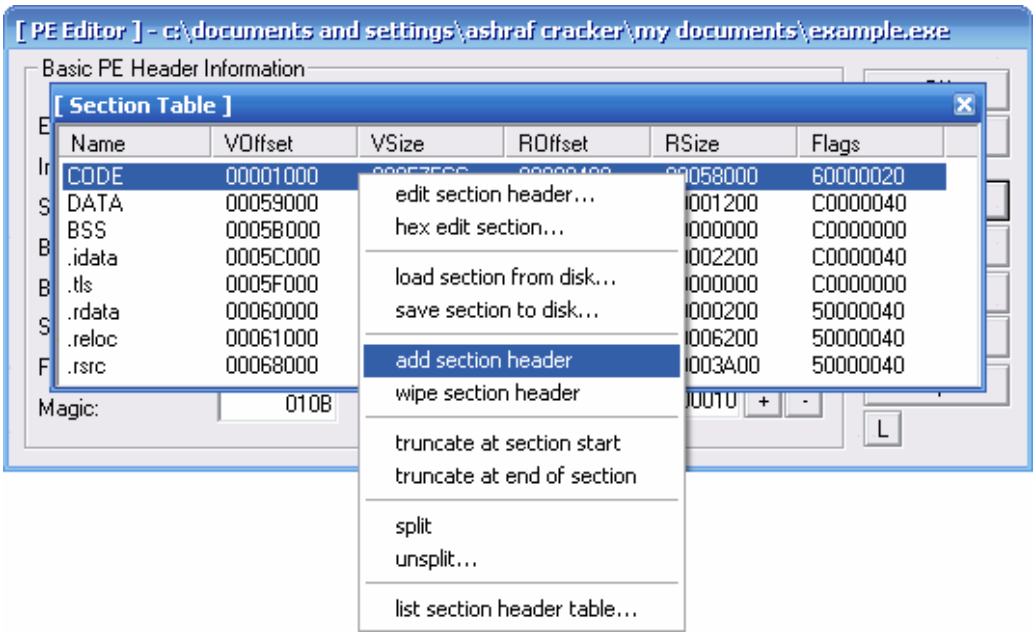
والأن سوف نتطرق إلي معرفة كيفية إستيراد وظيفة من ملف "DLL" وكما ذكرنا سابقاً أن الـImport Table له بنية خاصة به وهي **IMAGE_IMPORT_DESCRIPTOR** وفي برنامجنا هذا يوجد ١٣ جدول وعندما نريد أن نستورد وظيفة من ملف ما سوف نقوم بزيادة عدد الجدوال إلي ١٤ جدول وتوجد برامج تعمل علي إستيراد الوظائف أوتوماتيكياً ولكن إستيراد هذه الوظائف بنفسك وبدون الإعتماد علي برامج مساعدة أفضل ويزيدك خبرة في كيفية إضافة وظيفة أو أكثر إلي برنامجك وكما ذكرنا سابقاً أيضاً أن بيانات الـImport Table تأخذ بياناتها ٢٠ بايت وتوجد خطوات لابد من إتباعها لكي نُضيف وظيقة جديدة للبرنامج وهذه الخطوات تتلخص في إيجاد هذه القيم :

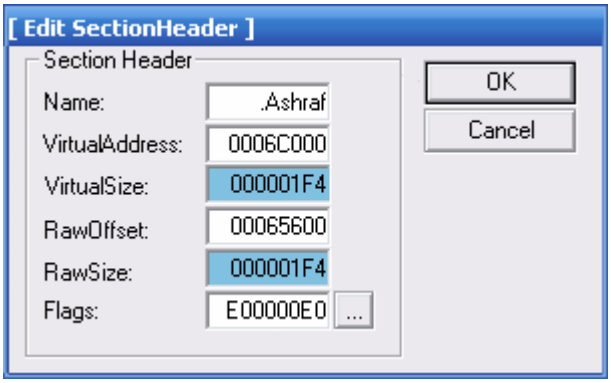
١- نقل الجداول إلي مكان أخر وإضافة جدول جديد يشمل بيانات الوظيفة الجديدة.
٢- تغيير قيمة الـDirectory الخاصة بالـImport Table إلي العنوان الجديد .
٣- إضافة الوظيفة التي سوف نقوم بإستيرادها إلي الوظائف الموجودة في البرنامج.
٤- إضافة المعلومات الخاصة بهذه الوظيفة إلي الجدول الذي قمنا بإضافته.
٥- تغيير الـEP إلي مكان جديد ثم إضافة أمر ينادي هذه الوظيفة ثم القفز إلي الـEP الحقيقي مرة أخري.

