

# الفصل الثالث

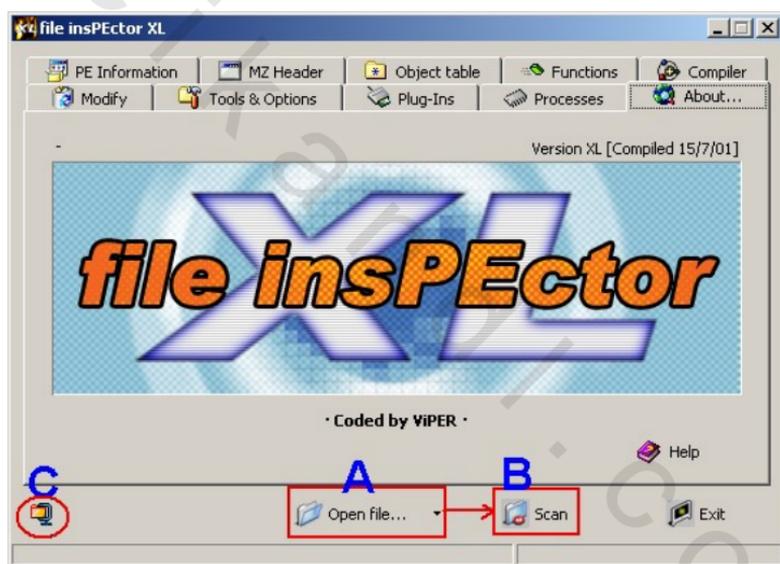


برامج فحص PE

## برامج فحص PE

برنامج File Inspector:

هذا البرنامج يقوم بإظهار العديد من المعلومات الهامة عن الملف التنفيذي المراد كسره منها اللغة المستخدمة في البرنامج وعنوان تحميل PE والإحجام المختلفة للملف مثل جزء البيانات data وجزء header ويبين أيضا الدوال التي يستخدمها البرنامج أو حتى التي يقوم بتعريفها تظهر الشاشة الافتتاحية كما بالشكل التالي:

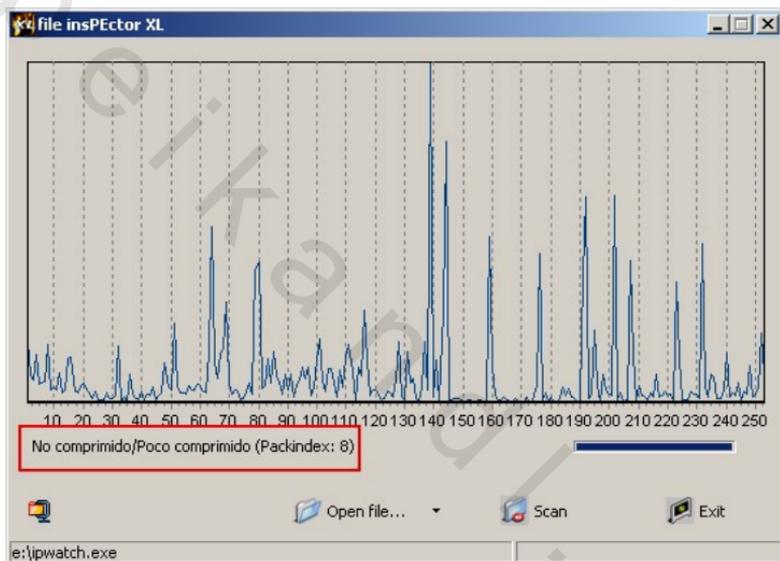


ولكي تقوم باستخدام البرنامج اتبع الخطوات التالية:

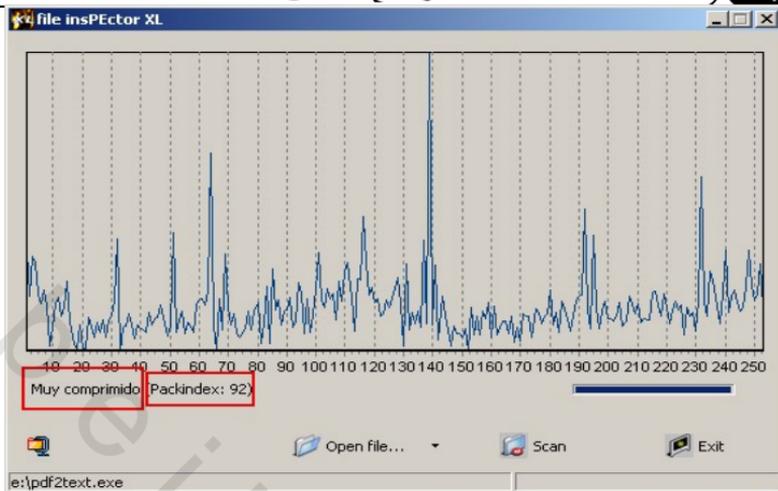
1. الأمر Open وهو يستخدم لاختيار الملف المراد فتحه ومعرفة معلومات عنه.
2. الأمر Scan حتى يبدأ البرنامج فعليا في فحص الملف وإظهار جميع المعلومات عنه.

3. أحيانا يكون الملف مضغوط فيستطيع هذا المفتاح إظهار رسالة إذا كان الملف مضغوط أي يحتوى الملف التنفيذي الفعلي.

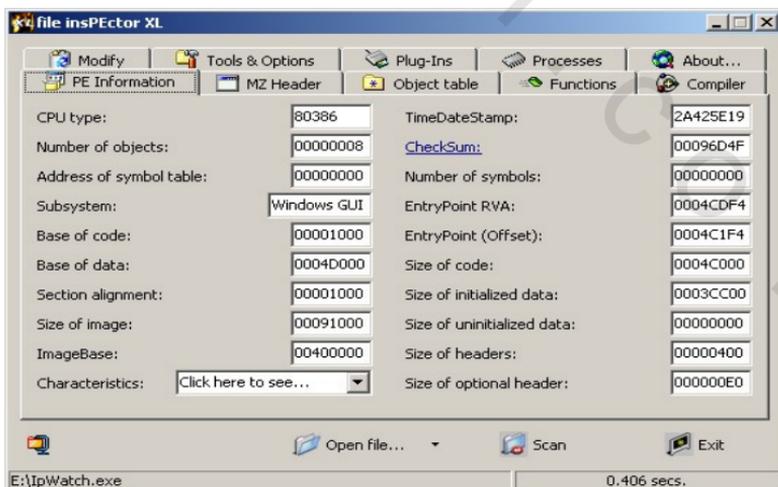
إذا ضغطت الآن على المفتاح C سترى الشكل التالي (بعد تنفيذ الأمر open بالطبع)



يقوم البرنامج بإظهار معدلات الضغط ورسالة بسيطة تظهر إذا كان الملف مضغوط أم لا ومن الشكل السابق يظهر البرنامج أن هذا الملف غير مضغوط أما إذا اخترنا ملف مضغوط فسترى الشكل التالي:

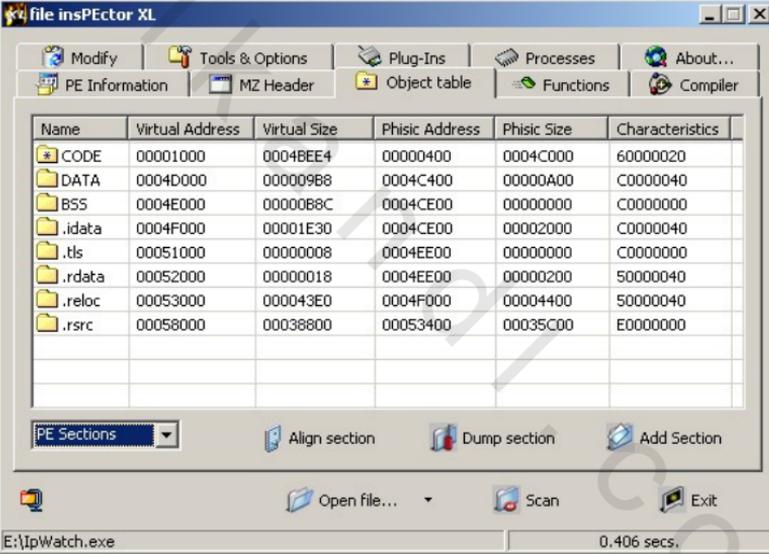


يظهر البرنامج أن الملف مضغوط ضغطا عاليا بنسبة 92 في المائة إذا ضغطت مره أخرى على المفتاح إظهار الضغط ستقوم بالرجوع إلى الشاشة الرئيسية قم الآن بالضغط على مفتاح scan إذا لم تقم بذلك فيقوم البرنامج بإظهار أول صفحه وهي صفحة معلومات نقطة الدخول للبرنامج أو PE info كما بالشكل:



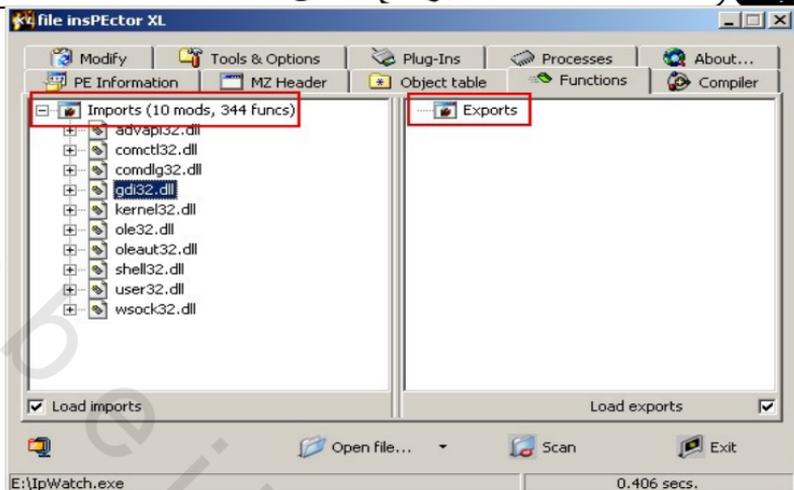
يظهر البرنامج في هذه الصفحة رقم checksum الذي يستخدم للتأكد من الحجم الصحيح للملف وانه لم يتم تغييره وعنوان نقطة الدخول EP offset وغيره من المعلومات المفيدة.

صفحة Object table تحتوي على معلومات عن الأجزاء التي يتجزأ منها ويعطى إمكانية أن يقوم بتخزين محتويات جزء بعينه فمثلا الجزء data الذي يحتوي جميع بيانات الملف وهو موجود دائما بالملفات التنفيذية كما رأينا في الجزء السابق (راجع باب البرمجة بالاسمبلى).



Name	Virtual Address	Virtual Size	Phisic Address	Phisic Size	Characteristics
CODE	00001000	0004BEE4	00000400	0004C000	60000020
DATA	0004D000	000009B8	0004C400	00000A00	C0000040
BSS	0004E000	00000B8C	0004CE00	00000000	C0000000
.idata	0004F000	00001E30	0004CE00	00002000	C0000040
.tls	00051000	00000008	0004EE00	00000000	C0000000
.rdata	00052000	00000018	0004EE00	00000200	50000040
.reloc	00053000	000043E0	0004F000	00004400	50000040
.rsrc	00058000	00038800	00053400	00035C00	E0000000

الصفحة التي تليها وهي صفحة Functions وتحتوى دوال API التي يقوم الملف باستخدامها سواء كانت دوال API الخاصة بالويندوز أو دوال خاصة بالملف

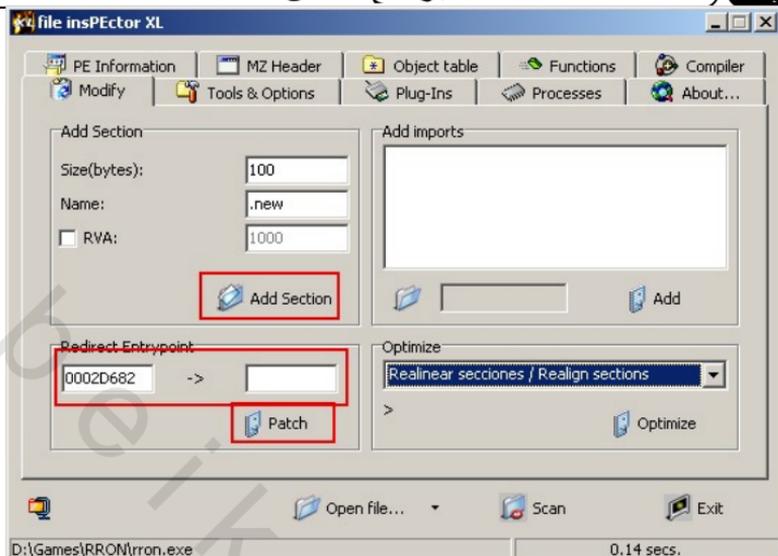


كما يظهر البرنامج أيضا ملف dll أو المكتبة الخاصة بالويندوز المسئولة عن تنفيذ دالة API مثل الملف GDI32.dll مثلا نجد به دوال التعامل مع رسومات الملف.

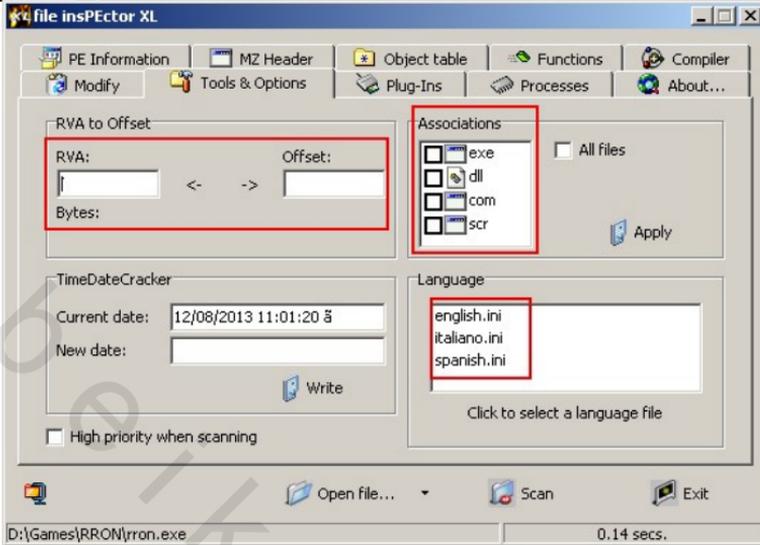
الصفحة التي تلي ذلك هي صفحة عن المعالج الذي قام بترجمة وإنتاج الملف التنفيذي وكما يظهر الشكل التالي فإن الملف مكتوب بلغة فيجوال سي بلس بلس



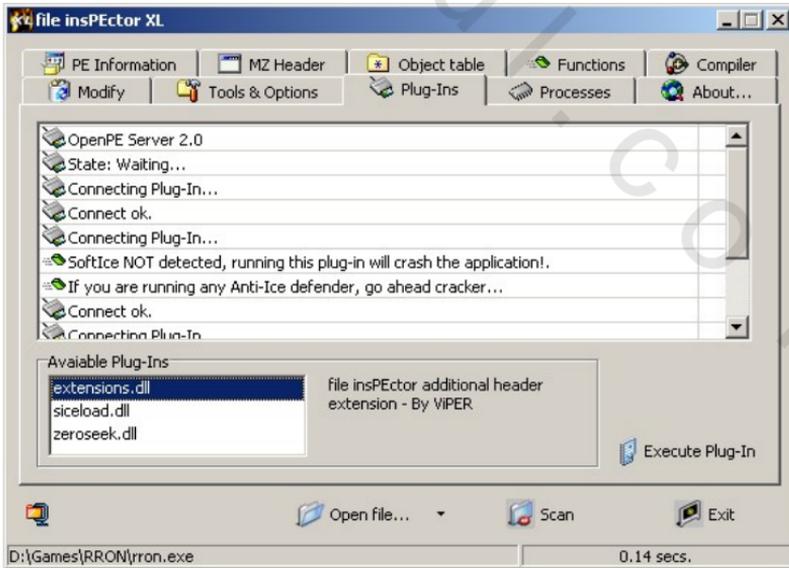
الصفحة التالية وهي صفحة Modify وتمكننا من التعديل في الملف فمثلا إذا أردنا أن نضيف جزء جديد إلى الملف وبه بيانات خاصة بنا نريد أن يقوم الملف بتحميلها قبل أن يستمر في تنفيذ باقي التعليمات الخاصة به فيمكن ذلك عن طريق تحديد EP الجديدة والضغط على مفتاح patch كما يمكنك أن ترى من الشكل:



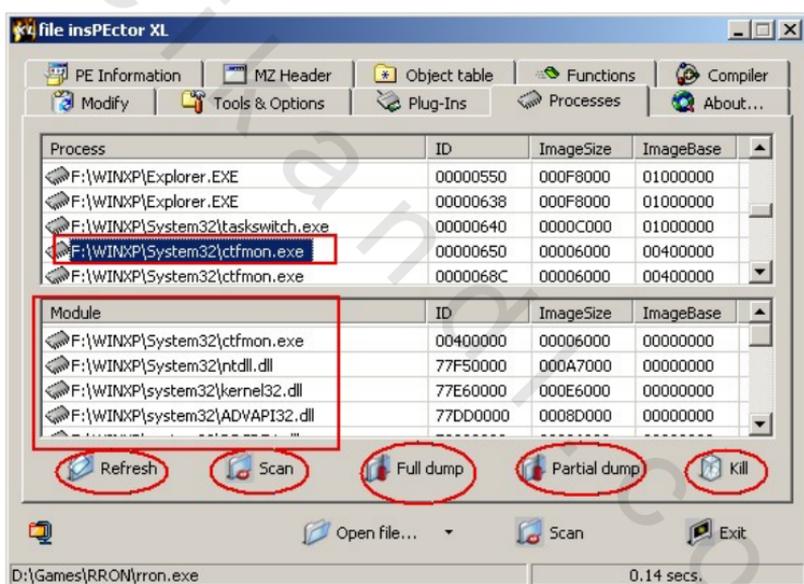
الصفحة التالية هي صفة الأدوات والاختيارات ويمكن عن طريقها تحويل عنوان من نوع RVA أي عنوان تخيلي إلى عنوان حقيقي وهو offset كما تمكنا هذه الصفحة من تثبيت البرنامج مع الملفات التنفيذية بأنواعها عن طريق الضغط على المفتاح الأيمن من الماوس كما يمكننا البرنامج من تغيير لغة واجهة البرنامج إلى الاسبانية أو الايطالية إذا لم تكن تعرف الانجليزية 😊



و هناك صفحة أخرى مخصصة للتعامل مع plugins والتي تزيد من  
 الإمكانيات العادية للبرنامج مثل إدراج خاصية التحميل مع برنامج soft-  
 ice خصوصا للبرامج المشفرة أو المضغوطة كما يمكننا من البحث داخل أي  
 جزء من أجزاء الملف وتغيير البيانات به



أما الصفحة process فتحتوي على جميع البرامج المحملة بالذاكرة بما فيها برامج services ويمكننا البرنامج من تسجيل محتويات هذه البرامج على الهارد ديسك باختيار أوامر dump أو إغلاقها إذا لزم الأمر بالمفتاح Kill و يحتوي الجزء العلوي من هذه الشاشة على البرامج المحملة بالذاكرة فإذا اخترت واحدا منها يظهر في الجزء السفلي الملفات أو المكتبات التي يعتمد عليها البرنامج المختار كما يمكن عمل scan لبرنامج محمل بالذاكرة بدلا من اختياره بالأمر .open.



فائدة أوامر ال dump هنا هو إذا كان الملف مضغوطا يمكننا أن نقوم باستخراج الملف الحقيقي التنفيذي عن طريق هذه الأوامر كما يمكن تنفيذ ذلك بالعديد من البرامج المستقلة مثل procdump .

