الفصل الثالث

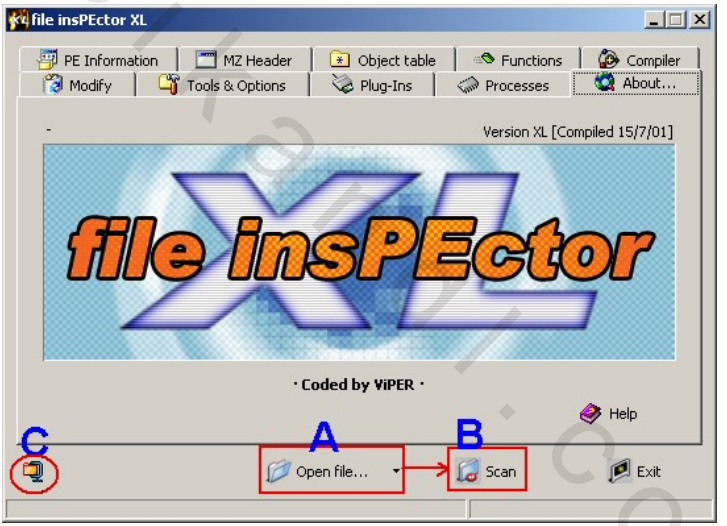
برامج فحص PE

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:

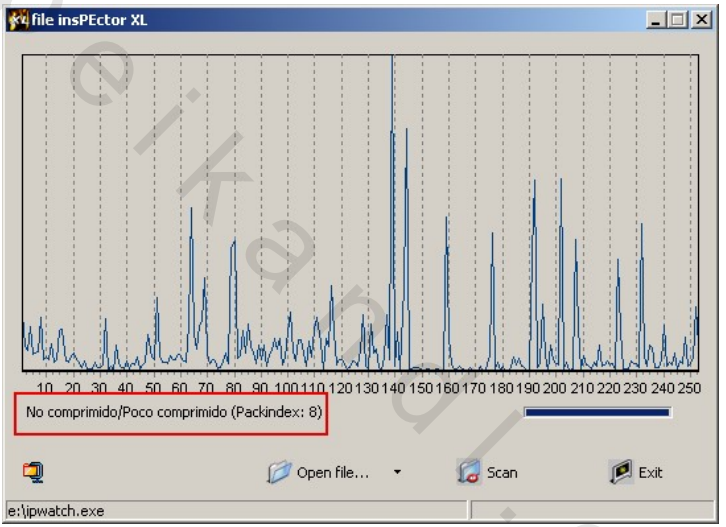


ولكي تقوم باستخدام البرنامج اتبع الخطوات التالية:

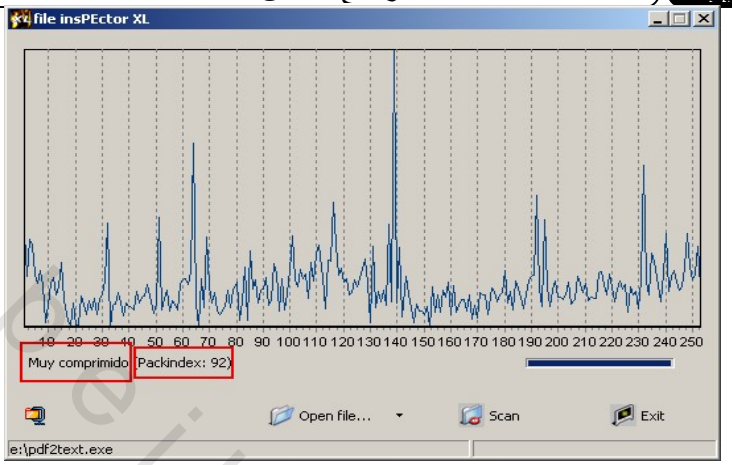
1. الأمر Open وهو يستخدم لاختيار الملف المراد فتحه ومعرفة معلومات عنه.
2. الأمر Scan حتى يبدأ البرنامج فعليا في فحص الملف وإظهار جميع المعلومات عنه.