كما رأينا فالخطوات السابقة لابد من إتمامها لكي نضيف وظيفة وتعمل بشكل صحيح ولذلك سوف نذهب إلي المكان الخاص بجدوال الـImport كما في هذا الشكل :

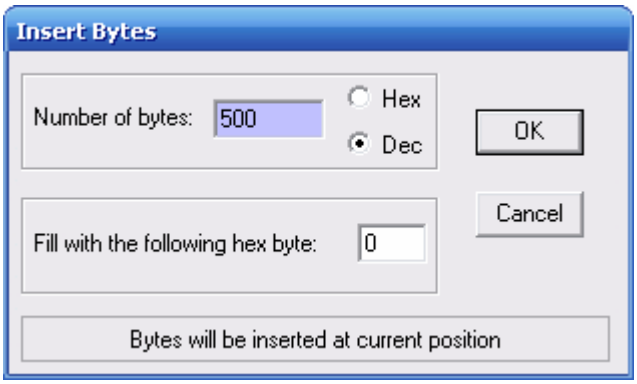
```
000595E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ........
000595F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ........
00059600 00 00 00 00 00 00 00 00 00 00 00 00 40 C7 05 00 ........
00059610 04 C1 05 00 00 00 00 00 00 00 00 00 00 00 00 00 ........
00059620 20 CA 05 00 AC C1 05 00 00 00 00 00 00 00 00 00  .......
00059630 00 00 00 00 66 CA 05 00 C0 C1 05 00 00 00 00 00 ....f...
00059640 00 00 00 00 00 00 00 00 A6 CA 05 00 D0 C1 05 00 ........
00059650 00 00 00 00 00 00 00 00 00 00 00 00 EE CA 05 00 ........
00059660 E0 C1 05 00 00 00 00 00 00 00 00 00 00 00 00 00 ........
00059670 3A CB 05 00 F4 C1 05 00 00 00 00 00 00 00 00 00 :.......
00059680 00 00 00 00 7A CB 05 00 04 C2 05 00 00 00 00 00 ....z...
00059690 00 00 00 00 00 00 00 00 64 CF 05 00 F4 C2 05 00 ........
000596A0 00 00 00 00 00 00 00 00 00 00 00 00 E0 D3 05 00 ........
000596B0 FC C3 05 00 00 00 00 00 00 00 00 00 00 00 00 00 ........
000596C0 1E DE 05 00 80 C6 05 00 00 00 00 00 00 00 00 00 ........
000596D0 00 00 00 00 34 DE 05 00 88 C6 05 00 00 00 00 00 ....4...
000596E0 00 00 00 00 00 00 00 00 C4 DF 05 00 E4 C6 05 00 ........
000596F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ........
00059700 00 00 00 00 4E C7 05 00 66 C7 05 00 7E C7 05 00 ....N...
```

كما تري الجداول السابقه نريد أن ننقلها إلي مكان أخر وإذا قمت بالنزول إلي أسفل سوف تري أنه لا توجد في هذا القسم مكان لإلصاق هذه الجداول وإضافة جدول جديد لذلك سوف نقوم بإنشاء جدول جديد عن طريق برنامج LordPE كما في هذا الشكل :





كما تري فلقد قمنا بإضافة حجم القسم فقط وكل البيانات تم إيجادها بطريقة آلية ولكن نحن ضفنا حجم القسم 1F4 وهو يساوي 500 بايت ثم قم بحفظ هذه البيانات وإذا أردت تشغيل البرنامج سوف تري أن البرنامج لا يعمل لذلك إذهب إلي برنامج Hex Workshop ثم حمل الملف بداخلة ثم إذهب إلي أخر الملف ثم أضف إضغط Right Click ثم إختر insert ثم أضف هذه القيمة :