من البرامج الخفيفة والعملية جدا لفك حماية البرامج المشفرة فيتم وضع مقطع وهمي مشفر بحيث يكون هذا المقطع هو EP وبعد تحميل هذا المقطع وليكن مثلا rData. يقوم بفحص CD إذا وجدها يتم النداء على المقطع الأصلي للملف التنفيذي وليكن مثلا text. ويعمل البرنامج بصورة طبيعيه أما إذا لم يجد CD فينادى المقطع على كود الخروج من البرنامج ولا يتم الذهاب أبدا إلى المقطع text.

أما الملفات المضغوطة فعادة يتم فيها وضع ملف exe الحقيقي المسئول عن تشغيل البرنامج داخل ملف تنفيذي آخر بحيث يستحيل على الكراكر فك الملف ببرامج إعادة الهندسة لذلك يجب أولا استخراج الملف الحقيقي للبرنامج مثل حماية SecurRom , Safe Disk وغيرها.  
الصورة الرئيسية للبرنامج:

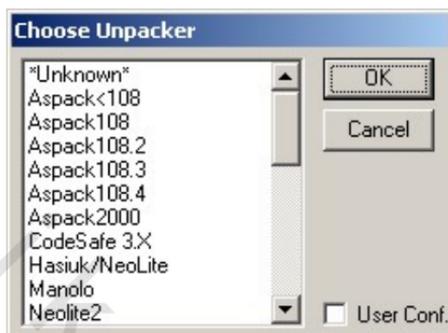
Task	PID	Address	Size	Owner
f:\winxp\system32\taskswitch.exe	0000063C	01000000	0000C000	00000000
f:\winxp\system32\ctfdmon.exe	00000644	00400000	00006000	00000000
f:\program files\kaspersky lab\kaspers...	00000684	00400000	00364000	00000000
f:\winxp\system32\nvsvc32.exe	000006E8	00400000	00021000	00000000
f:\program files\virtual cd v4\system\v...	0000073C	00400000	00008000	00000000
f:\progra~1\flashget\flashget.exe	00000538	00400000	00147000	00000000
f:\brooram files\internet explorer\ie.xpl...	0000069C	00400000	00019000	00000000

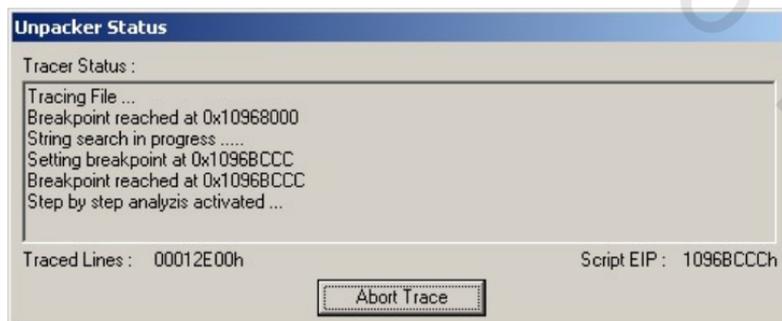
Module	MID	Address	Size
f:\program files\virtual cd v4\system\v...	00403D0F	00400000	00008000
f:\winxp\system32\ntdll.dll	00000000	77F50000	000A7000
f:\winxp\system32\kernel32.dll	77E7AE60	77E60000	000E6000
f:\winxp\system32\mfic42.dll	73DD5D23	73DD0000	000F2000
f:\winxp\system32\msvcrt.dll	77C1E94F	77C10000	00053000
f:\winxp\system32\gdi32.dll	00000000	7E090000	00041000
f:\winxp\svstem32\user32.dll	77D4C6F2	77D40000	00086000

من الشكل السابق يتضح أن البرنامج يقوم بمسح الذاكرة من جميع البرامج المحملة بها ويقوم بعرض كل برنامج في القائمة task والملفات التي يقوم باستخدامها في القائمة module .

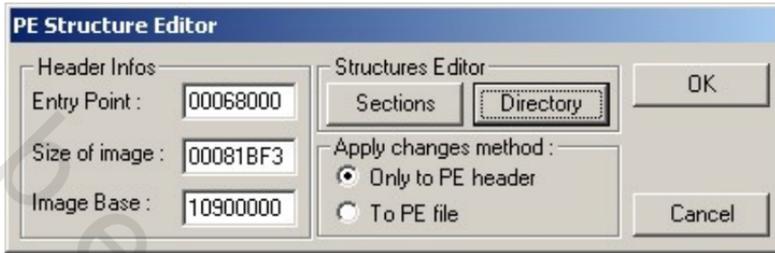
يمكننا البرنامج من التعامل مع الملفات التنفيذية بعدة طرق فالمفتاح الأول Unpack يعطينا إمكانية استخراج الملف التنفيذي الحقيقي وعند الضغط عليه يقوم البرنامج بإظهار ديلوج لاختيار نوع التشفير المستخدم :



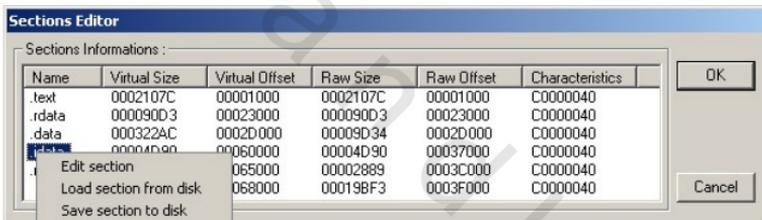
ومن أشهر الطرق التشفير استخدام النوع aspack ونرى أن البرنامج يحتوى على العديد من هذه الطرق قم الآن باختيار احد التشفيرات واضغط ok فنرى ديلوج لاختيار الملف المراد فكه قم باختيار الملف التنفيذي ثم تظهر رسالة تقييد الضغط على مفتاح ok إذا تم تحميل البرنامج بصورة صحيحة ثم يبدأ البرنامج في فحص الملف واختبار العناوين المخزنة داخل الملف فإذا وجد العنوان الرئيسي للملف PE فيبدأ باستخراجه .



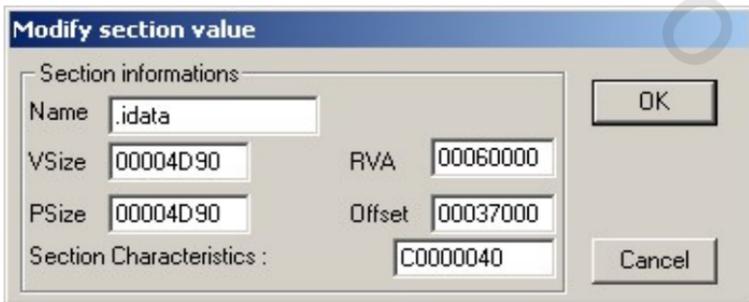
و عن طريق مفتاح PE Editor يمكننا تعديل نقطة الدخول الخاصة PE الخاصة بالملف إلى نقطه دخول أخرى



كما يمكن تعديل مقاطع الملف عن طريق المفتاح Sections



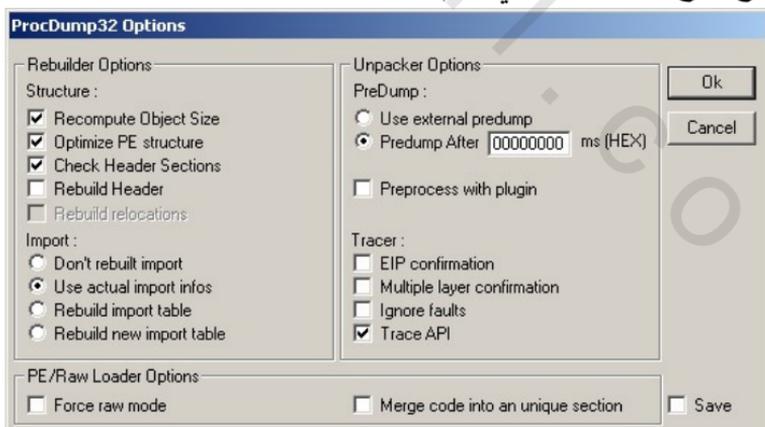
و يمكننا عن طريق قائمة الأوامر المختصرة تعديل مقطع في الملف أو إزالته تماما



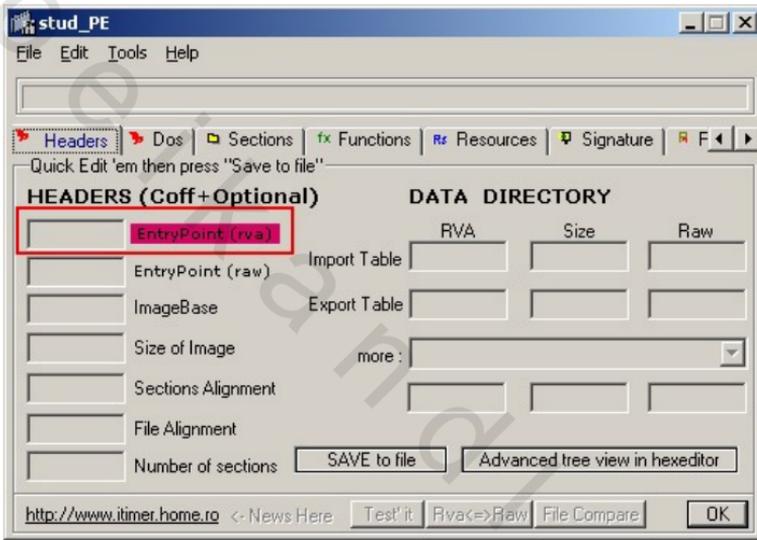
و إذا حدث خطأ يمكنك إعادة بناء الملف عن طريق المفتاح PE Rebuild .  
 أما المفتاح Bhrama Server فيقوم بتحميل برنامج صغير في الذاكرة  
 لاستخدامه مع اسطوانات الألعاب المحمية بطريقة securRom



المفتاح الأخير Option يعطينا إمكانية تغيير طريقة التحليل التي يتبعها  
 البرنامج ونوع الفحوصات التي يقوم بها .

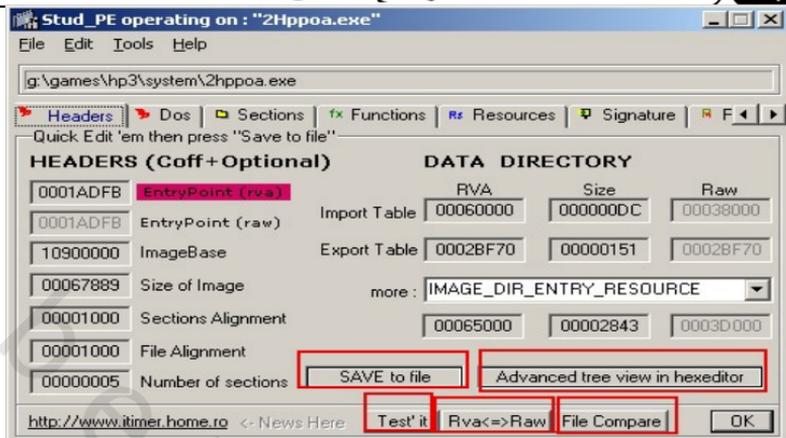


هذا البرنامج الصغير متميز جدا لاحتوائه على عدة إمكانيات توجد كل منها في برنامج منفصل بحد ذاته منها معرفة اللغة المستخدمة في البرنامج ومعرفة نوع الضغط أو التشفير الذي قد يكون مستخدم وعند فتح هذا البرنامج نرى الشاشة الافتتاحية التالية:

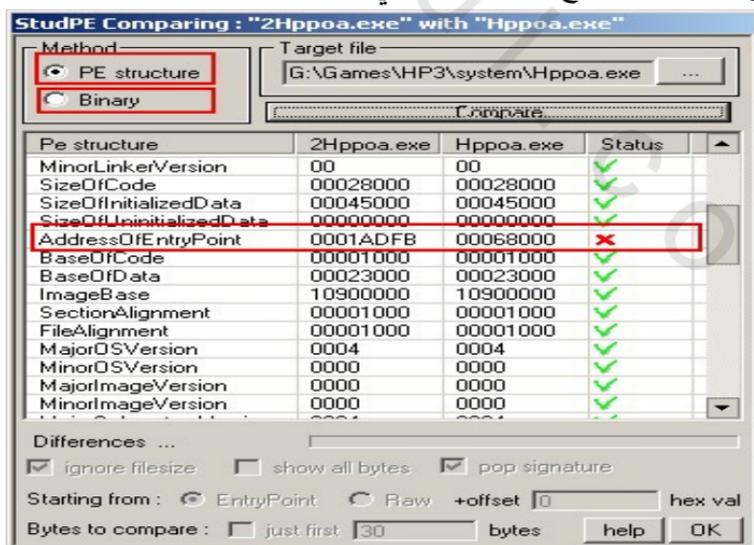


توضح أول صفحة من البرنامج بيانات مختلفة عن PE وإذا اخترنا الأمر open من القائمة File يمكنك أن ترى العناوين المختلفة الأخرى وعدد المقاطع في الملف

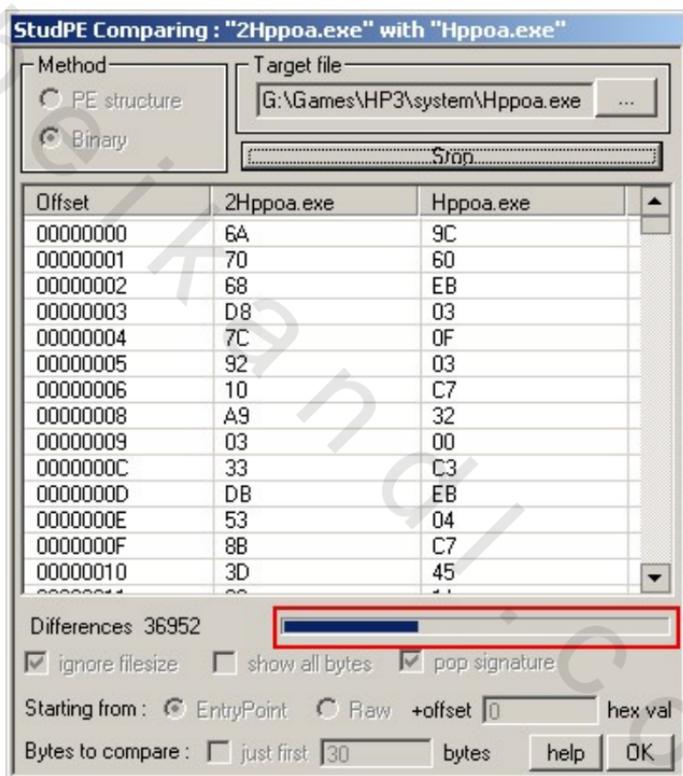
## برامج فحص PE



و كما ترى فمثلا هذا الملف مكون من 5 مقاطع ونقطة الدخول هي 1ADFB ويعطينا البرنامج الإمكانية للتحويل من العنوان التخيلي إلى العنوان الحقيقي عن طريق المفتاح Raw <-> Rva كما يمكنك أيضا من مقارنة ملف بآخر مقارنة على أساس هيكل PE أو على أساس ثنائي عن طريق المفتاح File Compare كما يتضح من الشكل التالي

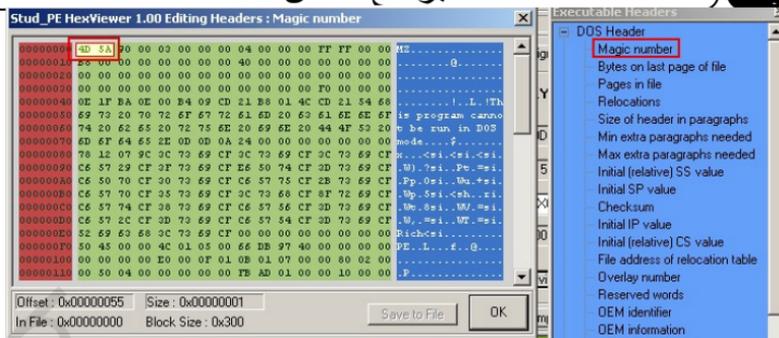


وكما ذكرنا يمكنك اختيار نوع المقارنة والضغط على المفتاح compare ويمكنك أن تشاهد علامة X أما التحاليل المختلفة مثل تغيير نقطة الدخول للبرنامج كما يتضح من الشكل السابق  
أما المقارنة الثنائية فهي تأخذ بعض الوقت ولكنها أدق في التفاصيل .

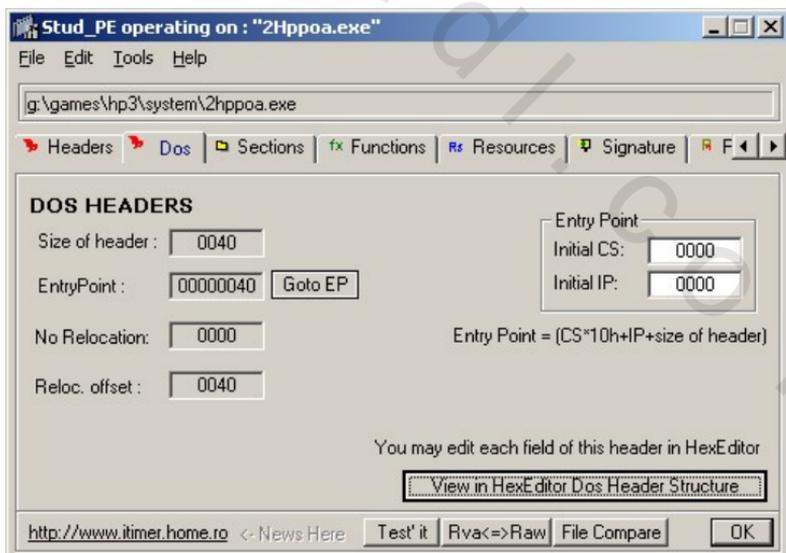


أما المفتاح Advanced tree view فيمكنك من مشاهدة الملف في صورة النظام السداسي عشر بجانب شجره توضح أهم عناوين الملف مثل عنوان بداية احد مقاطع الملف

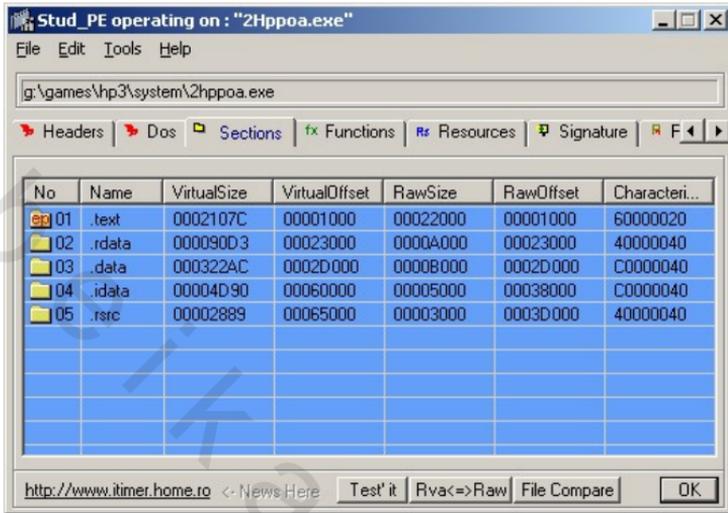
## برامج فحص PE



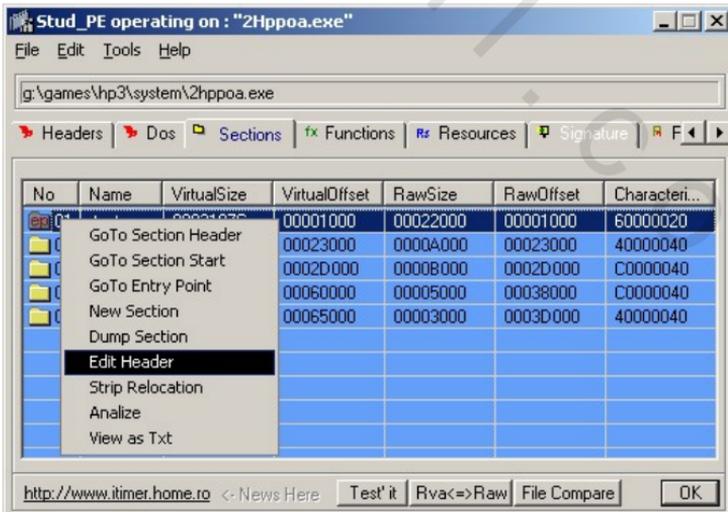
و يمكنك تغيير أي بيانات في هذه الشاشة أو في الشاشة الافتتاحية وحفظ التغييرات فوراً عن طريق المفتاح **Save to file** الصفحة التالية هي **Dos** وتحتوي على معلومات هامة عن بداية المقطع الذي يقرؤه الدوس ويمكنك أيضاً مشاهدة هذه المقاطع بالصورة السادسة عشر عن طريق مفتاح **view in hex editor dos header structure**



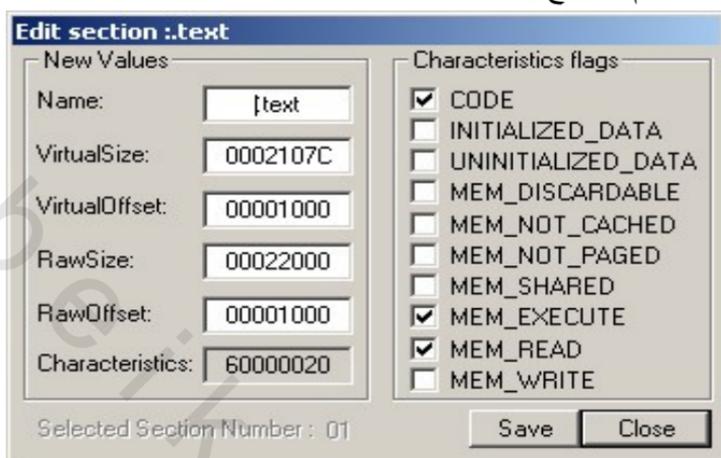
أما الصفحة sections فتحتوى على المقاطع الخاصة بالملف وتحديد أي مقطع هو ep أو نقطة البداية.



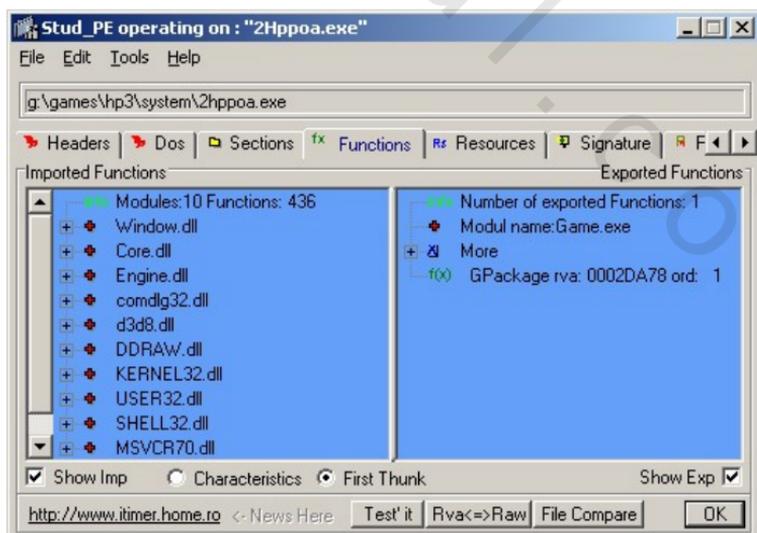
ويمكنك بالطبع اختيار احد المقاطع والتعديل فيها عن طريق أوامر القوائم المختصرة.



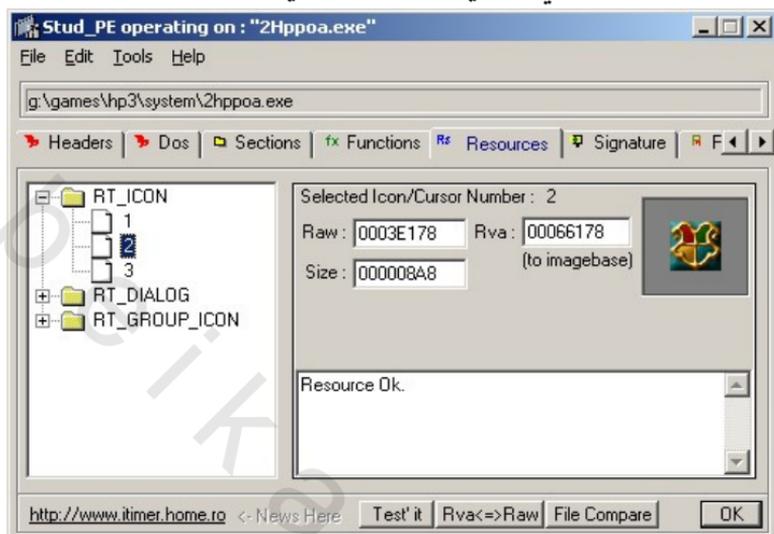
فمثلا إذا اخترت الأمر Edit Header يمكنك تغيير بايتات الخاصة برأس الملف مثل اسم المقطع والعلامات الخاصة به



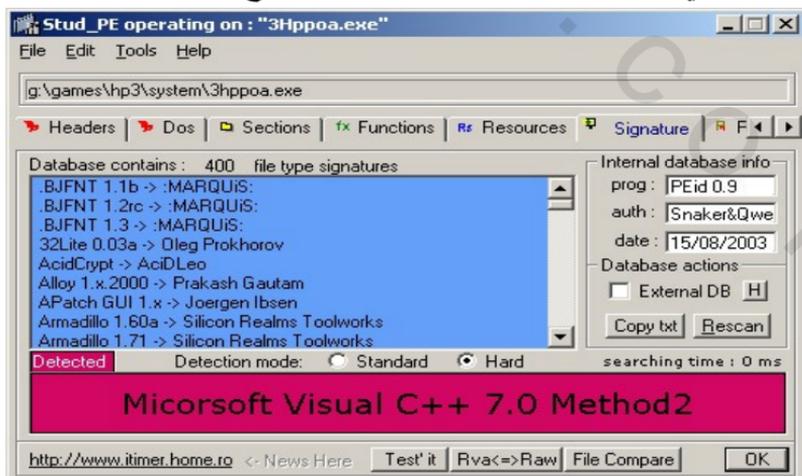
الصفحة التالية functions وهي الصفحة المعتادة التي رأيناها سابقا في العديد من البرامج وتحتوى هنا أيضا على الدوال المستخدمة في البرنامج



الصفحة resources تمكننا من رؤية المصادر التي يستخدمها البرنامج مثل  
الديالوجات والايقونات التي يحتويها الملف التنفيذي .

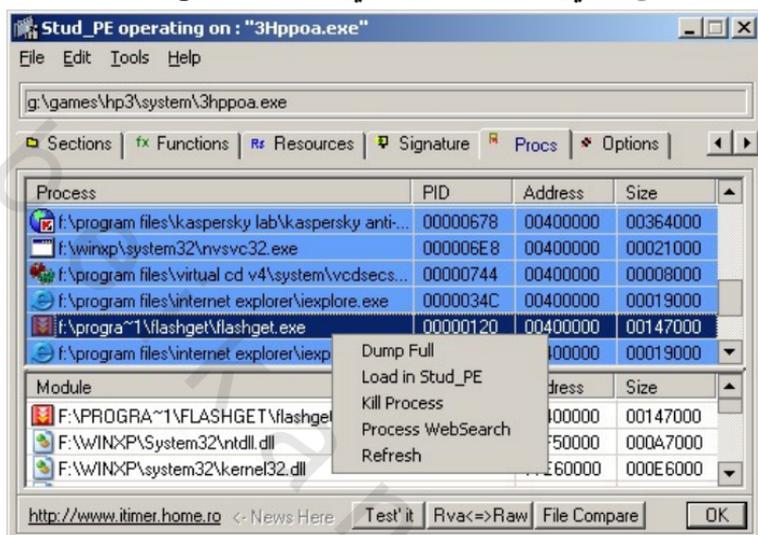


الصفحة التالية تبين اللغة المستخدمة في البرنامج أو نوع التشفير المستخدم إذا  
كان الملف مشفرا ويستخدم البرنامج الأداة Peid وملفات متعددة للتعرف على  
اللغة عن طريق وجود بيانات محده كل منها تحدد نوع اللغة المستخدمة

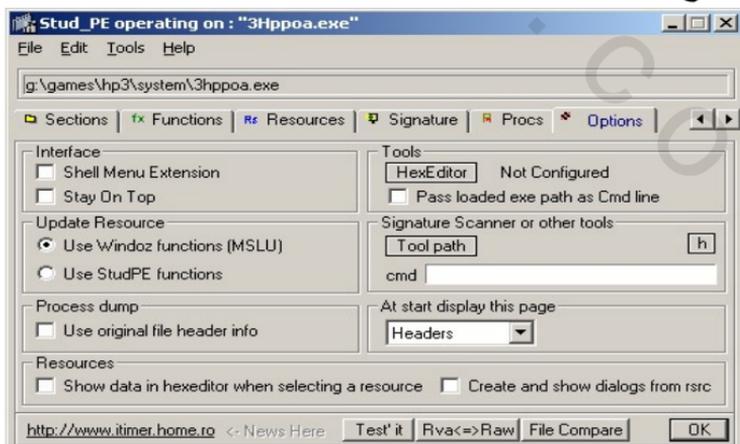


## برامج فحص PE

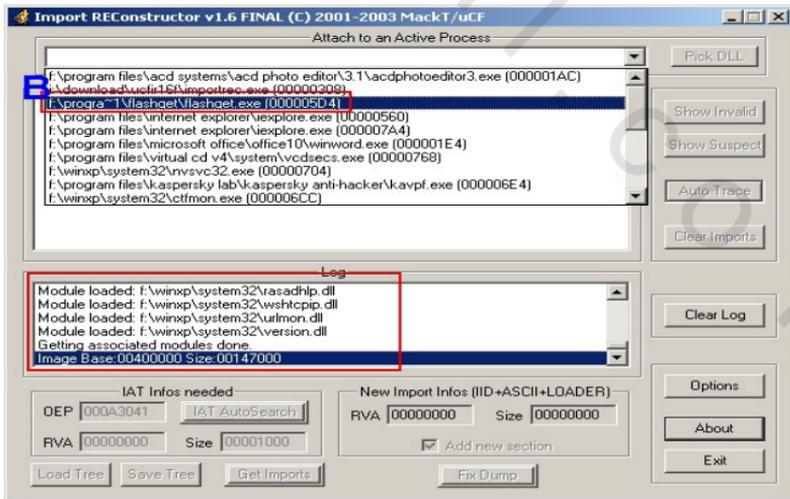
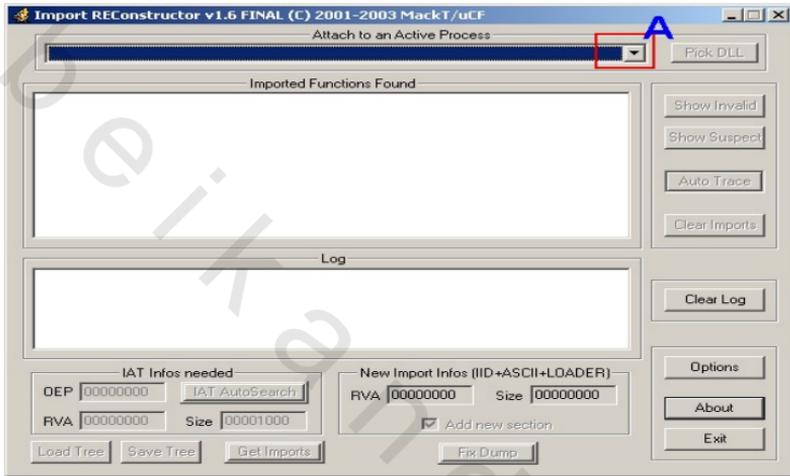
الصفحة التالية أيضا معروفه وهي صفحة Procs وتحتوى على جميع البرامج المحملة بالذاكرة ويمكنك أيضا تحميل أي برنامج بالذاكرة ببرنامج Stud\_PE أو إنهاؤه أو حتى تخزين محتوياته عن طريق الأمر .dump full



والصفحة الأخيرة وهي تحتوى على اختيارات البرنامج مثل إلحاق البرامج التنفيذية بهذا البرنامج أو اختيار محرر سداسي عشر خارجي غير المستخدم مع البرنامج



عند إعادة بناء الملفات والتعديل في هيكل PE قد يصبح جدول الدوال التي يستخدمها البرنامج فاسد بسبب التشفير لذلك يمكننا استخدام هذا البرنامج الذي وظيفته هو إعادة بناء جدول الدوال والشاشة الافتتاحية للبرنامج يمكننا من اختيار أي برنامج محمل حاليا بالذاكرة كما يتضح بالشكل التالي:



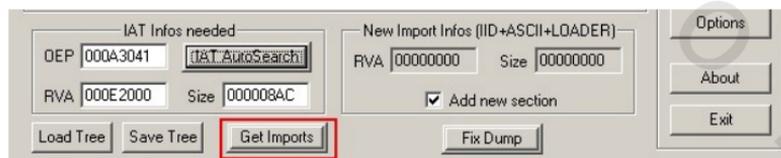
الخطوة التالية هي الضغط على المفتاح IAT Autosearch الذي يستطيع البحث عن العنوان المخصص للنداء على الدوال الخارجية كما بالشكل:



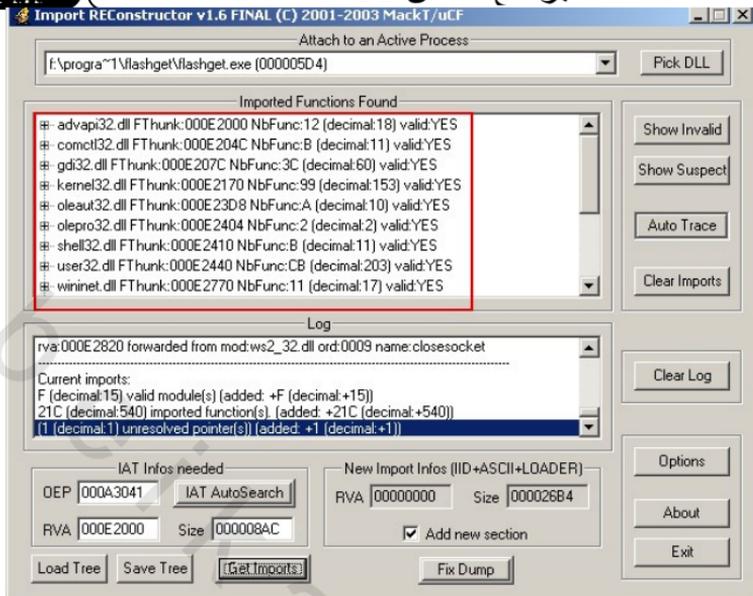
ستجد البرنامج اظهر رسالة تفيد احتمال وجود الدالة بعنوان معين



و يجب بعد ذلك الضغط على المفتاح Get Imports للحصول على الدوال المستخدمة بالملف المختار



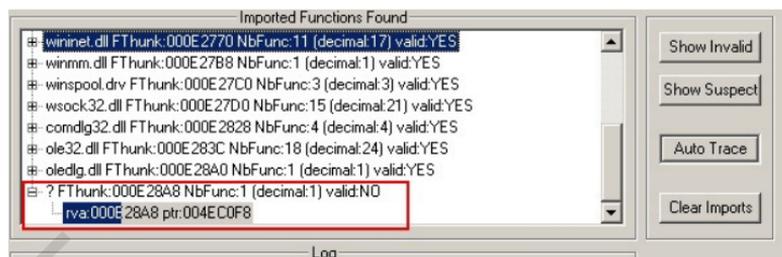
و تكون النتيجة هي إظهار الدوال المستخدمة في الشاشة الرئيسية كما يلي:



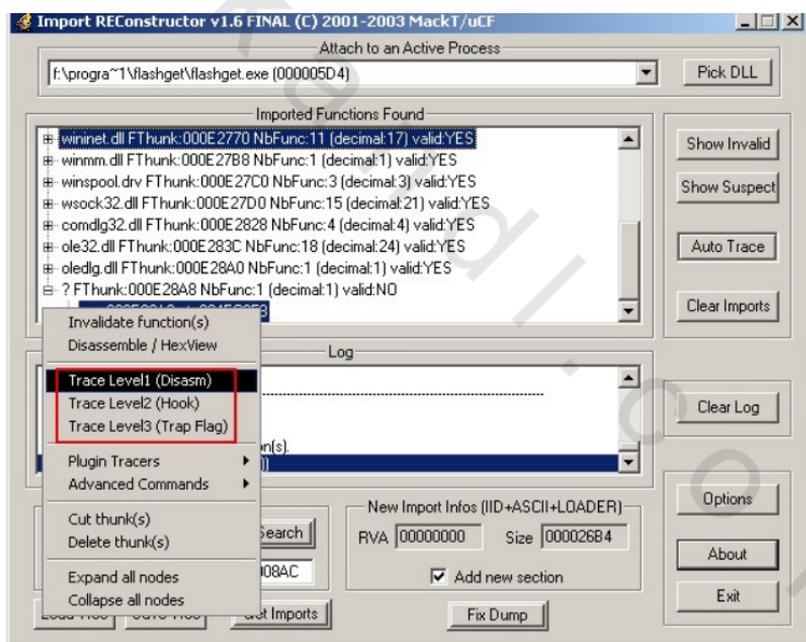
الخطوة التالية هي الضغط على المفتاح show invalide لإظهار الدوال الفاسدة فقط وتصليحها



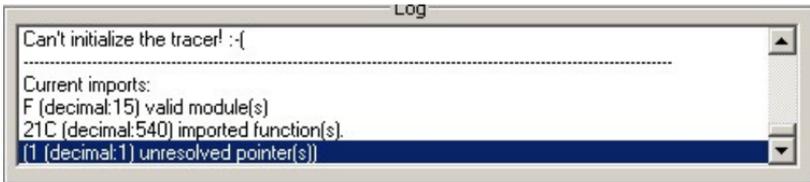
و بعد ذلك يقف البرنامج تلقائيا على الدالة الغير صالحه كما بالشكل التالي:



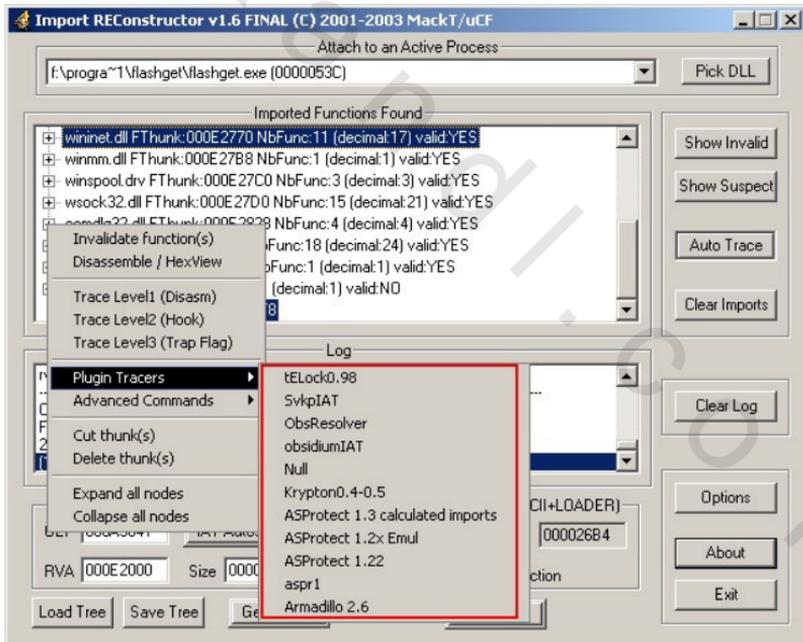
و بالضغط على المفتاح الأيمن للماوس يمكنك أن ترى قائمة الأوامر المختصرة وسنقوم هنا باستخدام احد أوامر التتبع trace لإزالة الدوال الغير صالحه



و تظهر النتيجة في الجزء Log كما يلي:

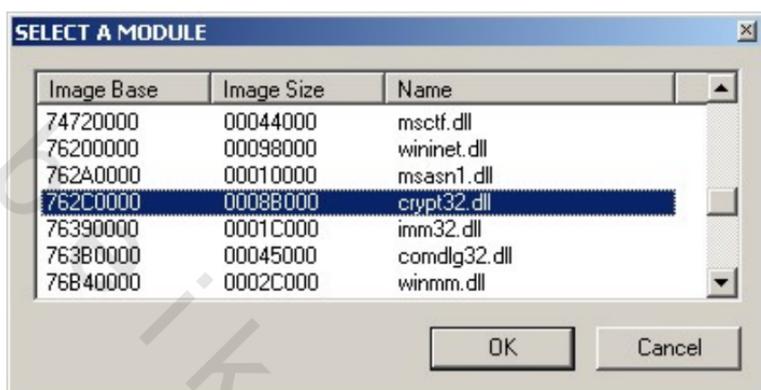


هذا ويحتوى البرنامج على عدة وظائف إضافية plugins يمكنك من خلالها تصليح الدوال التي لم تستجيب لأوامر trace وسيحدث ذلك إذا كان الملف مشفرا بأي من طرق التشفير المعروفة مثل armadillo او aspack ولعلاج ذلك قم باختيار نوع التشفير من القائمة المختصرة كما بالشكل:

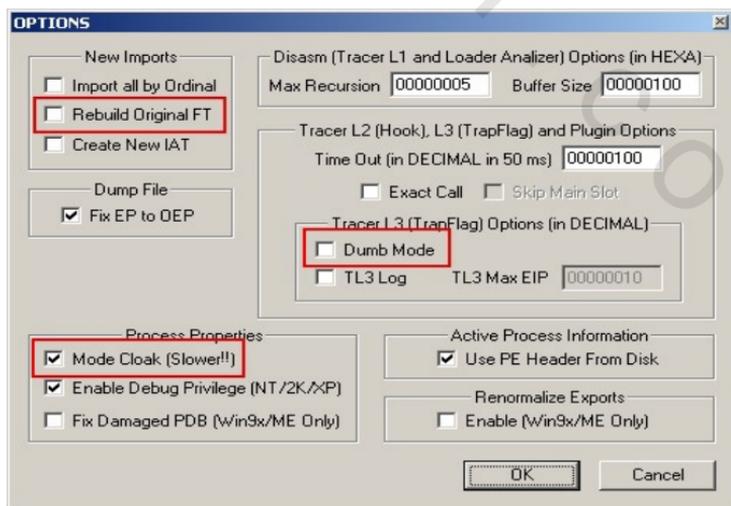


و يمكنك أيضا إصلاح باقي أجزاء الملف من المفتاح fix dump

و إذا أردت معرفة OEP الخاصة بملف DLL للملف التنفيذي الذي قمت باختياره فيمكنك ذلك من المفاتيح pick dll واختار الملف ثم اختار ok

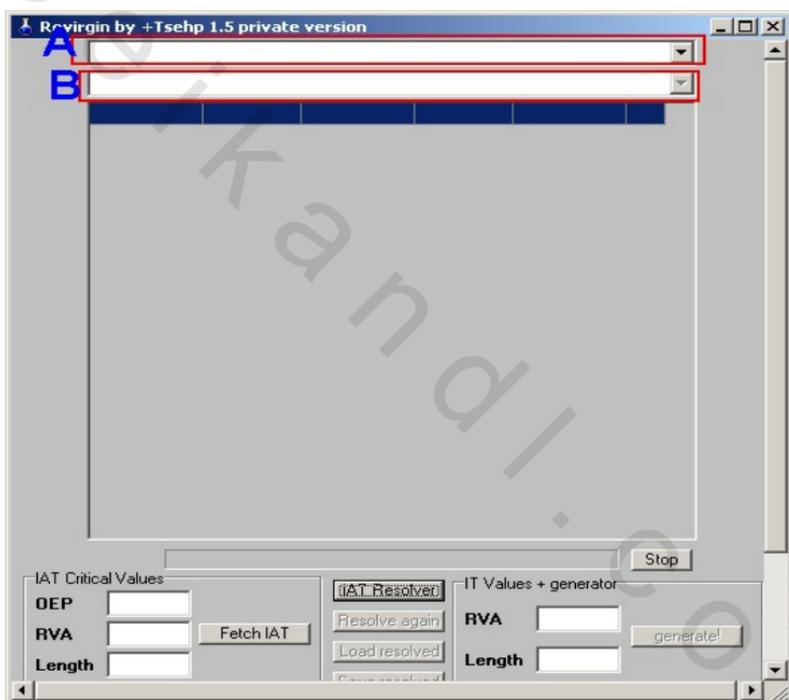


ويمكنك ضبط اختيارات البرنامج من المفاتيح option ومن التضييقات المفيدة الاختيار mode clock الذي يقوم بإخفاء البرنامج rec من الملف المحمل حتى لا يستطيع كشفه وإغلاق نفسه كما يوجد اختيارات أخرى لتغيير طريقة التتبع والفحص



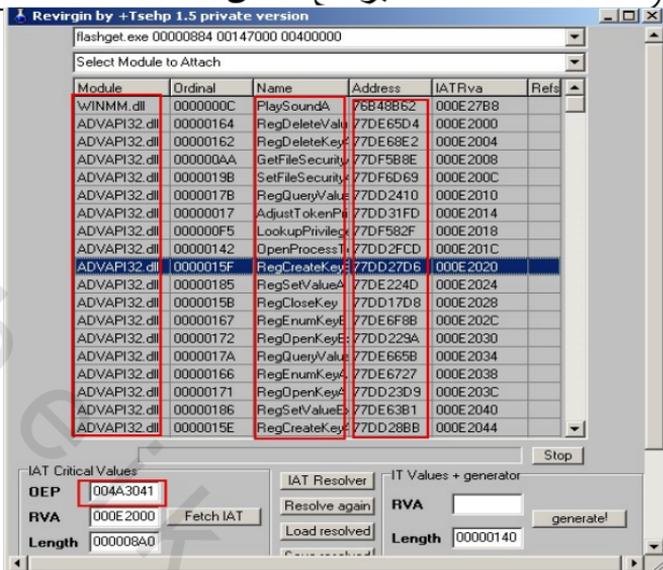
هذا البرنامج مشابه تقريبا للبرنامج Import REC ولكنه يحتوي على بعض المعلومات الإضافية نستعرضها فيما يلي:

أولا هذا البرنامج يحتاج للتهيئة أولا وبعد ذلك قم بتشغيل الايقونه الخاصة بالبرنامج من قائمة الويندوز revirgin.exe وسوف تشاهد الشاشة الافتتاحية التالية:



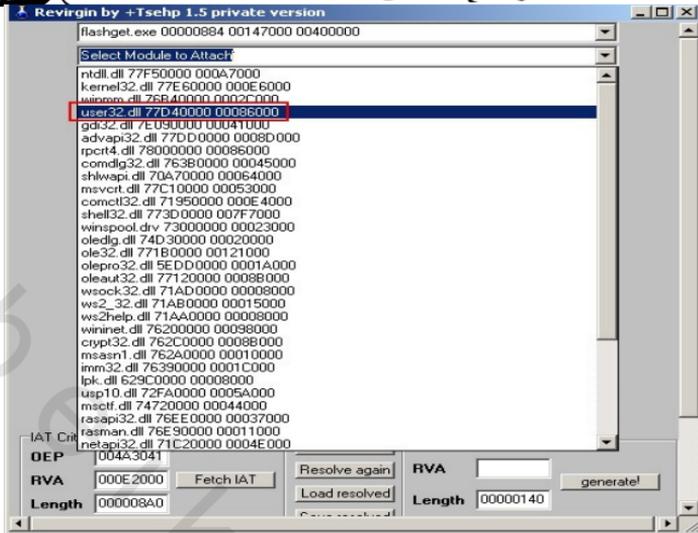
ومثل البرنامج Import REC قم باختيار الملف من الذاكرة من أول قائمه منسدلة فيقوم البرنامج بعرض كل ملفات المكاتبات (DLL Files) التي يقوم البرنامج بالنداء عليها والعنوان الخاص بكل داله واسم الدالة المستخدمة في الملف

## برامج فحص PE



و كما يتضح من الشكل السابق نشاهد البيانات في جدول مقسم إلى أعمده ويوجد بها كل من اسم الملف واسم الدالة وعنوان النداء على هذه الدالة في الملف

و يمكنك أيضا متابعه النداء على ملف معين باختياره من القائمة المنسدلة الثانية كما بالشكل:



بعد ذلك لتصليح دوال معينه يمكن الضغط على مفتاح IAT Resolver ولكن لاحظ أن هذا البرنامج قد يسبب فقدان للبيانات بالذاكرة في ويندوز اكس بي لذلك من الأفضل تجربة ذلك في ويندوز 98

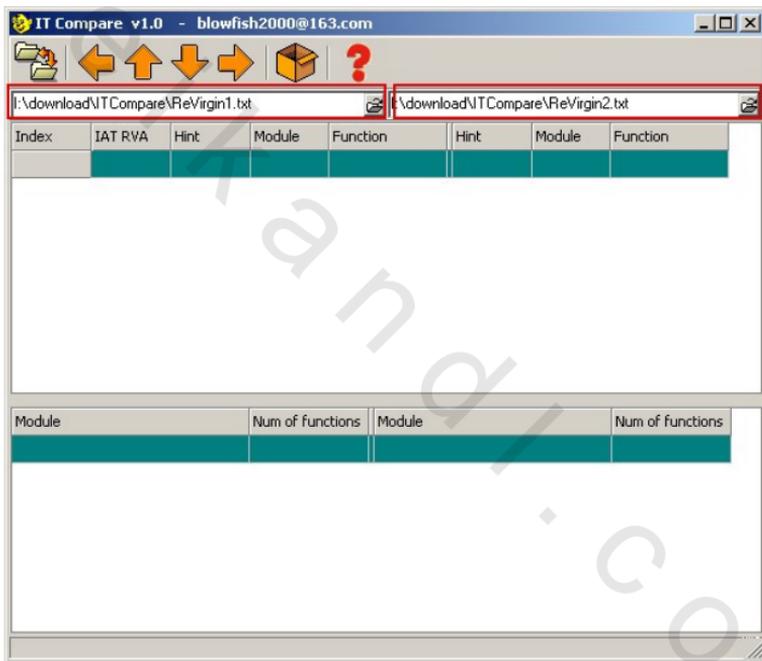


ثم بعد ذلك قم باختيار الدوال التي بها مشاكل بالماوس واختار من القائمة المختصرة الأمر Tracer:

**Tracer**

Edit	ASProtect 1.2x Emul.dll
Trace All	tELock.dll
Api Emulator	visual protect3.1.6.dll
Reset	

يتيح هذا البرنامج دقة أكثر وسهولة لرؤية الاختلاف بين ملفين والوصول إلى دوال API التي تحتاج إلى إصلاح فيعد استخدامك لبرنامج Import REC أو برنامج ReVirgin يمكنك عن طريق احد البرنامجين توليد ملفات مقارنة ويجب أن تكون هنا ملفات نصية txt file قم بتشغيل البرنامج وحدد الملفين المراد مقارنتهما كما بالشكل :



بعد ذلك قم بالضغط على مفتاح بداية المقارنة



سيبدأ البرنامج فوراً مقارنة الملفين ويظهر النتائج كالتالي:

Index	IAT RVA	Hint	Module	Function	Hint	Module	Function
0000000D	0022C2A0	000002C1	KERNEL32.	SetCurrentDir	00000289	KERNEL32.	SetCurrentDir
0000000E	0022C2A4	00000252	KERNEL32.	MultiByteToWide	00000203	KERNEL32.	MultiByteToWide
0000000F	0022C2A8	00000360	KERNEL32.	lstrlenA	00000335	KERNEL32.	lstrlen
00000010	0022C2AC	0000035D	KERNEL32.	lstrcpyNA	00000332	KERNEL32.	lstrcpyN
00000011	0022C2B0	0000022A	KERNEL32.	LoadLibraryExA	000001E1	KERNEL32.	LoadLibraryExA

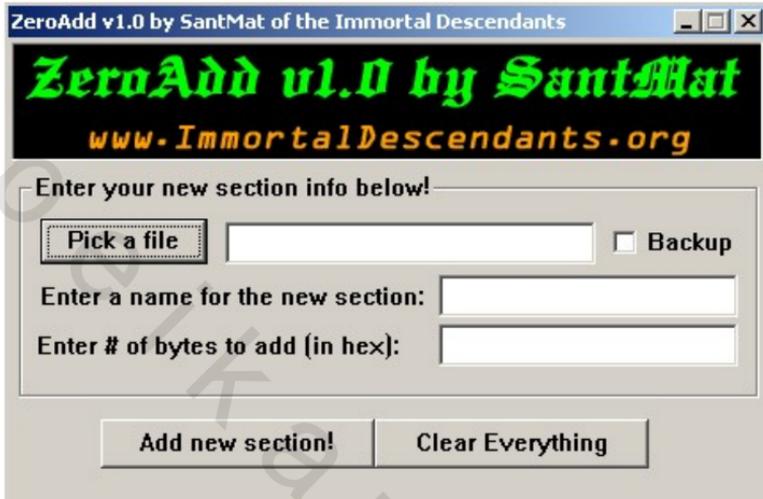
Module	Num of functions	Module	Num of functions
KERNEL32.dll	130	KERNEL32.dll	129
USER32.dll	202	USER32.dll	200
ADVAPI32.dll	13	ADVAPI32.dll	13
OLEAUT32.dll	26	OLEAUT32.dll	26
MPR.dll	1	MPR.dll	1
VERSION.dll	3	VERSION.dll	3
GDI32.dll	109	GDI32.dll	109

تحتوى النافذة العلوية على كل دوال API والنافذة السفلى على إحصائيات لعدد الدوال ويمكنك أن تشاهد الاختلاف باللون القرمزي والدوال التي ليست بها اختلاف باللون الأبيض ويمكنك الذهاب بسرعة لمواقع الاختلاف عن طريق مفاتيح الأسهم



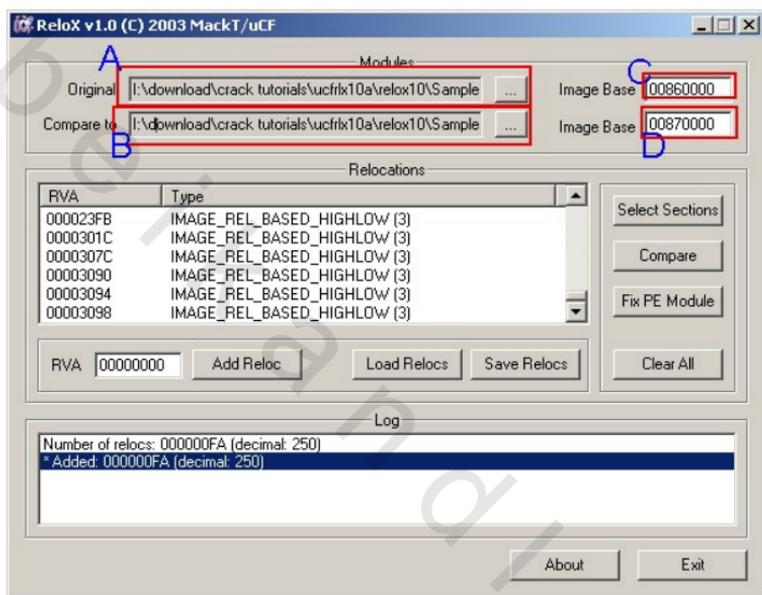
فأول مفتاح للذهاب إلى أول اختلاف والثاني الاختلاف الذي يسبقه والثالث الاختلاف الذي يليه والأخير إلى آخر اختلاف.

هذا البرنامج بسيط وهو يقوم ببساطه بإضافة مقطع جديد للملف



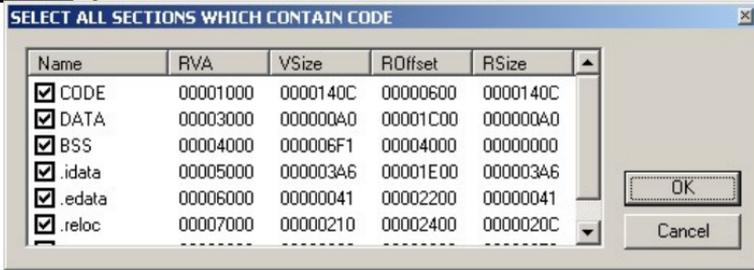
قم باختيار ملف من المفتاح pick a file ثم اضغط على مفتاح backup إذا أردت أن يقوم البرنامج بإنشاء نسخه احتياطيه للملف وادخل اسم المقطع وعدد البايت كرقم سداسي عشر وبعد التنفيذ يجب أن ترى رسالة Ok .

هذا البرنامج يقارن بين تفرغ ملفين dumped files من برامج Import REC أو revirgin وقد تكون المقارنة في كل الملف أو احد المقاطع ويعطى أيضا الإمكانية للتصليح .