ثم قم بحفظ هذه البيانات وسو تري أن البرنامج يعمل بشكل صحيح ثم قم بنسخ بيانات الجداول السابقة إلي هذا المكان :

```
000656D0 00 00 00 00 00 00 00 00 00 00 00 00 40 C7 05 00
000656E0 04 C1 05 00 00 00 00 00 00 00 00 00 00 00 00 00
000656F0 20 CA 05 00 AC C1 05 00 00 00 00 00 00 00 00 00
00065700 00 00 00 00 66 CA 05 00 C0 C1 05 00 00 00 00 00
00065710 00 00 00 00 00 00 00 00 A6 CA 05 00 D0 C1 05 00
00065720 00 00 00 00 00 00 00 00 00 00 00 00 EE CA 05 00
00065730 E0 C1 05 00 00 00 00 00 00 00 00 00 00 00 00 00
00065740 3A CB 05 00 F4 C1 05 00 00 00 00 00 00 00 00 00
00065750 00 00 00 00 7A CB 05 00 04 C2 05 00 00 00 00 00
00065760 00 00 00 00 00 00 00 00 64 CF 05 00 F4 C2 05 00
00065770 00 00 00 00 00 00 00 00 00 00 00 00 E0 D3 05 00
00065780 FC C3 05 00 00 00 00 00 00 00 00 00 00 00 00 00
00065790 1E DE 05 00 80 C6 05 00 00 00 00 00 00 00 00 00
000657A0 00 00 00 00 34 DE 05 00 88 C6 05 00 00 00 00 00
000657B0 00 00 00 00 00 00 00 00 C4 DF 05 00 E4 C6 05 00
000657C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000657D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

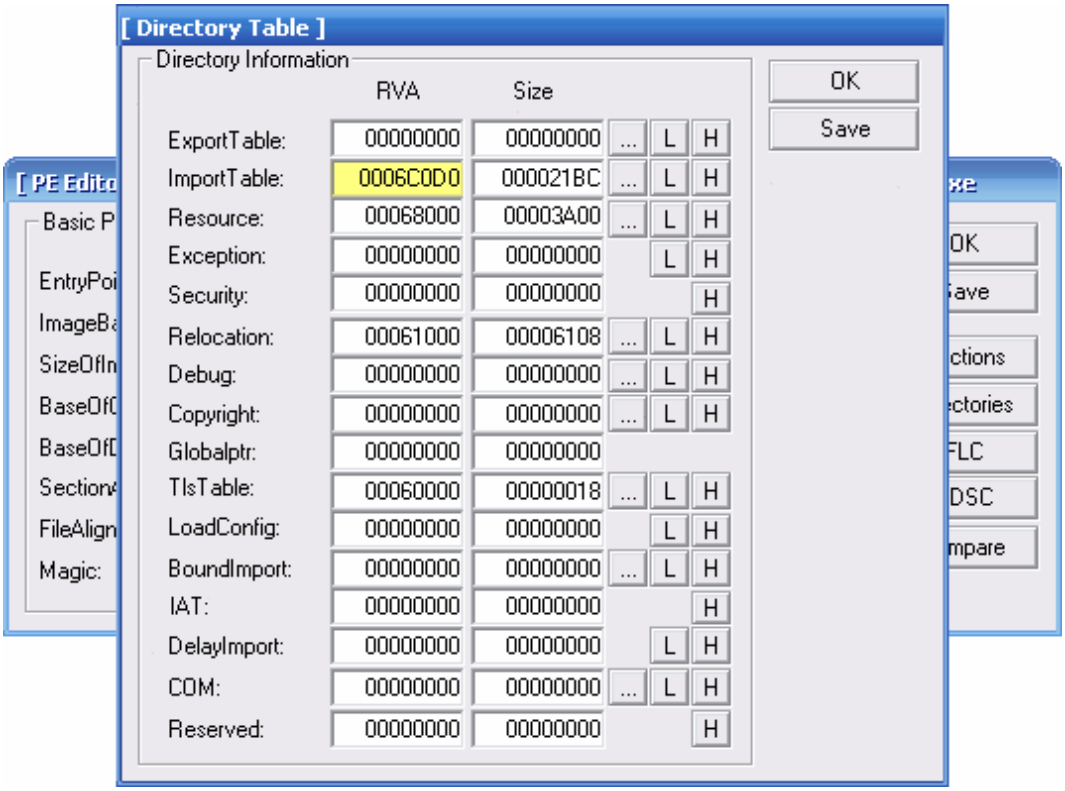
ثم قم بحفظ هذه البيانات وكما تري العنوان الذي قمنا بحفظ هذه البيانات عنده هو 000656D0 وسوف نحوله إلي VA حتي يستطيع البرنامج فيما بعد أن يقرأه من الذاكرة كما يلي :

VA = Raw Offset + (V.Offset of Section – R.Offset of Section)

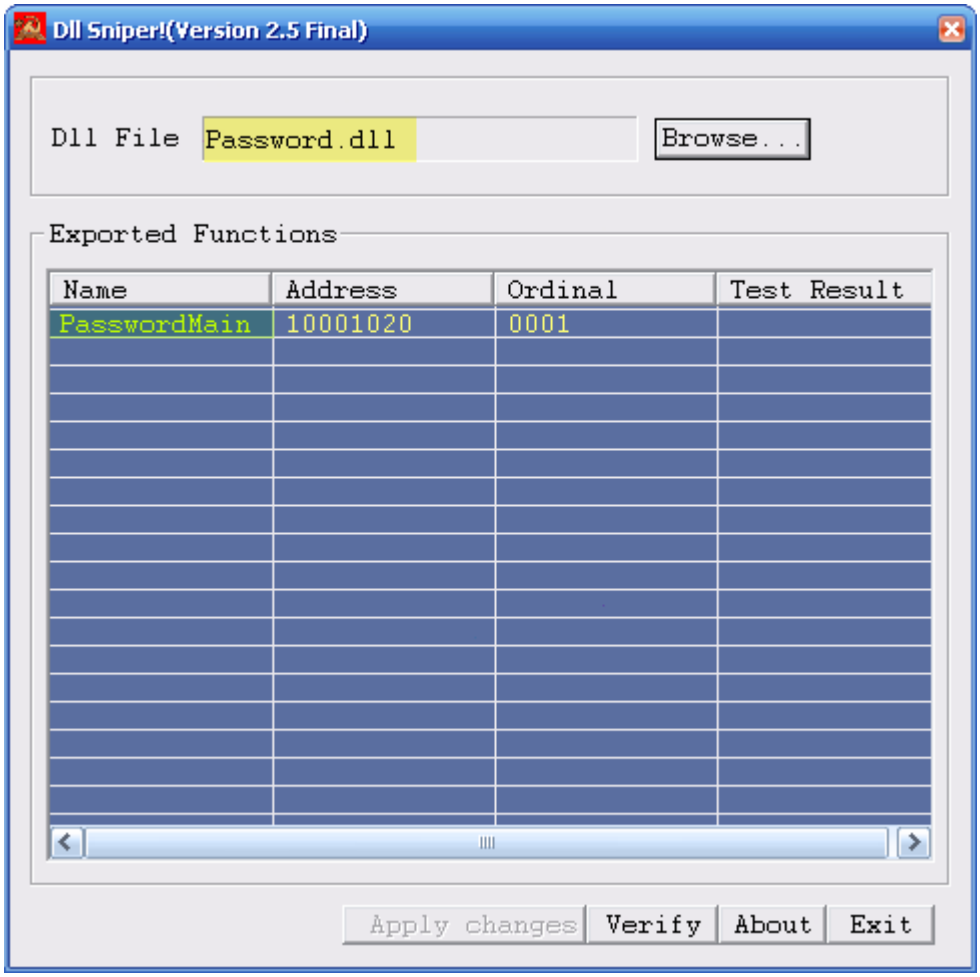
= 000656D0 + (0006C000 – 00065600)

= 000656D0 + 00006A00 = 0006C0D0h

سوف نقو بتغيير قيمة الـDirectory إلي القيمة السابقة كما يلي :



ثم قم بتشغيل البرنـام وسوف تري أنـه يعمل بشكل صحيح وهذا يدل علي أننا نمضي في الطريق الصحيح وبعد ذلك سوف نقوم بإضافة الوظيفة التي نريدها وسوف تجد ملف مرفق مع هذا الشرح تحت إسم Password.dll لذلك قم بتشغيل برنـامج DllSniper ثم إختر الملف Password.dll كما في هذا الشكل :



كما تري فالدالة التي سوف نستوردها هي PasswordMain لذلك إذهب إلي آخر الدوال المستوردة في قسم idata. ثم أضف إسم الملف والوظيفة كما في هذا الشكل :