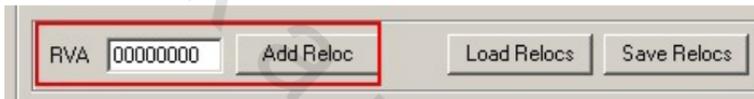


الخطوة الأولى هي تحديد ملف التفرغ للملف الأصلي ثم ملف التفرغ للملف الذي تريد مقارنته به وبعد ذلك تقوم بتحديد رقم أساس للملفين مختلف إذا لم يستطع البرنامج اكتشافهم في الخطوة C و D

قم بعد ذلك بالضغط على المفتاح Compare وسيعرض البرنامج ولكن يمكنك تحديد المقارنة على مقاطع معينة عن طريق الضغط على المفتاح Select Sections

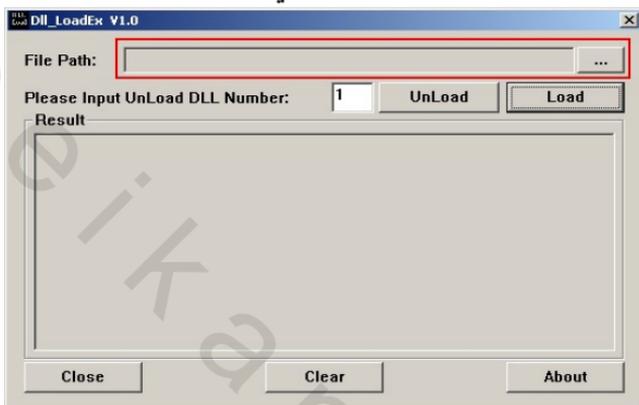


قم بعد ذلك بالضغط على المفتاح compare وإذا وجدت فروق يمكنك إصلاح ذلك بالمفتاح fix PE وسيقوم البرنامج بإنشاء مقطع .reloc جديد أو إضافة مقطع يدوي إلى الملف بالمفتاح Add Reloc

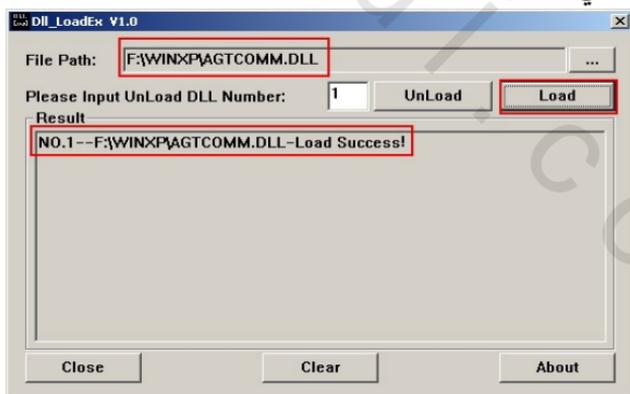


## برنامج DLL\_Loader

هذا البرنامج من الأدوات الهامة جدا لبرنامج OllyDbg فحتى يمكنك اختبار ملف dll بدون الملف التنفيذي الخاص به يجب تحميل هذا الملف في الذاكرة ويقوم هذا البرنامج تلقائيا بتحميل ملف dll في الذاكرة إذا استعملته مع برنامج ollydbg أو يدوى كما بالشكل التالي:



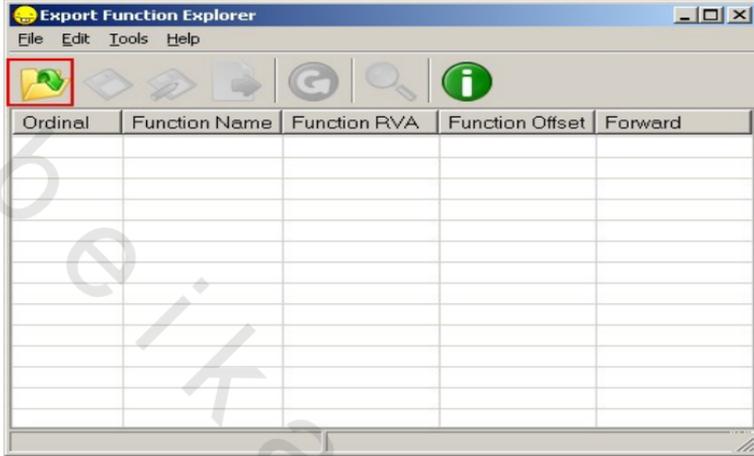
قم أولاً باختيار ملف للتحميل ثم اضغط على المفتاح load الذي يقوم بتحميل ملف المكتبة في الذاكرة



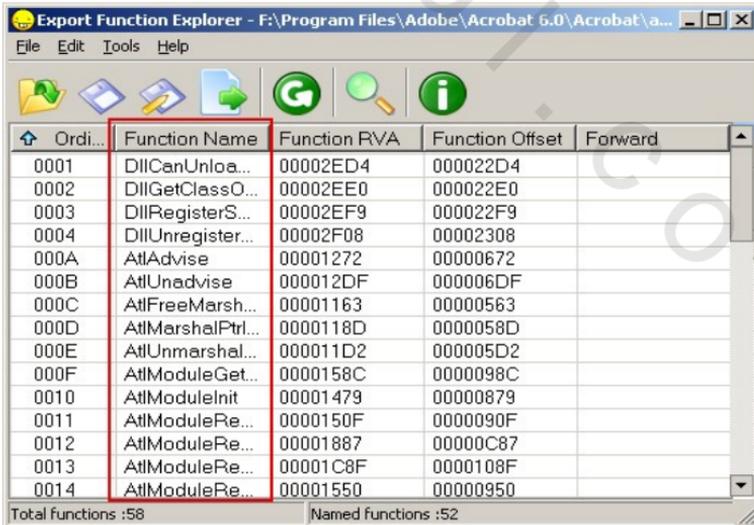
و يجب بعد الانتهاء أزاله الملف من الذاكرة حتى لا يشغل مساحه إضافية ويتم تنفيذ ذلك عن طريق المفتاح Unload

هذا البرنامج البسيط يستطيع معرفة كل الدوال الخاصة بملف exe أو

ملف dll والتي يقوم الملف بتعريفها



قم أولاً باختيار الملف المراد تحميله وسيقوم البرنامج بعرض لجميع الدوال  
ومكان كل داله حسب العنوان التخيلي RVA أو الحقيقي Offset كما يوضح  
الشكل التالي:



كما يمكن البحث عن داله معينه بواسطة مفتاح البحث



و يمكن أيضا حفظ النتيجة في ملف نصي



برامج الـ ديباجر كلها تعتمد على فكره واحده وهي إمكانية التوقف عن نقطه معينه في البرنامج وقراءه محتويات الذاكرة لهذا الجزء من البرنامج وهي من الأدوات الأساسية للمبرمجين حتى يمكن اختبار واكتشاف الأخطاء بسهوله فلا يوجد ما يسمى بالبرنامج الخالي من الأخطاء.

ومن أشهر واعرق برامج الـ ديباجر هو برنامج SoftIce فهو يعتمد على بيئة الـ دوس القوية ويستطيع إيقاف أي برنامج عند تحميله في الذاكرة وقراءة محتوياته ولكن للأسف هذا البرنامج الآن غير متوافق مع النظام xp وما يليه وقد يؤدي تماما إلى تلف الـ ويندوز لذلك قام احد الكراكر بصنع برنامج جديد قوى يسمى ollydbg وهو ما سنركز عليه هنا.

### برنامج OllyDbg:

يعتمد البرنامج على البنية 32 bit لتحليل أي ملف إلى لغة الـ اسمبلى ويدعم كل المعالجات الحديثه تقريبا مثل 80x86, Pentium, MMX, 3Dnow!, Athlon extentions, SSE و يحتوى البرنامج على أكثر من 100 اختيار تتحكم في طريقة عمل البرنامج واستخدامه للمعالج وتغيير هيئة البرنامج نفسه تحتوى النافذة الرئيسية للبرنامج على قوائم للأوامر ومفاتيح اختصار يمكن عن طريقها إظهار أو إخفاء نوافذ البرنامج .

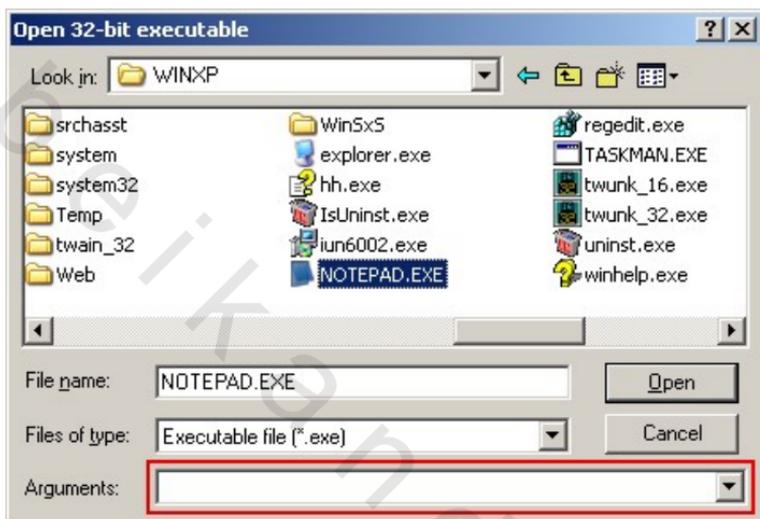
## برنامج فحص PE

أول خطوه هي اختيار برنامج عن طريق الأمر | File | Open أو اختياره من الذاكرة إذا تم تشغيل البرنامج من قبل:

Process	Name	Window	Path
00000184	smss		SystemRoot\System32\smss.exe
0000023C	csrss		SystemRoot\System32\csrss.exe
00000254	winlogon	NetDDE Agent	SystemRoot\System32\winlogon.exe
00000280	services		SystemRoot\System32\services.exe
0000028C	lsass		SystemRoot\System32\lsass.exe
00000330	svchost		SystemRoot\System32\svchost.exe
00000368	svchost		SystemRoot\System32\svchost.exe
000003C0	ACDPhoto	Preview Bar	Program Files\ACD Systems\ACD Photo
000003C4	svchost		SystemRoot\System32\svchost.exe
00000494	spoolsv		SystemRoot\System32\spoolsv.exe
00000538	EXPLORER	?????-???? - Microsoft Int	Program Files\Internet Explorer\IEX
0000056C	Explorer	SysFader	SystemRoot\System32\Explorer.EXE
00000574	Hppoa		Games\HP3\system\Hppoa.exe
00000644	WINWORD	Bold	Program Files\Microsoft Office\Offi
00000654	taskswit		SystemRoot\System32\taskswitch.exe
00000674	ctfmon	CiceroUIWndFrame	SystemRoot\System32\ctfmon.exe
00000688	alg		SystemRoot\System32\alg.exe
000006F0	KRUPF	Kaspersky Anti-Hacker	Program Files\Kaspersky Lab\Kaspers

و يتضح من الشكل السابق أن النافذة تحتوى على البيانات التالية رقم البرنامج - اسم البرنامج - اسم أو عنوان النافذة الرئيسية للبرنامج - المسار للملف التنفيذي الخاص بالبرنامج.

أما الأمر Open فهو يختلف في شيء واحد فقط عن أمر Open وهو احتوائه على سطر لأوامر المعاملات وبالتالي يمكن إعطاء الملف المراد اختباره قيمه معينه ومشاهدة كيف يتم تنفيذ البرنامج بناء على هذه القيمة



القائمة التالية وهي View وتحتوى على النوافذ المختلفة لبرنامج ollydbg وكما يحتوى سطر المفاتيح المختصرة على مفاتيح لكل منها حرف يعبر عن نافذة معينه

### النافذة LOG:

ويعبر عنها المفتاح L تحتوى على معلومات عن عملية تحميل الملف الخارجي فيمكنك مشاهدة عملية التحميل للملف وملفات dll الخاصة به وما إذا كان تم تحميل هذه الملفات بصوره صحيحة.



Dump in cpu: لعرض محتويات الملف في نافذة CPU

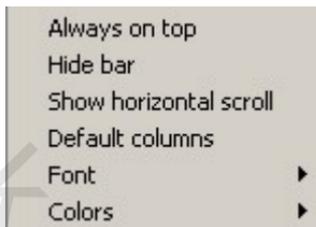
Clear window: لمسح محتويات النافذة

Log to file: لتخزين معلومات log في ملف مباشرة

Copy to clipboard: لنسخ معلومات النافذة ويمكنك أن تختار سطر واحد

أو كل النافذة أو عنوان محدد

Appearance:



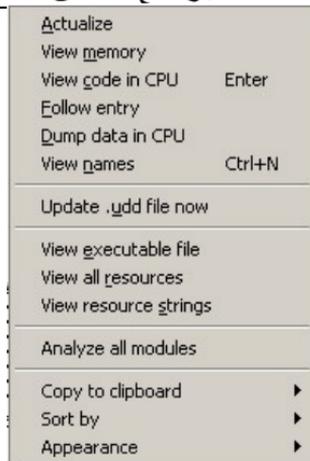
و بها الأوامر التالية : تمكين النافذة لتصبح دائما في المقدمة - إخفاء أو إظهار شريط الحالة - إظهار أو إخفاء شريط التمرير - الأعمدة الافتراضية للظهور - نوع الخط وحجمه - الألوان

### نافذة Executable Modules :

تحتوي هذه النافذة على الملفات التنفيذية (سواء exe أو dll) الذي قام البرنامج بالنداء عليها كما يظهر البرنامج الملف الحالي للنداء عند مكان التوقف بلون احمر

Base	Size	Entry	Name	File version	Path
00320000	00006000	003224AF	ogg		G:\Games\HP3\system\ogg.dll
00330000	0001F000	00349320	vorbis		G:\Games\HP3\system\vorbis.dll
00350000	00068400	00648D08	Engine		G:\Games\HP3\system\Engine.dll
00410000	0003C000	00A2F1BC	IF23	2.3.4	G:\Games\HP3\system\IF23.dll
00450000	00122000	00A769D2	DD08	5.3.0000001.090	F:\WINXP\System32\DD08.DLL
00530000	00006000	00B813A5	d3d8thk	5.3.0000000.900	F:\WINXP\System32\d3d8thk.dll
10000000	00007000	100030B2	vorbisfif		G:\Games\HP3\system\vorbisfile.dll
10100000	00152000	10159C20	Core		G:\Games\HP3\system\Core.dll
10900000	00082000	10952000	Hppoa		G:\Games\HP3\system\Hppoa.exe
11000000	00000000	11037D37	WINDOW		G:\Games\HP3\system\WINDOW.DLL
30000000	00072000	30006AB1	binkw32	1.5u	G:\Games\HP3\system\binkw32.dll
51000000	00050000	510348B4	DDRAW	5.3.0000001.090	F:\WINXP\System32\DDRAW.DLL
629C0000	00008000	629C2C82	LPK	5.1.2600.0 (xpc	F:\WINXP\System32\LPK.DLL
70A70000	00064000	70A78386	SHLWAPI	6.00.2800.1106	F:\WINXP\system32\SHLWAPI.dll
71950000	000E4000	7195EDD8	comctl32	6.0 (xpsp1.0200)	F:\WINXP\WinSxS\x86_Microsoft.Windows.Common-...
72F40000	00000000	72F44361	USP10	1.9409.2606.1106	F:\WINXP\System32\USP10.dll
73BC0000	0000F000	73BC106C	DCIMAN32	5.1.2600.0 (xpc	F:\WINXP\System32\DCIMAN32.dll
76390000	0001E000	763912A4	IMH32	5.1.2600.1106 (	F:\WINXP\System32\IMH32.DLL
763B0000	00045000	763B1604	condlg32	6.00.2800.1106	F:\WINXP\system32\condlg32.dll
76B40000	00021000	76B4297E	WINMM	5.1.2600.1106 (	F:\WINXP\System32\WINMM.dll
771B0000	00121000	771C0783	ole32	5.1.2600.1263 (	F:\WINXP\system32\ole32.dll
77340000	0000E000	773419ED	COMCTL32	5.82 (xpsp1.0200)	F:\WINXP\System32\COMCTL32.dll
773D0000	007F7000	773F6164	SHELL32	6.00.2800.1106	F:\WINXP\System32\SHELL32.dll
77520000	00007000	7752116A	USER32	5.1.2600.0 (xpc	F:\WINXP\system32\USER32.dll

## برامج فحص PE



و تحتوى قائمة الأوامر المختصرة على الأوامر التالية:  
**Actualize**: وضع علامة أن البرنامج قديم فلا يتم تمييزه باللون الأحمر  
**View memory**: تمكننا من مشاهدة محتويات الملف الحالي بالذاكرة

Address	Size	Owner	Section	Contains	Type	Access	Initial	Mapped as
72FA0000	00001000	USP10		PE header	Image	R	RWE	
72FA1000	0003A000	USP10	.text	code, import	Image	R	RWE	
72FDB000	00009000	USP10	.data	data	Image	R	RWE	
72FE4000	00002000	USP10	Shared		Image	R	RWE	
72FE6000	00012000	USP10	.rsrc	resources	Image	R	RWE	
72FF8000	00002000	USP10	.reloc	relocations	Image	R	RWE	
73000000	00001000	WINSPOOL		PE header	Image	R	RWE	
73001000	0001D000	WINSPOOL	.text	code, import	Image	R	RWE	
7301E000	00002000	WINSPOOL	.data	data	Image	R	RWE	
73020000	00001000	WINSPOOL	.rsrc	resources	Image	R	RWE	
73021000	00002000	WINSPOOL	.reloc	relocations	Image	R	RWE	
76390000	00001000	IMMS2		PE header	Image	R	RWE	
76391000	00014000	IMMS2	.text	code, import	Image	R	RWE	
763A5000	00001000	IMMS2	.data	data	Image	R	RWE	
763A6000	00005000	IMMS2	.rsrc	resources	Image	R	RWE	
763AB000	00001000	IMMS2	.reloc	relocations	Image	R	RWE	
763B0000	00001000	condlg32		PE header	Image	R	RWE	
763B1000	0002C000	condlg32	.text	code, import	Image	R	RWE	
763DD000	00004000	condlg32	.data	data	Image	R	RWE	
763E1000	00011000	condlg32	.rsrc	resources	Image	R	RWE	
763F2000	00003000	condlg32	.reloc	relocations	Image	R	RWE	
77340000	00001000	COMCTL32		PE header	Image	R	RWE	
77341000	00066000	COMCTL32	.text	code, import	Image	R	RWE	
77347000	00001000	COMCTL32	.data	data	Image	R	RWE	
773A8000	0001F000	COMCTL32	.rsrc	resources	Image	R	RWE	
773C7000	00004000	COMCTL32	.reloc	relocations	Image	R	RWE	
773D0000	00001000	SHELL32		PE header	Image	R	RWE	
773D1000	001E0000	SHELL32	.text	code, import	Image	R	RWE	
775E1000	0001C000	SHELL32	.data	data	Image	R	RWE	
775CD000	0005E000	SHELL32	.rsrc	resources	Image	R	RWE	
77BA0000	0001A000	SHELL32	.reloc	relocations	Image	R	RWE	
77C10000	00001000	USER32		PE header	Image	R	RWE	

**View code in cpu**: يمكننا هذا الأمر من مشاهدة تعليمات الاسمبلى في نافذة CPU للملف الحالي  
**Follow entry**: تمكننا من تتبع البرنامج المختار في نافذة cpu



يمكننا هذا الأمر من مشاهدة محتويات الملف بالنظام السداسي عشر في نافذة  
dump  
:View All Resources

Address	Type	Name	Language	Size	Information
71AA60A0	VERSION	1	0409 English	000003A8	Windows Socke
71AA6448	STRING	1	0409 English	0000009E	Windows Socke

تمكننا هذه النافذة من معرفة جميع المصادر (سواء النصية أو حتى الصور)  
الخاصة بالبرنامج وعنوان كل مصدر  
:View resource strings

Address	Index	Language	String
763E2372	170	0409 English	Save & in:
763E2386	171	0409 English	&Save
763E2392	172	0409 English	&Open
763E239E	173	0409 English	&Print
763E23CA	180	0409 English	Open
763E23D4	181	0409 English	Save As
763E23E4	182	0409 English	Save file as &type:
763E240C	183	0409 English	Drive %c: does not exist..Please verify the approx
763E249E	184	0409 English	Windows is unable to read drive %c:. Make sure th
763E259C	185	0409 English	A different disk is expected in drive %c:. Pleas
763E263C	186	0409 English	The disk in drive %c: is not formatted;.Please in
763E2706	187	0409 English	%s.File not found..Please verify the correct file
763E270A	188	0409 English	%s.Path does not exist..Please verify the correct
763E280E	189	0409 English	%s.The above file name is invalid.
763E2854	18A	0409 English	%s.This file is already in use..Select a new name
763E291A	18B	0409 English	%s.Cannot access this file..Check security privil
763E29B6	18C	0409 English	%s.This file exists with Read Only attributes..Pl
763E2A5A	18E	0409 English	%s.This file name is a reserved device name..Plea
763E2AEC	18F	0409 English	Disk %c: is write-protected..A file cannot be sav
763E2B8A	190	0409 English	This directory of disk %c: is full..Choose anothe

وتمكننا هذه النافذة من مشاهدة المصادر النصية فقط التي يحتويها البرنامج  
:Analyze all modules يمكننا هذا الأمر من تحليل كل البرامج وبالطبع  
سيأخذ تنفيذ هذا الأمر بعض الوقت

Copy to clipboard: لنسخ محتويات النافذة إلى الذاكرة ويمكنك نسخ سطر واحد أو عدة اسطر أو حتى أعمده محددة.