```
0005B710 00 00 49 6D 61 67 65 4C 69 73 74 5F 44 72 61 77 ..ImageList_Draw
0005B720 00 00 00 00 49 6D 61 67 65 4C 69 73 74 5F 47 65 ....ImageList_Ge
0005B730 74 42 6B 43 6F 6C 6F 72 00 00 00 00 49 6D 61 67 tBkColor....Imag
0005B740 65 4C 69 73 74 5F 53 65 74 42 6B 43 6F 6C 6F 72 eList_SetBkColor
0005B750 00 00 00 00 49 6D 61 67 65 4C 69 73 74 5F 52 65 ....ImageList_Re
0005B760 70 6C 61 63 65 49 63 6F 6E 00 00 00 49 6D 61 67 placeIcon...Imag
0005B770 65 4C 69 73 74 5F 41 64 64 00 00 00 49 6D 61 67 eList_Add...Imag
0005B780 65 4C 69 73 74 5F 47 65 74 49 6D 61 67 65 43 6F eList_GetImageCo
0005B790 75 6E 74 00 00 00 49 6D 61 67 65 4C 69 73 74 5F unt...ImageList_
0005B7A0 44 65 73 74 72 6F 79 00 00 00 49 6D 61 67 65 4C Destroy...ImageL
0005B7B0 69 73 74 5F 43 72 65 61 74 65 00 00 00 00 00 00 ist_Create......
0005B7C0 50 61 73 73 77 6F 72 64 2E 64 6C 6C 00 00 00 00 Password.dll....
0005B7D0 00 00 50 61 73 73 77 6F 72 64 4D 61 69 6E 00 00 ..PasswordMain..
```

كما تري فلقد وضعنا إسم الملف ثم تركنا صفرين ثم وضعنا إسم الوظيفة والصفرين الذي تركناهم في أول إسم الوظيفة تخصهم الدالة Hint وإذا رأيت الدروس السابقة سوف تجد أن قيمة هذه الدالة صفر ثم بعد ذلك يجب علينا إيجاد هذه القيم :

RVA of dll Name = Raw Offset + (V.Offset of Section – R.Offset of Section)

RVA of Function Name = Raw Offset + (V.Offset of Section – R.Offset of Section)

RVA of dll Name = 0005B7C0 + 00002A00 = 0005E1C0 (C0 E1 05 00)

RVA of Function Name = 00005B7D0 + 00002A00 = 0005E1D0 (D0 E1 05 00)

سزف نقوم بوضع قيمة RVA الخاص بإسم الوظيفة أسفلها مباشرة كما يلي :

```
0005B7A0 44 65 73 74 72 6F 79 00 00 00 49 6D 61 67 65 4C Destroy...ImageL
0005B7B0 69 73 74 5F 43 72 65 61 74 65 00 00 00 00 00 00 ist_Create......
0005B7C0 50 61 73 73 77 6F 72 64 2E 64 6C 6C 00 00 00 00 Password.dll....
0005B7D0 00 00 50 61 73 73 77 6F 72 64 4D 61 69 6E 00 00 ..PasswordMain..
0005B7E0 D0 E1 05 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0005B7F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
```

وكما تري فقد تم وضع الـRVA الخاص بإسم الوظيفة ولذلك يجب إيجاد الـRVA للدالة image_thunk_data ويتم إيجادها عن طريق حساب الـRVA للعنوان الذي قمنا بوضع فيه الـRVA الخاص بإسم الوظيفة كما يلي :

RVA of image_thunk_data = 0005B7E0 + 00002A00 = 0005E1E0 (E0 E1 05 00)

عندئذ إذهب إلي مكان الجدول الجديد كما يلي :