Sort by : لتغيير نظام الفهرسة وترتيب البيانات في الذاكرة حسب عمود معين

Appearance : لتغيير شكل النافذة كما سبق شرحه.

### النافذة Memory Map:

وتعرض محتويات الذاكرة للملف الذي تم تحميله ومكان كل مقطع من الملف مثلا المقطع data. ومكانه بالذاكرة وحجمه ونوعه .

Address	Size	Owner	Section	Contains	Type	Access	Initial	Mapped as
00010000	00001000				Priv	RW	RW	
00020000	00001000				Priv	RW	RW	
00120000	00001000				Priv	RW	Guard	
00120000	00003000			stack of ma	Priv	RW	Guard	
00130000	00001000				Map	R	R	
00140000	00006000				Priv	RW	RW	
00240000	00006000				Priv	RW	RW	
00250000	00001000				Map	RW	RW	
00260000	00016000				Map	R	R	
00280000	00034000				Map	R	R	\\Device\Harddisk
002C0000	00041000				Map	R	R	\\Device\Harddisk
00310000	00006000				Map	R	R	\\Device\Harddisk
00320000	00001000	ogg		PE header	Map	R	R	\\Device\Harddisk
00321000	00002000	ogg	.text	code	Image	R	RWE	
00323000	00001000	ogg	.rdata	imports,exp	Image	R	RWE	
00324000	00001000	ogg	.data	data	Image	R	RWE	
00325000	00001000	ogg	.reloc	relocations	Image	R	RWE	
00330000	00001000	vorbis		PE header	Image	R	RWE	
00331000	00019000	vorbis	.text	code	Image	R	RWE	
00334000	00001000	vorbis	.rdata	imports,exp	Image	R	RWE	
00334000	00003000	vorbis	.data	data	Image	R	RWE	
0034E000	00001000	vorbis	.reloc	relocations	Image	R	RWE	
00350000	00001000	Engine		PE header	Image	R	RWE	
00351000	00351000	Engine	.text	code	Image	R	RWE	
006A2000	00005000	Engine	.rdata	exports	Image	R	RWE	
00777000	00230000	Engine	.data	data	Image	R	RWE	
00924000	00008000	Engine	.idata	imports	Image	R	RWE	
009BF000	00045000	Engine	.reloc	relocations	Image	R	RWE	
009A10000	00001000	IFC23		PE header	Image	R	RWE	
009A11000	00027000	IFC23	.text	code	Image	R	RWE	
009A30000	0000C000	IFC23	.rdata	imports,exp	Image	R	RWE	
009A40000	00004000	IFC23	.text	code	Image	R	RWE	

و تحتوي قائمة الأوامر المختصرة للنافذة على التالي:

Actualize	
Dump in CPU	
Dump	
Search	Ctrl+B
Set break-on-access	F2
Set memory breakpoint on access	
Set memory breakpoint on write	
Set access	▶
Copy to clipboard	▶
Sort by	▶
Appearance	▶



يقوم هذا الأمر بوضع نقطة إيقاف عند مكان معين بالذاكرة حتى يتوقف البرنامج عندما يصل إلى هذه النقطة.

Address	Size	Owner	Section	Contains	Type	Access	Initial	Mapped as
00072000	001F5000	Acrobat	.rsrc	resources	Inag	R	RWE	
00067000	00075000	Acrobat	.reloc	relocations	Inag	R	RWE	
000E0000	00005000				Map	R	RWE	
00EA0000	00002000				Map	R	RWE	
00EB0000	00103000				Map	R	RWE	
00FC0000	00073000				Map	R	RWE	
012C0000	00001000				Priv	RW	RWE	
01340000	00004000				Priv	RW	RWE	
01350000	00004000				Priv	RW	RWE	
05000000	00001000	ACE		PE header	Inag	R	RWE	
05001000	0005C000	ACE	.text	code	Inag	R	RWE	
0505D000	0001D000	ACE	.rdata	imports,exp	Inag	R	RWE	
0507A000	00003000	ACE	.data	data	Inag	R	RWE	
05085000	00001000	ACE	.rsrc	resources	Inag	R	RWE	
05087000	00007000	ACE	.reloc	relocations	Inag	R	RWE	
06000000	00001000	AGM		PE header	Inag	R	RWE	
06001000	0010E000	AGM	.text	code	Inag	R	RWE	
0610F000	00032000	AGM	.rdata	imports,exp	Inag	R	RWE	
06141000	00042000	AGM	.data	data	Inag	R	RWE	
06183000	00001000	AGM	.rsrc	resources	Inag	R	RWE	
06184000	00017000	AGM	.reloc	relocations	Inag	R	RWE	
07000000	00001000	BIB		PE header	Inag	R	RWE	
07001000	00116000	BIB	.text	code	Inag	R	RWE	
07017000	00006000	BIB	.rdata	imports,exp	Inag	R	RWE	
0701D000	00007000	BIB	.data	data	Inag	R	RWE	
07024000	00001000	BIB	.rsrc	resources	Inag	R	RWE	
07025000	00003000	BIB	.reloc	relocations	Inag	R	RWE	
08000000	00001000	CoolType		PE header	Inag	R	RWE	
08001000	00104000	CoolType	.text	code	Inag	R	RWE	
08105000	0001E000	CoolType	.rdata	imports,exp	Inag	R	RWE	
08123000	00006000	CoolType	.data	data	Inag	R	RWE	
08123000	00001000	CoolType	.rsrc	resources	Inag	R	RWE	

لاحظ مكان نقطة الإيقاف باللون الأحمر

أوامر Memory Breakpoints تمكننا من وضع نقطة إيقاف عند قراءة أو كتابة على جزء محدد من الذاكرة  
 set access: يمكننا من تحديد نوعه التعامل مع هذا الجزء من الذاكرة فيمكننا من وضع القيم التالية

No access  
 Read only  
 Read/write  
 Execute  
 Execute/read  
 Full access

: النافذة Threads

## برامج فحص PE

Ident	Entry	Data block	Last error	Status	Priority	User time	System time
00000220	00410070	7FFDE000	ERROR_MOD_NOT_FOU	Active	32 + 0	0.0156 s	0.0156 s

عند إيقاف برنامج للمراقبة يمكنك أن تشاهد حاله البرنامج الحالي أو الملفات الفرعية التي يقوم بتحميلها وتبين هذه النافذة في العمود status حالة الإيقاف فقد تكون احد القيم التالية:

Active : أن البرنامج يعمل أو متوقف مؤقتا عند نقطة توقف محدد

Suspended : البرنامج متوقف

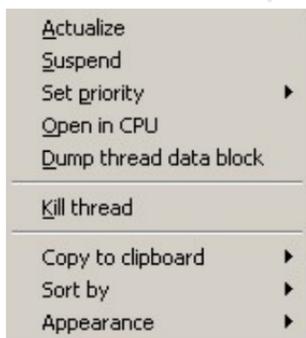
Traced : البرنامج متوقف لكن برنامج ollydbg يقوم بفحص سطور البرنامج

Paused: البرنامج يعمل لكن ollydbg قام بإيقافه مؤقتا حتى يمكنه فحص

البرامج الأخرى

Finished: البرنامج متوقف تماما

و تحتوى قائمه الأوامر المختصرة لهذه النافذة على الأوامر التالية:

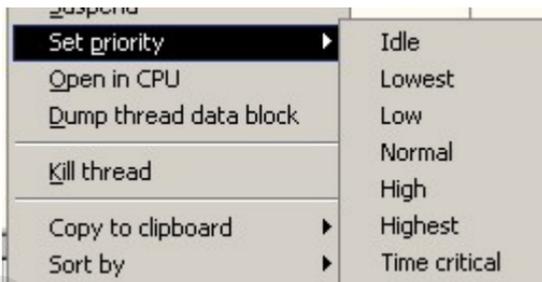


Actualize: وضع علامة على كل البرامج أنها تم تنفيذها من قبل

Suspend: إيقاف البرنامج

Resume: يظهر بعد استخدام الأمر السابق لمتابعة تنفيذ البرنامج

Set priority: يمكن وضع احد القيم التالية لأولية تنفيذ البرنامج



Idle

Lowest

Low

Normal

High

Highest

Time critical - أعلى قيمه

Open in CPU: فتح حالة خط البرنامج الحالي في نافذة CPU ويمكن تنفيذ

ذلك بالضغط المزدوج

Copy to clipboard: يمكننا هذا الأمر من نسخ محتويات الذاكرة إلى

ذاكرة clipboard

Whole line: يمكننا من نسخ السطر الحالي إلى الذاكرة كعدة سطور

Whole table: تنسخ كل محتويات الذاكرة كنص متعدد السطور

Appearance: وهي الشكل العام للنافذة مثل الألوان ومكان النافذة كما سبق

وأوضحنا.

**النافذة Windows:**

## برامج فحص PE

Handle	Title	Parent	WinProc	ID	Style	ExtStyle	Thread

وتعرض كل النوافذ الخاصة بالبرنامج الذي قمنا بتحميله ومعاملات كل نافذة. ويمكنك عن طريق هذا النافذة وضع نقاط توقف للرسائل التي تستقبلها النافذة مثل الرسالة WM\_PAINT أو حتى مجموعته من الرسائل الخاصة بالماوس فمثلا إذا كان لدينا مفتاح له التعريف 00001234 وبعد وضع نقطة الإيقاف يمكنك مشاهدة البيانات التالية :

Condition: [ESP+4]==00001234 && [ESP+8] IN (0F0..0F7,135)

Explanation: <WinProc>

Pause program: On condition

و يمكن تفسير المعاملات السابقة لنقطة التوقف كما يلي

[ESP+00] Return address  
 [ESP+04] Window's handle  
 [ESP+08] Message  
 [ESP+0C] wParam  
 [ESP+10] lParam

و هذا يعني انه يتم التوقف عند تعريف رقم 00001234 للنافذة والرسائل التي يتم التعرف عليها هي

BM\_GETCHECK...BM\_SETIMAGE,  
 WM\_CTLCOLORBTN

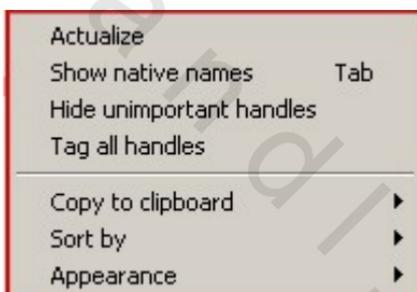
لاحظ أن رقم تعريف النافذة (window handle) يتغير في كل مرة نقوم بها بتنفيذ البرنامج لذلك لن يكون شرط نقطة التوقف صالح بعد إغلاق النافذة.

إذا أردت أن تعرف المزيد عن الرسائل راجع الملحق أ

Handle	Type	Refs	Access	T	Info	Name
00000028	Desktop	2185	000F01FF			\Default
00000035	Directory	69	00000003			\KnownDlls
00000014	Directory	32	000F000F			\Windows
00000034	Directory	695	0002000F			\BaseNamedObjects
00000020	Event	3	001F0003			
0000000C	File (dir)	2	00100020			F:\Program Files\Adobe\Acrobat 6.0\Acrobat
0000001C	File (dir)	2	00100020			F:\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Cont
00000038	Key	2	000F003F			HKEY_LOCAL_MACHINE
00000050	Key	2	000F003F			HKEY_CURRENT_USER
00000054	Key	2	000F003F			HKEY_CLASSES_ROOT
00000004	KeyEvent	30	000F0003			\KernelObjects\CritSecOutOfMemoryEvent
0000007C	Mutant	19	00120001			\BaseNamedObjects\ShInCacheMutex
00000018	Port	3	001F0001			
00000010	Section	23	000F001F			\BaseNamedObjects\ShInSharedMemory
00000080	Section	19	00000002			

تحتوي هذه النافذة على الرقم التعريفي handle للنوافذ الخاصة بالبرنامج الذي يتم فحصه وبجانب رقم التعريف تحتوي هذه النافذة على نوع النافذة والعنوان والاسم

و تحتوي قائمة الأوامر المختصرة على الأوامر التالية:



Actualize: كما شرحنا سابقا

Show native names: افتراضيا يتم إظهار مكان النافذة في الكمبيوتر كدليل فرعى حقيقي ولكن عن طريق استخدام هذا الأمر يتم تغيير المسار والأسماء إلى المسار المنطقي

Hide unimportant handles: يقوم هذا الاختيار بإخفاء النوافذ التي ليست لها أهميه للبرنامج المراد اختباره.

Tag all handles: يمكن عن طريقها تعليم النوافذ بعلامة خاصة

Copy to clipboard: لنسخ محتويات النافذة للذاكرة

## برامج فحص PE

Sort by: لتحديد ترتيب البيانات حسب عمود محدد

Appearance: لتغيير شكل وهيئة النافذة

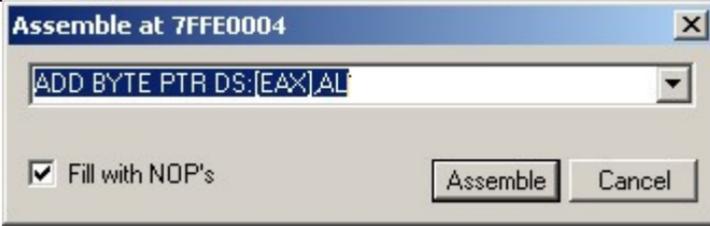
النافذة CPU:

وهي النافذة الرئيسية للبرنامج وتنقسم إلى عدة نوافذ داخلية هي:

The screenshot shows the CPU - main thread window with four tabs: Disassembler, Registers, Information, and Dump. The Disassembler tab is active, showing assembly instructions like MOV EDX, ESP and SVSCALL. The Registers tab shows the state of various registers like EAX, ECX, EDI, etc. The Information tab shows return addresses and error messages. The Dump tab shows a hex dump of memory with ASCII characters.

Disassembler:

وتحتوى هذه النافذة على كود البرنامج الذي يتم اختباره وتتضمن أربع أعمده هي العنوان والتفريغ السداسي عشر للأمر الحالي وأوامر الاسمبلى وأخيرا التعليقات الضغط المزدوج على أي أمر اسمبلى في العمود assembly يظهر دياالج يمكننا من تعديل الأمر:



و الضغط المزدوج في العمود Hex dump يمكنك من وضع نقطة توقف في هذا السطر وأخير عمود التعليقات يتيح لك إدخال تعليقات للسطر الحالي.



عندما يتم إيقاف البرنامج المراد اختباره فيتم التوقف عند الأمر الذي يعتبر EIP أو نقطة الدخول للبرنامج

و تتيح لك هذه النافذة البحث والتعديل وحفظ المتغيرات للملف التنفيذي وفيما يلي الأوامر المختصرة لهذه النافذة:

## برامج فحص PE

B <u>ackup</u>	▶
C <u>opy</u>	▶
B <u>inary</u>	▶
A <u>ssemble</u>	Space
L <u>abel</u>	:
C <u>omment</u>	;
B <u>reakpoint</u>	▶
H <u>it trace</u>	▶
R <u>un trace</u>	▶
Follow immediate constant	
F <u>ollow SE handler</u>	Enter
N <u>ew origin here</u>	Ctrl+Gray *
G <u>o to</u>	▶
F <u>ollow in Dump</u>	▶
V <u>iew call tree</u>	Ctrl+K
S <u>earch for</u>	▶
F <u>ind references to</u>	▶
V <u>iew</u>	▶
C <u>opy to executable</u>	▶
A <u>nalysis</u>	▶
B <u>ookmark</u>	▶
A <u>pearance</u>	▶

:Backup

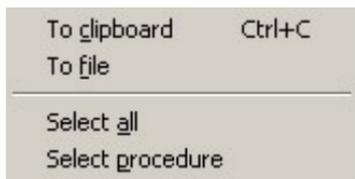
و يحتوى على الثلاث أوامر التالية:

C <u>reate backup</u>
L <u>oad backup from file</u>
S <u>ave data to file</u>

Create backup يقوم بإنشاء ملف احتياطي لبيانات الاسمبلى  
 Load backup from file يستخدم عند إجراء تعديل لأوامر الاسمبلى وتريد الرجوع عنه  
 Save data to file يقوم البرنامج بتخزين بيانات الذاكرة في ملف بالامتداد \*.mem  
 وتظهر أوامر أخرى بعد إجراء backup مثل update لتحديث النسخة الاحتياطية و delete لحذف النسخة الاحتياطية.

:Copy

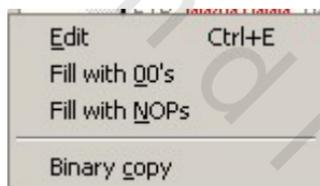
يحتوى الأوامر التالية:



يمكنك عن طريق هذه الأوامر نسخ بيانات النافذة إلى الذاكرة clipboard أو إلى ملف على الترتيب

:Binary

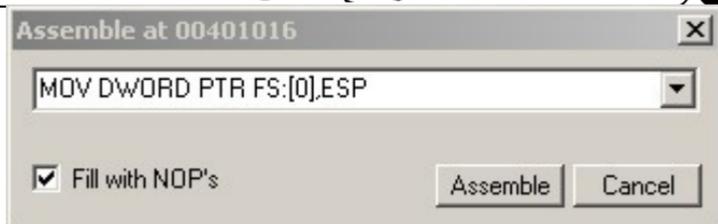
يحتوى الأوامر التالية:



تمكننا هذه القائمة من تعديل الملف بالنظام السداسي عشر أو تغيير أمر ووضع أصفار أو وضع الأمر NOP على الترتيب.