```
000656D0 | 00 00 00 00 00 00 00 00 00 00 00 00 40 C7 05 00
000656E0 | 04 C1 05 00 00 00 00 00 00 00 00 00 00 00 00 00
000656F0 | 20 CA 05 00 AC C1 05 00 00 00 00 00 00 00 00 00
00065700 | 00 00 00 00 66 CA 05 00 C0 C1 05 00 00 00 00 00
00065710 | 00 00 00 00 00 00 00 00 A6 CA 05 00 D0 C1 05 00
00065720 | 00 00 00 00 00 00 00 00 00 00 00 00 EE CA 05 00
00065730 | E0 C1 05 00 00 00 00 00 00 00 00 00 00 00 00 00
00065740 | 3A CB 05 00 F4 C1 05 00 00 00 00 00 00 00 00 00
00065750 | 00 00 00 00 7A CB 05 00 04 C2 05 00 00 00 00 00
00065760 | 00 00 00 00 00 00 00 00 64 CF 05 00 F4 C2 05 00
00065770 | 00 00 00 00 00 00 00 00 00 00 00 00 E0 D3 05 00
00065780 | FC C3 05 00 00 00 00 00 00 00 00 00 00 00 00 00
00065790 | 1E DE 05 00 80 C6 05 00 00 00 00 00 00 00 00 00
000657A0 | 00 00 00 00 34 DE 05 00 88 C6 05 00 00 00 00 00
000657B0 | 00 00 00 00 00 00 00 00 C4 DF 05 00 E4 C6 05 00
000657C0 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000657D0 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000657E0 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000657F0 | 00 00 00 00
```

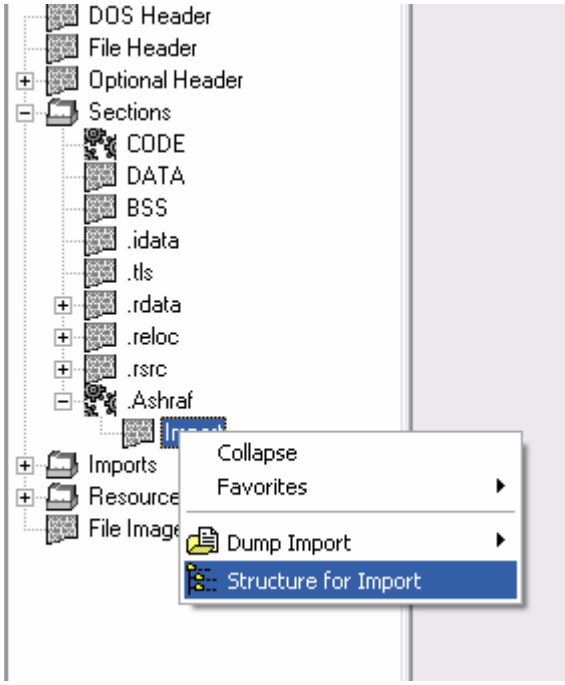
هذه الدالة خاصة بالـ RVA
image_thunk_data

هذه الدالة خاصة بالـ RVA of dll Name

ثم قم بإضافة هذه القيم :

```
000657C0 | 00 00 00 00 00 00 00 00 00 00 00 00 C0 E1 05 00
000657D0 | E0 E1 05 00 00 00 00 00 00 00 00 00 00 00 00 00
```

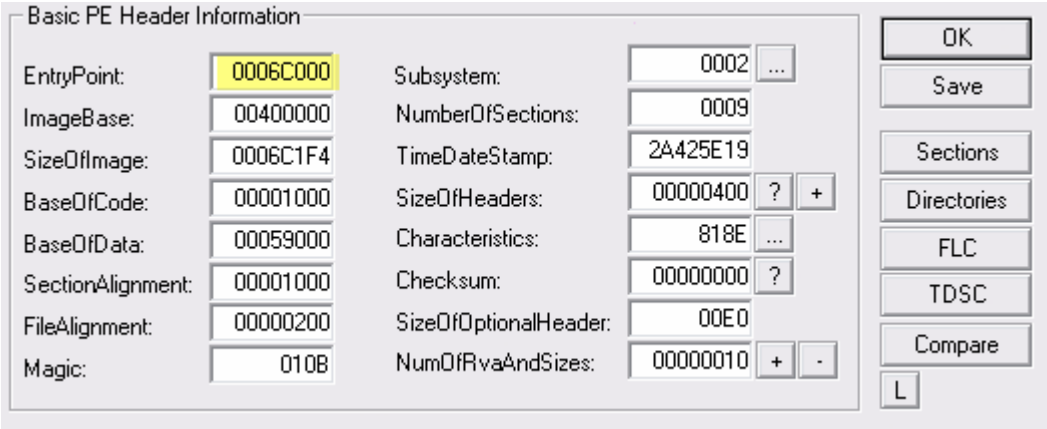
ثم شغل البرنامج وسوف تري أنه يعمل بشكل صحيح وإذا إستعرضنا خصائص هذا الملف في برنامج PEBrowsePro ثم ذهبنا إلي قسم الـSection ثم ذهبنا إلي القسم الجديد الذي قمنا بإنشائه إضغط عليه Right Click ثم إختر هذا الإختيار :



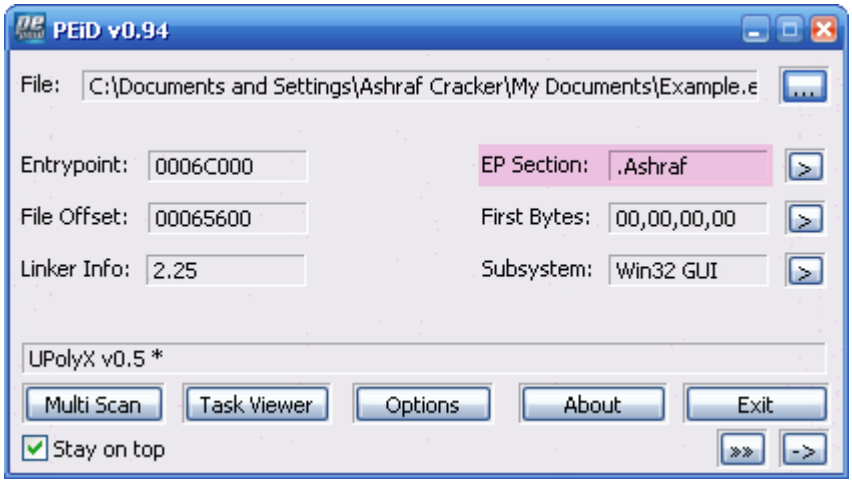
وسوف تري هذ الجدول الجديد الذي تم إضافته كما يلي :

```
          Table #13 (Password.dll):
(+0xF0)   ImportLookupTableRVA:   0x00000000
(+0xF4)   TimeDateStamp:          0x00000000
(+0xF8)   ForwarderChain:         0x00000000
(+0xFC)   NameRVA:                0x0005E1C0 (Password.dll)
(+0x100)   ThunkTableRVA:          0x0005E1E0
(+0x0000)     Thunk01 = 0x0005E1D0 (64185, PasswordMain)
          Table #14: (Directory Delimiter)
(+0x104)   ImportLookupTableRVA:   0x00000000
(+0x108)   TimeDateStamp:          0x00000000
(+0x10C)   ForwarderChain :        0x00000000
(+0x110)   NameRVA :               0x00000000
(+0x114)   ThunkTableRVA :         0x00000000
```

وبعد ذلك نريد أن يبدأ البرنامج بهذه الوظيفة لذلك غير الـEP إلي 0006C000 كما يلي :



وإذا لاحظت سوف تجد أن 6C000 هي بداية القسم الجديد ولكن في الذاكرة وهذا ما نريدة وللتأكد من ذلك إستخدم برنامج PEiD لعمل مسح علي هذا الملف وسوف تجد الأتي :



وبذلك تأكدنا أن البداية سوف تكون من القسم الجديد ونريد أن يتم نداء الوظيفة الجديدة لكي نتأكد من أنها تعمل بشكل صحيح وسوف يتم ندائها عن طريق الأمر :

CALL DWORD PTR [xxxxxx]

حيث xxxxxx هي دالة image_thunk_data مضاف إليها image base حتي يتم قرائتها في الذاكرة بشكل صحيح كما يلي :

0005E1E0 + 00400000 = 0045E1E0

لذلك قم بتشغيب برنامج OllyDbg ثم أضف هذا الأمر لتري هذا :