:Assemble

يعطى هذا الأمر الإمكانية لتعديل بيانات الاسمبلى في البرنامج عن طريق الديالوج التالي:



:Label

يمكن عن طريق هذا الأمر تعديل أوامر التجميع بإدراج معرف أو label في مكان معين ويتم ذلك عن طريق الحوار التالي:



:Comment

لتسجيل تعليقات مفيدة حتى يسهل معرفة الغرض من سطر معين وهذا هام جدا كما ذكرنا في الباب الأول ويمكنك كتابة التعليق في سطر واحد في الحوار الذي سيظهر



:Breakpoint

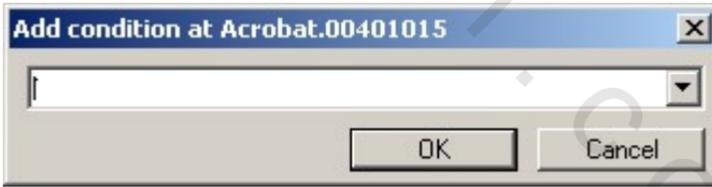
تظهر قائمه فرعيه للتعامل مع نقاط التوقف وتحتوى الأوامر التاليه:

<u>T</u> oggle	F2
<u>C</u> onditional	Shift+F2
Conditional <u>l</u> og	Shift+F4
<u>R</u> un to selection	F4
<hr/>	
Memory, on <u>a</u> ccess	
Memory, on <u>w</u> rite	
<hr/>	
<u>H</u> ardware, on execution	

:Toggle

لوضع أو إزالة نقطة إيقاف في المكان الحالي:

:Conditional



يتيح لك هذا الديالوج وضع شرط محدد للتوقف

مثال 1:  $EAX < 0$  للتوقف عندما تكون EAX اقل من الصفر

مثال 2:  $MSG == 111$  يتحقق هذا الشرط إذا كانت الرسالة

WM\_COMMAND تساوى 0111x

مثال 3:

$[ESP+8] == WM\_PAINT$

يمكنك استخدام دوال API في الشروط ويجب مراجعة ملف المساعدة الخاص بالبرنامج للحصول على المعلومات الكاملة للشروط.  
 يتم تنفيذ الشروط وفقا لأولية استخدام معامل معين ويوضح الجدول التالي المعامل وترتيب الأولوية

- |    |                |           |
|----|----------------|-----------|
| 0  | Unary          | ! ~ + -   |
| 1  | Multiplication | * / %     |
| 2  | Addition       | + -       |
| 3  | Shifts         | << >>     |
| 4  | Comparisons    | < <= > >= |
| 5  | Comparisons    | == !=     |
| 6  | Boolean AND    | &         |
| 7  | Boolean XOR    | ^         |
| 8  | Boolean OR     |           |
| 9  | Logical AND    | &&        |
| 10 | Logical OR     |           |

فمثلا

$$2+3*4 = 14$$

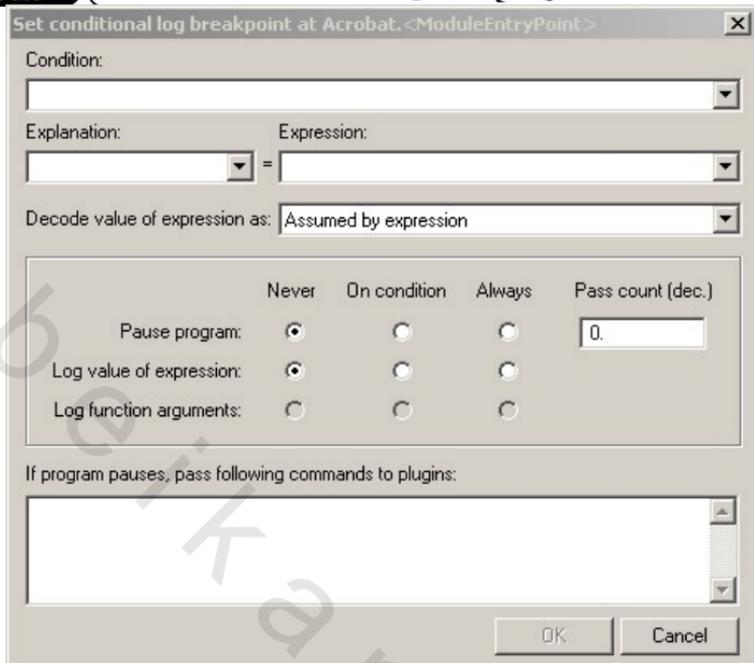
أما

$$(2+3)*4 = 20$$

يمكنك تغيير الترتيب إذا استخدمت الأقواس.

:Conditional log

يمكن عن طريق هذا الأمر وضع نقطة إيقاف شرطية لكن مع إمكانية إرسال قيمة داله أو متغير معين إلى نافذة log في كل مره يتحقق فيها الشرط ويتم إيقاف التنفيذ



ميزة هذه النافذة هي سرعة البحث في المعلومات وقيم المتغيرات بدلا من الضغط على F9 لاستكمال تنفيذ البرنامج عدد كبير من المرات ويمكنك أيضا تخزين الناتج في ملف خارجي.

الاختيار pass count يمكنك من التوقف عند عدد محدد من المرات مثلا إذا كان هناك حلقة تكراريه تتكرر 100 مره فيمكنك وضع قيمة الاختيار 99 فيتم التوقف فقط عند آخر حلقة .

ويمكنك أيضا conditional log من تمرير أوامر أو معاملات معينه إلى plugins مثل تغيير قيمة مسجل مثلا

أوامر التوقف الخاصة بالذاكرة memory تستخدم للتوقف عن استخدام هذا الجزء من الذاكرة أو تعديل محتوياتها على الترتيب

برامج فحص PE

كما يمكن التوقف عند عمل اتصال بأي جزء من مكونات الكمبيوتر بالأمر  
**hardware on execution**  
 : Hit trace

- Add selection
- Add procedure
- Add all recognized procedures

تستخدم هذه الأوامر لوضع نقاط إيقاف عند كل أمر ونقاط التوقف العادية سيتم حذفها ويتم وضع علامة عند كل أمر تم تنفيذه .

:Run trace

- Add selection
  - Add procedure
  - Add branches in procedure
  - Add entries of all procedures
- 
- Skip selection when tracing
  - Set condition Ctrl+T

الاختلاف في هذه الأوامر انه يتم وضع بيان إضافي إلى بيانات التتبع مع عدم حذف نقاط الإيقاف العادية.

Follow in dump: تستخدم لتفريغ السطر الحالي في نافذة dump (السفلى)  
 :View call tree

Called from	Procedure	Calls	Comment
		> Acrobat.004DC100	Leaf
		> Acrobat.004DC1E0	Leaf
		> Acrobat.004DC590	Leaf
		> Acrobat.004DC610	Leaf
		> Acrobat.004DC960	Leaf
		> Acrobat.004DC9E0	Leaf
		> Acrobat.004DC990	Leaf
		> Acrobat.004DD1C0	Leaf
		> Acrobat.004DD100	Leaf
		> Acrobat.004DD334	Leaf
		> Acrobat.004DD590	Leaf
		> Acrobat.004DD990	Leaf
		> Acrobat.004DD96C	Leaf
		Acrobat.004DC695	

تظهر النافذة السابقة وتعرض قائمه لكل الدوال التي تم النداء عليها عن طريق إجراء معين وقد يوجد بالنافذة تعليق يحدد نوع النداء

Leaf لا يتم النداء على أي داله أخرى

Pure لا يتم النداء على أية داله أخرى مع عدم وجود آثار جانبية  
(الإجراء لا ينادى نفسه بنفسه)

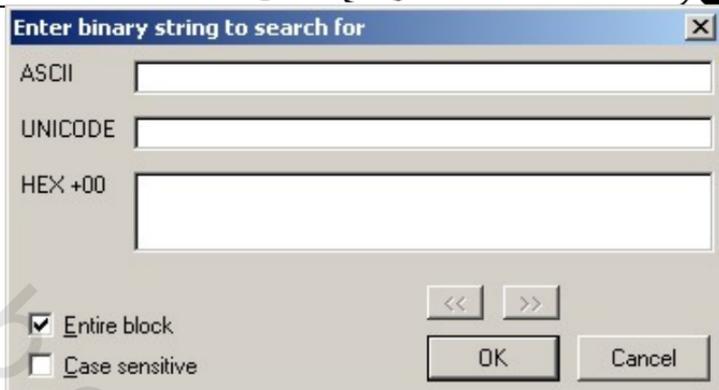
RETN يحتوى على أمر الرجوع

Sys الدالة هي من دوال النظام

:Search for

Name (label) in current module	Ctrl+N
Name in all modules	
Command	Ctrl+F
Sequence of commands	Ctrl+S
Constant	
Binary string	Ctrl+B
All intermodular calls	
All commands	
All sequences	
All constants	
All switches	
All referenced text strings	
User-defined label	
User-defined comment	

المجموعة الأولى للبحث عن معرف في الإجراء الحالي أول كل البرنامج.  
المجموعة الثانية للبحث عن أمر أو مجموعه من الأوامر أو ثابت أو نص أو كود بالنظام السداسي عشر.



المجموعة الثالثة للبحث في فئات محدد مثلا كل الثوابت all constant أو كل النصوص المرجعية all referenced text المجموعة الأخيرة للبحث في التعليقات أو المعرفات التي قمت بإدخالها.

Label	Disassembly	Comment
	PUSH EBX	(Initial CPU selection)

:Find reference to

للبحث عن تعريف الإجراء الذي قام بالنداء على السطر الحالي ويظهر العنوان الخاص بهذا الإجراء

Address	Disassembly	Comment
00401020	PUSH EBX	(Initial CPU selection)

:View

يمكنك عن طريقها اختيار احد الإجراءات المعرفة بالبرنامج والذهاب إليه:

Executable file
Relative address
Module 'ACE'
Module 'AGM'
Module 'BIB'
Module 'CoolType'
Module 'JP2KLib'
Module 'OPP'
Module 'LPK'
Module 'SHLWAPI'
Module 'COMCTL32'
Module 'WINSPOOL'
Module 'oledlg'
Module 'IMM32'
Module 'comdlg32'
Module 'WINMM'
Module 'OLEAUT32'
Module 'ole32'
Module 'SHELL32'
Module 'VERSION'

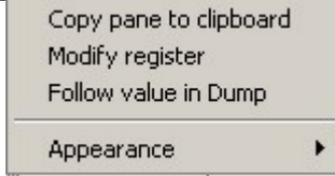
Analyze code: يقوم البرنامج بإجراء تحليل إضافي للكود ومن فوائد هذا الأمر انه يستطيع التفريق بين أجزاء البيانات وأجزاء الكود في الملف والتعرف على أماكن القفز والنصوص والحلقات التكرارية.

Bookmark: لتعليم جزء من الكود والرجوع إليه بسرعة عن طريق أمر goto bookmark

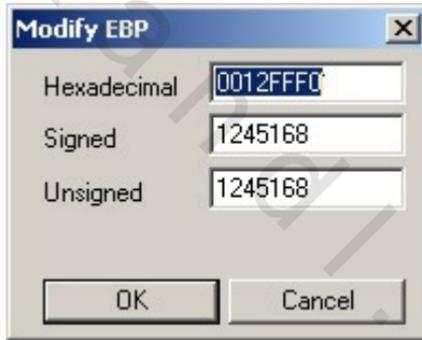
Appearance: للتحكم بشكل النافذة

**نافذة Information:**

## برامج فحص PE

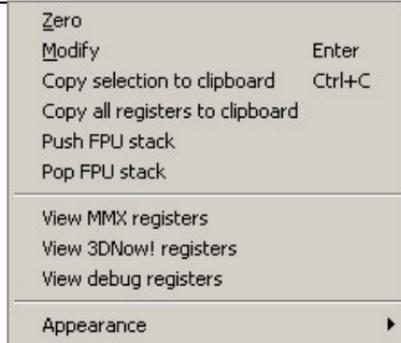


أثناء التوقف على سطر محدد في نافذة disassembly يمكنك رؤية قيمة ومعاملات المسجل الحالي في هذه النافذة وإذا قمت بعمل analyze code يمكنك رؤية الأوامر التي أدت للسطر الحالي وإذا كانت معلومات الديبج متضمنة في الملف يمكنك رؤية سطر الكود المصدر المكافئ للسطر الحالي. ومن هذه النافذة يمكنك تتبع مسجل معين في نافذة disassembler أو dump وتغييرها



## نافذة Registers:

هذه النافذة تحتوي على المسجلات الخاص بالبرنامج ويمكن عن طريق هذه النافذة متابعة التغيير في قيم المسجلات وتغيير قائمة الأوامر المختصرة بتغيير مكان الوقوف.



من شريط الحالة في أعلى النافذة يمكنك أن ترى أن المسجلات الافتراضية هي FPU ويمكن أيضا رؤية تعليمات ومسجلات MMX و 3Dnow الخاص بمعالجات AMD عن طريق الضغط مره واحد على شريط الحالة أو باختيار نوع المعالج من القائمة المختصرة.

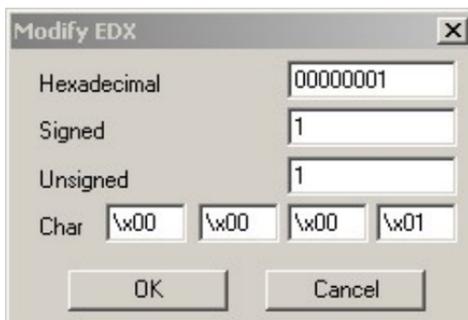
Zero: يمكنك من تصفير قيمة مسجل

Increment: لزيادة قيمة المسجل بواحد والعكس decrement

Set to 1: لوضع القيمة 1 للمسجل

### نافذة Modify:

يمكنك عن طريق هذه النافذة تغيير قيم صحيحة بالنظام السداسي عشر أو قيم كسريه



## برامج فحص PE



## نافذة Dump:

يمكن تغيير شكل عرض البيانات في هذه النافذة إلى عدة أشكال مثل نصوص ASCII وأرقام وتفرغ للاسمبلى والشكل الافتراضي هو السداسي عشر.



كما يمكنك رؤية كود نقطة الدخول في رأس الملف عن طريق الأمر | special PE header

Address	Hex dump	Data	Comment
00ACAA00	00	DB 00	
00ACAA01	00	DB 00	
00ACAA02	00	DB 00	
00ACAA03	00	DB 00	
00ACAA04	40	DB 40	
00ACAA05	C9	DB C9	
00ACAA06	98	DB 98	
00ACAA07	00	DB 00	
00ACAA08	6F	DB 6F	
00ACAA09	C9	DB C9	
00ACAA0A	98	DB 98	
00ACAA0B	00	DB 00	
00ACAA0C	A1	DB A1	
00ACAA0D	C9	DB C9	

### نافذة Stack:

هذه النافذة تظهر محتويات الذاكرة stack للتشغيل الحالي للبرنامج ويتم دائما الوقوف عند بداية الديويج عن المسجل ESP الذي يعتبر هو المؤشر للذاكرة stack من نوع 32 بيت .

كما تقوم هذه النافذة بتتبع معرفات هيكل الأخطاء أو SHE ويعتبر [FS:0] هو بداية هذا الهيكل وتحتوى هذه النافذة على العديد من الأوامر المختصرة يمكن تلخيصها فيما يلي:

Address	
Show ASCII dump	
Show UNICODE dump	
Lock stack	
Copy to clipboard	Ctrl+C
Modify	
Edit	Ctrl+E
Push DWORD	
Pop DWORD	
Search for address	
Search for binary string	Ctrl+B
Go to ESP	*
Go to EBP	
Go to expression	Ctrl+G
Follow in Dump	
Follow in Stack	Enter
Appearance	

:Address

## برامج فحص PE

Relative to selection  
Relative to ESP  
Relative to EBP

يمكنك رؤية العنوان الذي له علاقة بالاختيار الحالي أو بالمسجل ESP أو المسجل EBP في حالة المسجلات تتغير العناوين بتغير المسجلات  
Show ASCII dump: يقوم بإظهار عمود إضافي يحتوى على تفريغ نصي  
Unicode Dump: يظهر عمود إضافي يظهر نص قياسي من نوع Unicode  
Lock stack: يقوم بإيقاف الانتقال التلقائي إلى المسجل ESP في كل مرة يتم فيها إيقاف البرنامج.

Modify: يمكنك من تعديل أول رقم من نوع doubleword



Edit: يمكنك من تعديل المحتويات بالنظام السداسي عشر أو نصوص ASCII أو Unicode



Push DWORD: إنقاص المسجل ESP بمقدار 4

Pop DWORD: زيادة المسجل ESP بمقدار 4

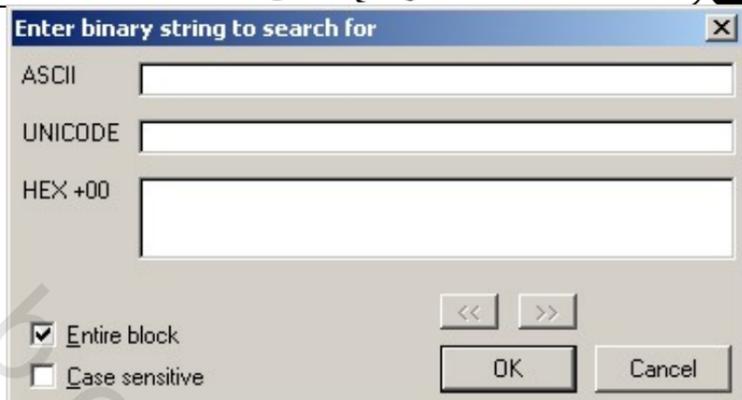
Search for address: يمكنك من البحث عن عنوان بالنظام السداسي عشر



Search for binary string: يمكنك من البحث عن نص باستخدام تشكيلات

البحث القياسية مثل:

12 ?? ?6 78



goto ESP: للذهاب إلى محتويات المسجل ESP

Goto EBP: للذهاب إلى محتويات المسجل EBP

Follow in Disassembler: لتتبع أول رقم DWORD مختار في نافذة disassembler

و هكذا أمر follow in dump للتتبع في نافذة dump و follow in stack للتتبع في نافذة stack

و بذلك نكون انتهينا من نافذة CPU وهي أهم نافذة ببرنامج OllyDBG

### نافذة Patches:

Address	Size	State	Old	New	Comment
0048F514	2	Hot Live	JE SHORT Acrobat.0048F569	JNZ SHORT Acrobat.0048F569	

إثناء إجراء تعديلات في تعليمات الاسمبلى يقوم برنامج Ollydbg بتخزين هذه التعديلات وعرضها في نافذة patches التي تتيح حذف أو إبطال عدد من التعديلات

فمثلا إذا قمت في نافذة Dissassembler بتغيير يمكنك أن تراه في نافذة Patches عن طريق الأوامر View Patches وعن طريق قائمة الأوامر المختصرة يمكنك التعامل مع هذا التعديل:

Actualize	
Follow in Disassembler	Enter
Restore original code	Space
Delete record	Del
Delete all records in module Acrobat	
Toggle breakpoint	F2
Conditional breakpoint	Shift+F2
Conditional log breakpoint	Shift+F4
Copy to clipboard	▶
Appearance	▶

Follow in disassembler : لتتبع التعديل في نافذة disassembler  
 Restore original code : لإلغاء التعديل واستعادة الكود الأصلي  
 Delete record : لحذف كل القيم بنافذة patches  
 Toggole breakpoint : لوضع أو حذف نقطة إيقاف  
 Conditional breakpoint : لوضع نقطة إيقاف شرطيه  
 Copy to clipboard : للنسخ إلى الذاكرة  
 Appearance : للتحكم بشكل النافذة  
**النافذة Call Stack**

Address	Stack	Procedure / arguments	Called from

تقوم هذا النافذة بإظهار النداءات التي تتم في الذاكرة stack ويمكنك معرفة العنوان الذي تم منه النداء عن طريق العمود address و العمود stack يقوم بإظهار القيمة التي تم إرجاعها من العنوان أو المعامل المستخدم

أما العمود procedure فيظهر عنوان الدالة التي تم النداء عليها وأحيانا لا يكون ollydbg متأكد من العنوان لذلك يظهر احد القيم التالية:

نقطة الدخول لا يعتمد عليها ?

العنوان تم تخمينه لعدم التوصل لنقطة دخول Maybe

لا يمكن الوصول إلى نقطة دخول ولكن تم إيجاد العناوين الظاهرة Includes بالداله

Called from: هو العنوان الخاص بالأمر الذي قام بالنداء على هذا الدالة

### النافذة Breakpints:

Address	Module	Active	Disassembly
0048F514	Acrobat	Always	JE SHORT Acrobat.0048F569

يمكننا عن طريق هذه النافذة مشاهدة كل نقاط الإيقاف المستخدمة والذهاب إليها أو إيقافها أو حتى إلغاؤها تماما .

العمود address يحتوي العنوان الخاص بنقطة الإيقاف ويليه اسم البرنامج ثم نوع الحالة التي عليها نقطة الإيقاف وأخير الأمر الذي يتم التوقف عنده.