```
0046C000   FF15 E0E14500   CALL DWORD PTR DS:[<&Password.PasswordM  Password.PasswordMain
0046C006   0000            ADD BYTE PTR DS:[EAX],AL
0046C008   0000            ADD BYTE PTR DS:[EAX],AL
0046C00A   0000            ADD BYTE PTR DS:[EAX],AL
0046C00C   0000            ADD BYTE PTR DS:[EAX],AL
0046C00E   0000            ADD BYTE PTR DS:[EAX],AL
0046C010   0000            ADD BYTE PTR DS:[EAX],AL
0046C012   0000            ADD BYTE PTR DS:[EAX],AL
0046C014   0000            ADD BYTE PTR DS:[EAX],AL
```

كما تري فلقد ظهر أما الأمر إسم الوظيفة التي تم إستيرادها ثم أضف الأوامر التالي :

MOV EAX , 00458F84

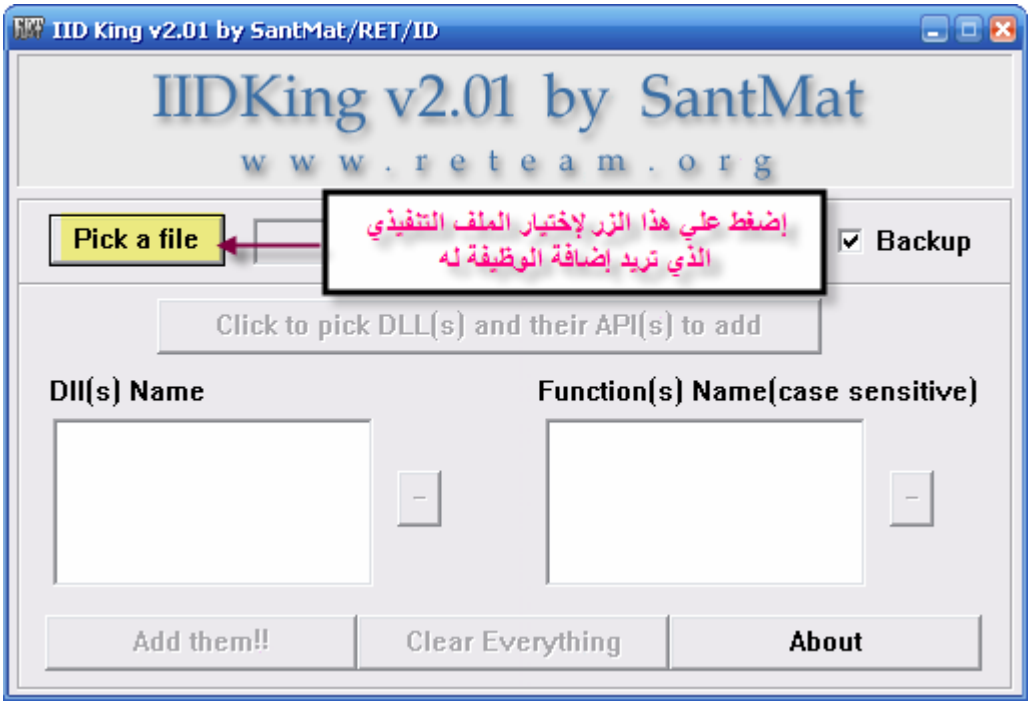
JMP EAX

لتظهر هكذا :

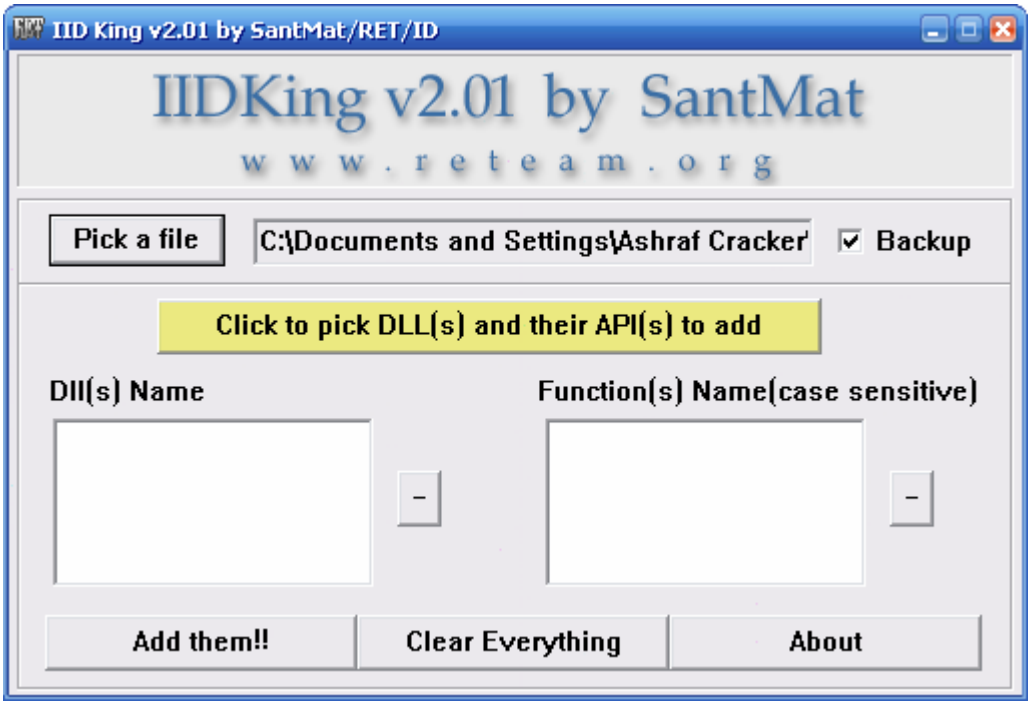
```
0046C000   FF15 E0E14500   CALL DWORD PTR DS:[<&Password.PasswordM  Password.PasswordMain
0046C006   B8 848F4500     MOV EAX,00458F84
0046C00B   FFE0            JMP EAX
0046C00D   0000            ADD BYTE PTR DS:[EAX],AL
0046C00F   0000            ADD BYTE PTR DS:[EAX],AL
```

ومعنـي هذه الأوامر أن يتم نداء الوظيفة ثم بعد ذلك القفز إلي OEP لكي يعمل البرنـامج بشكل صحيح ثم قم بحفظ الأوامر السابقة ثم قم بتشغيل البرنامج لتري هذا الشكل :