Remove: لحذف نقطة الإيقاف من النافذة وإلغاؤها تماما

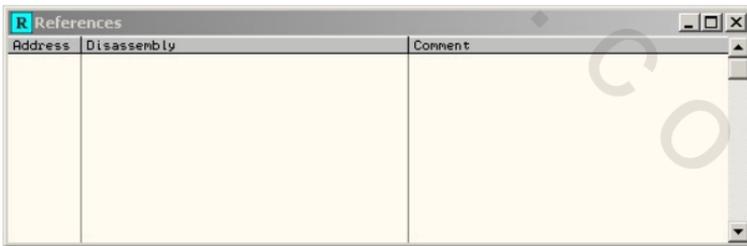
Disable: يقوم برنامج ollydbg بتجاهل نقطة الإيقاف

Edit condition: للتعديل في شروط نقطة التوقف

Follow in disassembler: لتتبع نقطة الإيقاف في نافذة disassembler

Disable all: لإبطال جميع نقاط التوقف

### النافذة References:



تتيح لك هذه النافذة التنقل بسرعة بين العناوين والنصوص في نافذة

disassembler

النافذة Run Trace

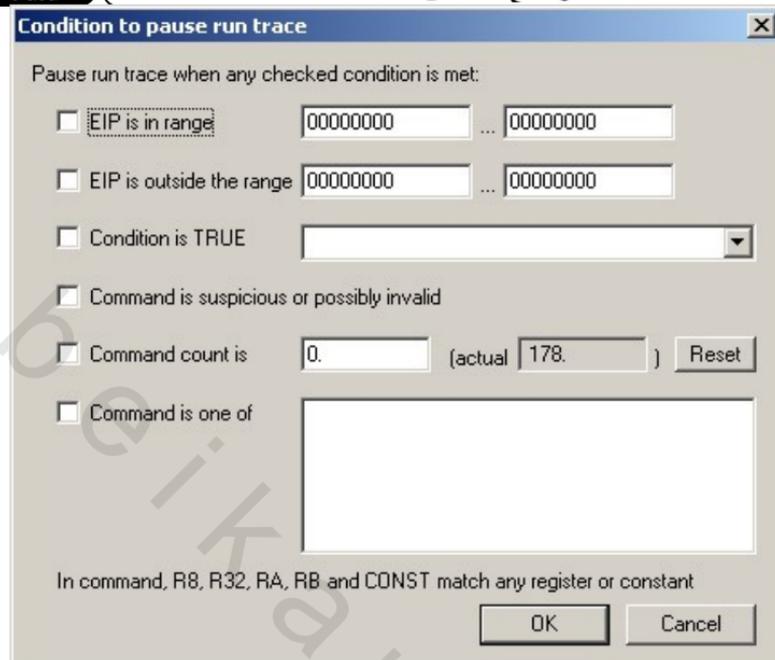
## برامج فحص PE

Back	Thread	Module	Address	Command
178.	Main	Acrobat	00401000	PUSH EBP
177.	Main	Acrobat	00401001	MOV EBP,ESP
176.	Main	Acrobat	00401003	PUSH -1
175.	Main	Acrobat	00401005	PUSH Acrobat.009F66C0
174.	Main	Acrobat	0040100A	PUSH Acrobat.004E05AC
173.	Main	Acrobat	0040100F	MOV EAX,DWORD PTR FS:[0]
172.	Main	Acrobat	00401015	PUSH EAX
171.	Main	Acrobat	00401016	MOV DWORD PTR FS:[0],ESP
170.	Main	Acrobat	0040101D	SUB ESP,58
169.	Main	Acrobat	00401020	PUSH EBX
168.	Main	Acrobat	00401021	PUSH ESI
167.	Main	Acrobat	00401022	PUSH EDI
166.	Main	Acrobat	00401023	MOV DWORD PTR SS:[EBP-18],ESP
165.	Main	Acrobat	00401026	CALL DWORD PTR DS:[<&KERNEL32.G
164.	Main	kernel32	77E7D142	MOV EAX,DWORD PTR FS:[18]

يمكن عن طريق هذه النافذة تتبع عمليات البرنامج قبل حدوث بعض الأحداث المعينة فبرنامج ollydbg يقوم بالتتبع داخل البرامج خطوه بخطوه ولكنه لا يقوم بتحديث نوافذه أول بأول لذلك يمكننا عن طريق هذا النافذة من مشاهدة كل التغييرات التي تحدث.

يمكنك البدء في التتبع بالضغط على المفاتيح ctrl+f11 وهذا النوع من التتبع يقوم بالدخول في كل إجراء فرعى يجده أو الضغط على المفاتيح ctrl+f12 وهذا النوع لا يقوم بالدخول في الإجراءات الفرعية.

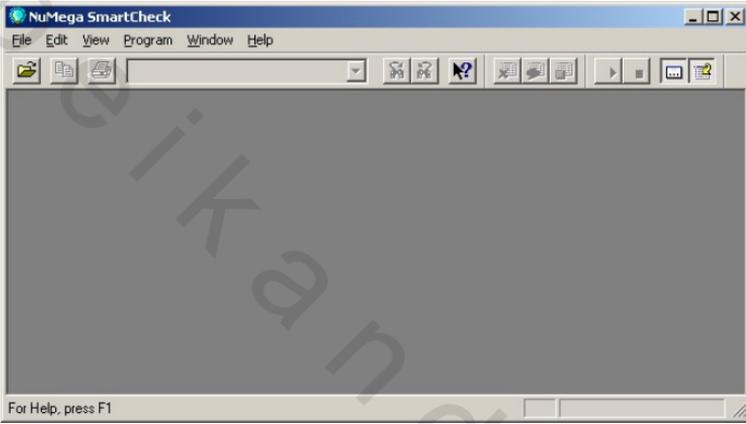
يمكنك أيضا وضع شرط للتوقف ويتم ذلك بالضغط على المفاتيح ctrl+t وتظهر النافذة التالية:



و يمكنك عن طريق النافذة السابقة تحديد التوقف مثلا عندما يصل المسجل EIP إلى مدى محدد من العناوين أو خارج هذه العناوين أو عندما يتحقق شرط معين وهكذا.

## برنامج SmartCheck :

هذا البرنامج من أفضل برامج الـ debugger والتي تلي ollydbg في القوة ويمكن عن طريقها اختبار البرامج المكتوبة بلغة فيجوال بيزيك .  
تحتوى النافذة الافتتاحية للبرنامج على قوائم ومفاتيح اختصار يمكن عن طريقها تتبع البرنامج بسرعة وبسهولة.



و يمكنك عن طريق هذا البرنامج اختبار استخدام الفيجوال بيزيك لداله من دوال API وما إذا كانت المعاملات الخاصة بالدالة صحيحة حتى تعمل دالة API وتعطى النتيجة المطلوبة

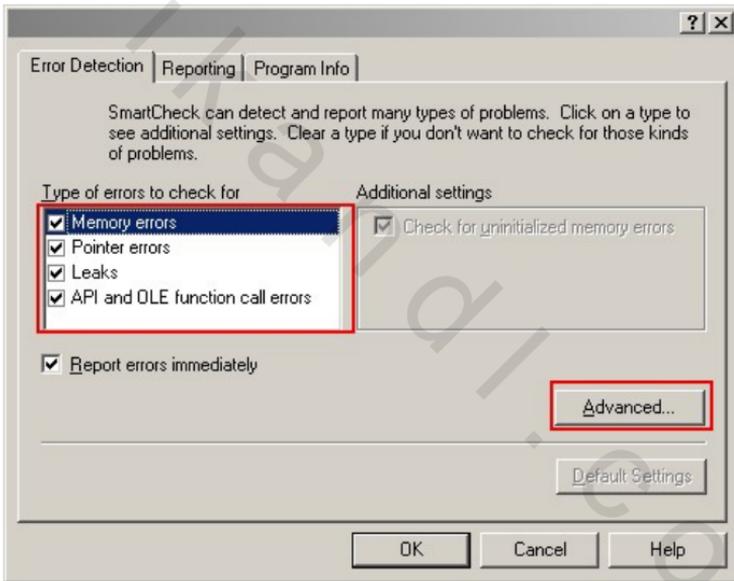
افتراضيا يقوم البرنامج بعمل التالي:

- ✦ اكتشاف الأخطاء الناتجة عند إحداث معينه للبرامج المكتوبة بلغة فيجوال بيزيك
- ✦ تقديم تحليل كامل للأخطاء الناتجة أثناء وقت التنفيذ وغير قابله للتعامل معها من خلال فيجوال بيزيك
- ✦ يقوم بتقديم تقرير فوري للأخطاء عند حدوثها وتقديم معلومات كاملة عنها حتى يمكنك تحديد الأسلوب المناسب للتعامل معها

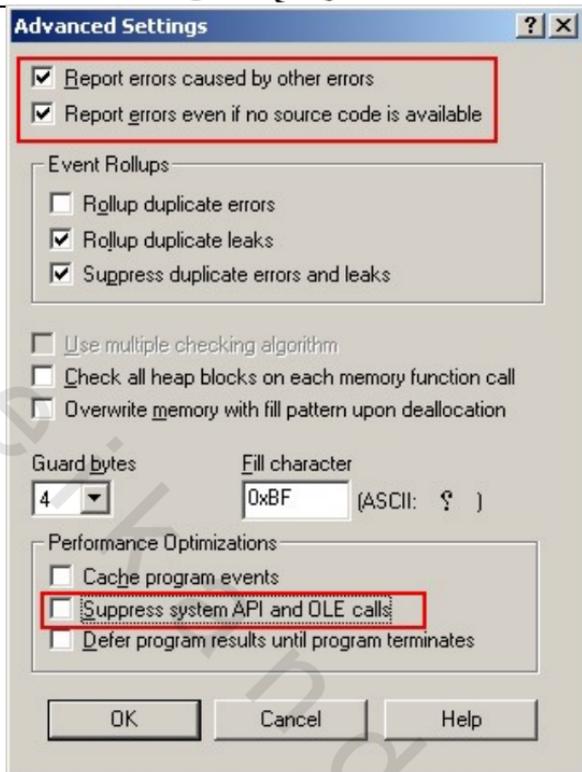
✦ يقوم بفتح نافذة result عند الانتهاء والتي تحتوى على جميع الأحداث والأخطاء التي وقعت

ولكي يخدم البرنامج أغراضنا (الجيدة) لاكتشاف الأخطاء يجب تغيير بعض الاختيارات كما يلي:

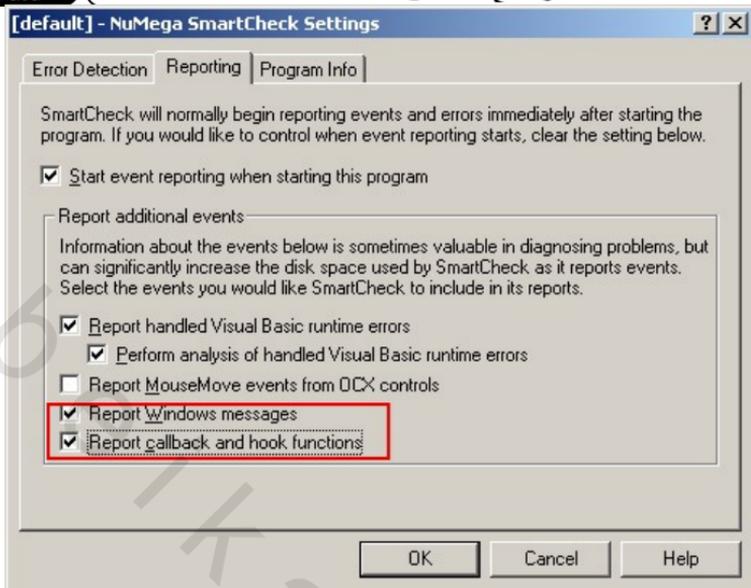
1. قم بفتح نافذة settings بالضغط على الأوامر | Program :settings



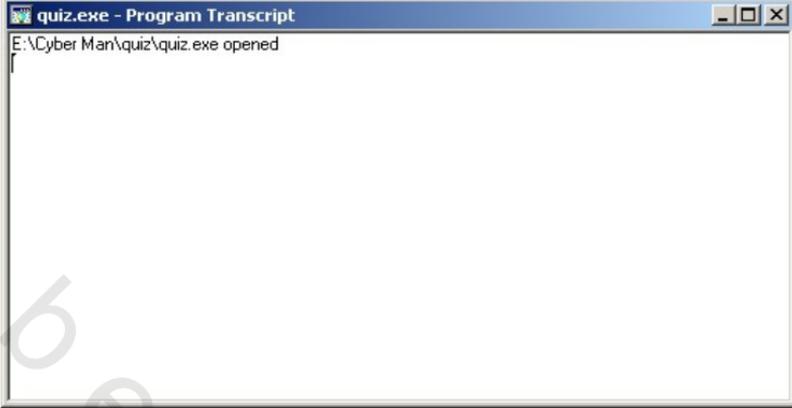
كما ترى الاختيارات الافتراضية للبرنامج هي اكتشاف جميع أنواع الأخطاء قم بالضغط على المفتاح :advanced



2. قم بالتأكد من اختيار `report errors caused by other errors` والاختيار `report error even if no source code` حتى يقوم البرنامج باكتشاف الأخطاء للبرامج التي نريد كسر حمايتها فلن يكون متاح لنا بالطبع الكود الخاص بتلك البرامج
3. تأكد أيضا من إزالة الاختيار `suppress system API calls` حتى لا يتجاهل البرنامج النداء على دوال API التي سنقوم باختبارها ومعرفة مكان الحماية
4. قم بالضغط على OK ثم من النافذة السابقة اختر الصفحة Reporting



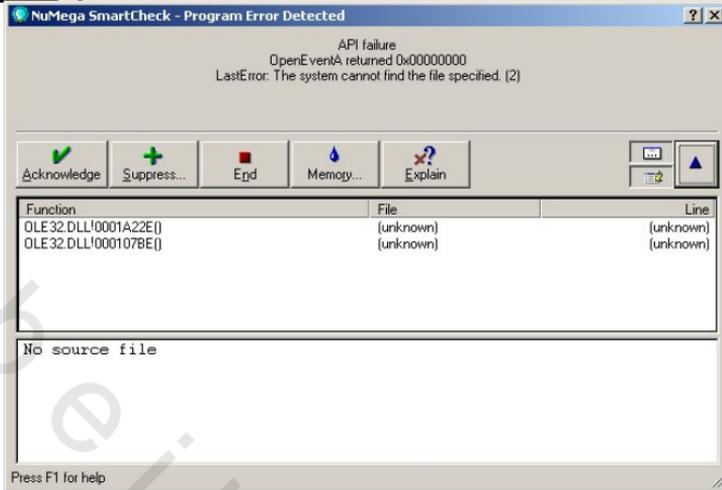
5. تأكد من اختيار الاختيارين report windows messages حتى يقوم البرنامج بتتبع رسائل الويندوز الناتجة عن النداء لبعض الدوال وأيضا الاختيار report callback and hook functions حتى يمكننا تتبع جميع أنواع الدوال
6. لكي تقوم بتتبع برنامج اختار الأوامر File | Open أو المفتاح المختصر  ثم قم باختيار الملف التنفيذي المراد تتبعه وبعد أن يظهر البرنامج رسالة تفيد انتهاء فتح الملف اضغط على مفتاح بداية التنفيذ أو المفتاح F5 أو الأوامر Program | Start.



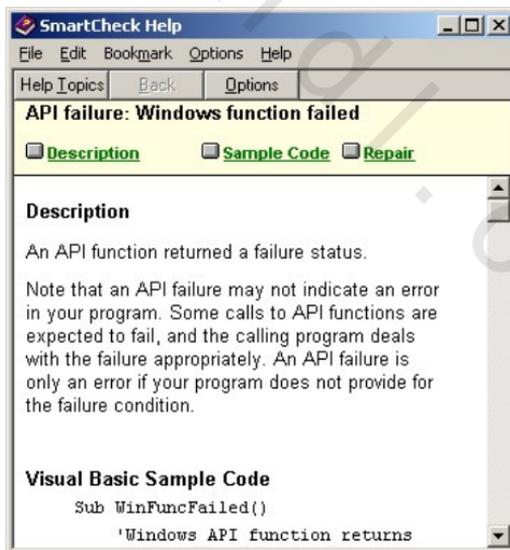
يقوم البرنامج الآن بفتح النافذة result وإذا حدث خطأ يقوم البرنامج بإظهار الرسالة التالية:



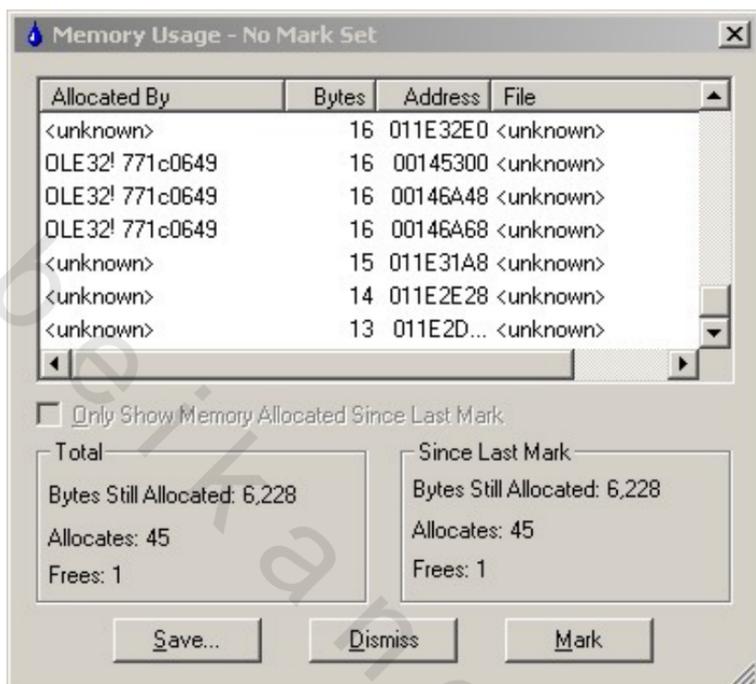
و يمكنك معرفة اسم الدالة المسببة للخطأ ورقم السطر إذا كان الكود الأصلي للبرنامج متوفر عن طريق الضغط على المفتاح إلى يوجد به سهم لأسفل.



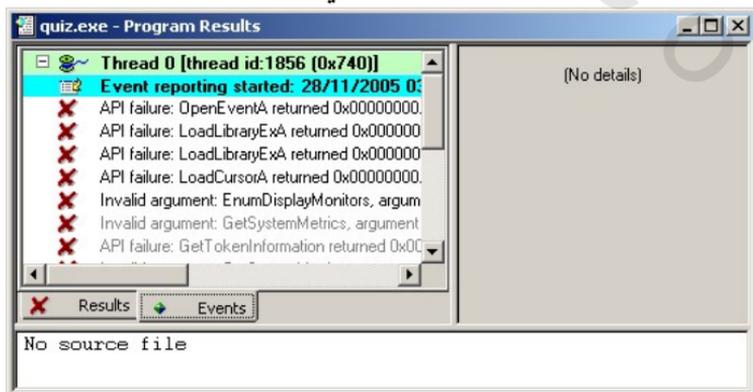
و يمكنك أيضا الحصول على شرح من ملفات المساعدة عن طريق المفتاح explain



أما المفتاح memory فيمكنك من رؤية الملفات المحملة بالذاكرة عند هذه النقطة وعنوان كل داله تم النداء عليها من الملف



أما إذا أردت تجاهل الخطأ وما يماثله فقم باختيار `suppress` أو إنهاء الاختبار  
 قم باختيار `end` أو متابعه تنفيذ الملف بالمفتاح `acknowledge`  
 و في نهاية العملية يمكنك مشاهدة النتيجة في نافذة `result`:



بذلك نكون انتهينا من أهم برامج الديبجر وسننتقل الآن إلى برامج إعادة الهندسة وهي من الأدوات الأساسية أيضا لفك حماية معظم البرامج