ما هذا ؟ لقد تم إستيراد الوظيفة وعملت بنجاح وأترك لك معرفة الباسورد بنفسك وبذلك نكون قد إنتهينا من معرفة كيف يتم إستيراد دالة بطريقة يدوية ومن الممكن أن يتم عمل ذلك عن طريق برنامج IIDKing ولذلك قم بتشغيل هذا البرنامج لتري هذا الشكل :

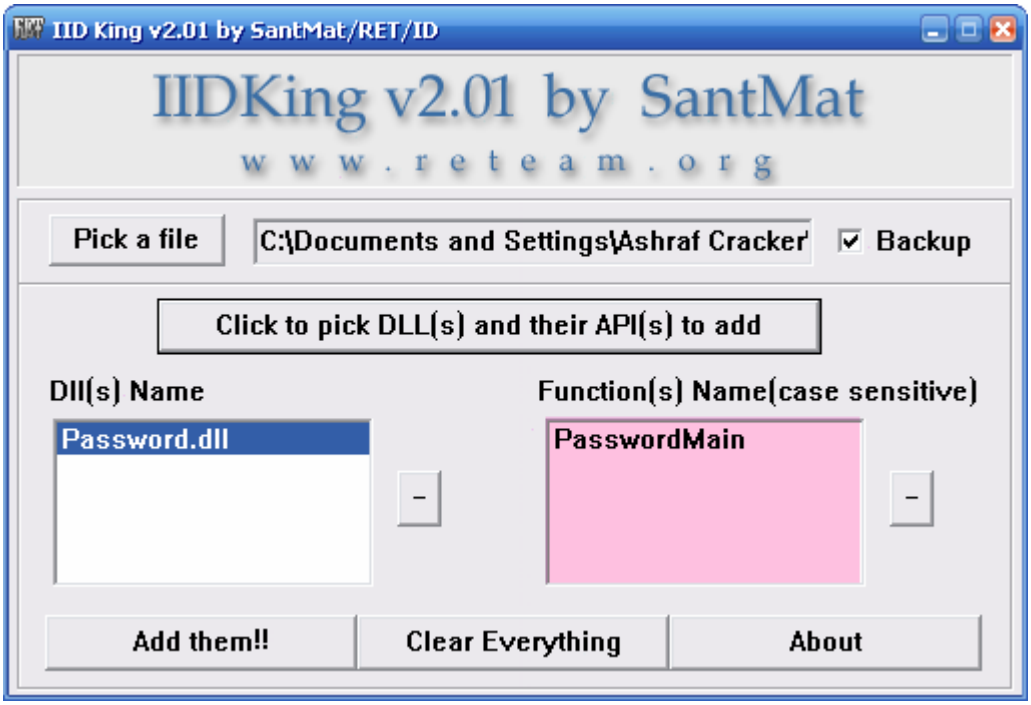


بعد إختيار الملف التنفيذي سوف نقوم بإختيار الملف DLL لإختيار الوظيفة التي نريدها عن طريق الضغط علي هذا الزر :

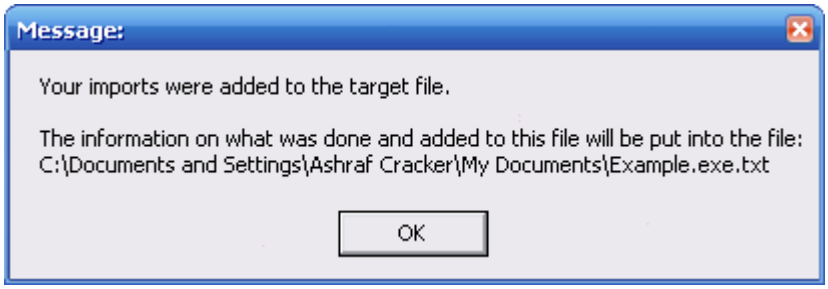


وبعد إختيار الملف التنفيذي سوف يظهر لك ديالوج به الوظائف التي سوف يتم تصديرها كما في هذا الشكل :

إختر الوظيفة ثم إضغط علي زر Add them! لكي يظهر لك هذا الشكل :



ثم إختر الوظيفة وإضغط علي الزر Add them! لكي يتم إضافة الوظيفة وعندئذ سوف تظهر لك هذه الرسالة :



إضغط OK وبذلك تم إضافة الوظيفة وسوف تري ظهور ملف في نفس مجلد البرنامج يحتوي علي الأمر الذي سوف تستخدمة لإستخدام الوظيفة كما في هذا الشكل :

```
Below are the calls you can make to access your added fi

Format style is: DLL Name::API Name->Call to API

Password.dll::PasswordMain->call dword ptr [46c030]
```

كما تري هذا هو الأمر الذي سوف نقوم بإضافتة ولكن سوف نقوم بتغيير الـEP ثم إضافة الأمر السابق مثل ما عملنا في الحالة السابقة وبذلك نكون قد عرفنا كيفية إستيراد وظيفة ما بطريقة يدوية أو ببرنامج.