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

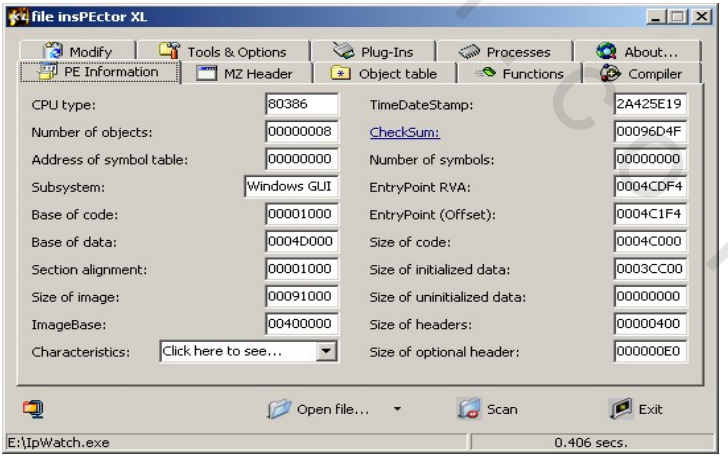
إذا ضغطت الآن على المفتاح C سترى الشكل التالي (بعد تنفيذ الأمر open بالطبع)



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:

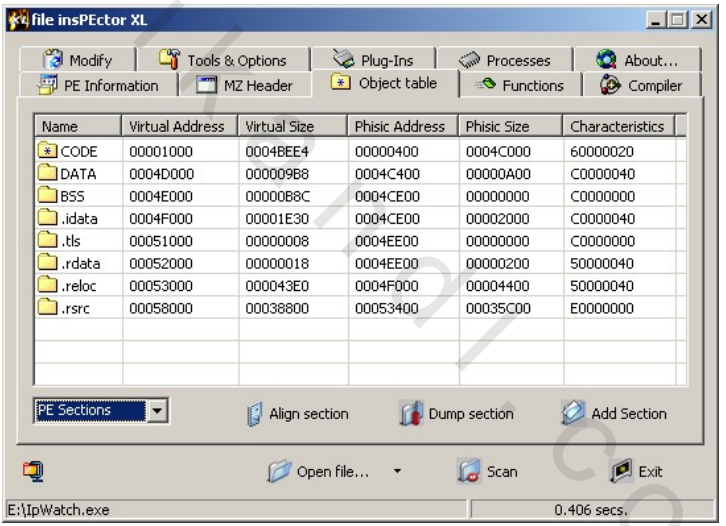


يظهر البرنامج أن الملف مضغوط ضغطا عاليا بنسبة 92 في المائة
إذا ضغطت مره أخرى على المفتاح إظهار الضغط ستقوم بالرجوع إلى الشاشة الرئيسية قم الآن بالضغط على مفتاح scan إذا لم تقم بذلك فيقوم البرنامج بإظهار أول صفحه وهي صفحة معلومات نقطة الدخول للبرنامج أو PE info كما بالشكل:

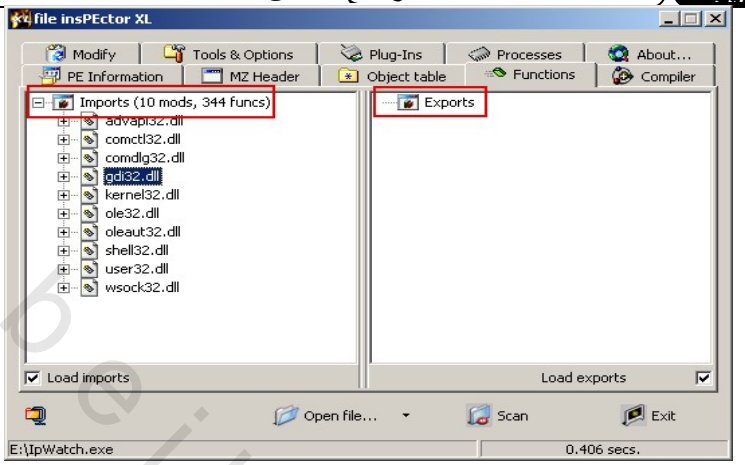


يظهر البرنامج في هذه الصفحة رقم checksum الذي يستخدم للتأكد من الحجم الصحيح للملف وانه لم يتم تغييره وعنوان نقطة الدخول EP offset وغيره من المعلومات المفيدة.

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

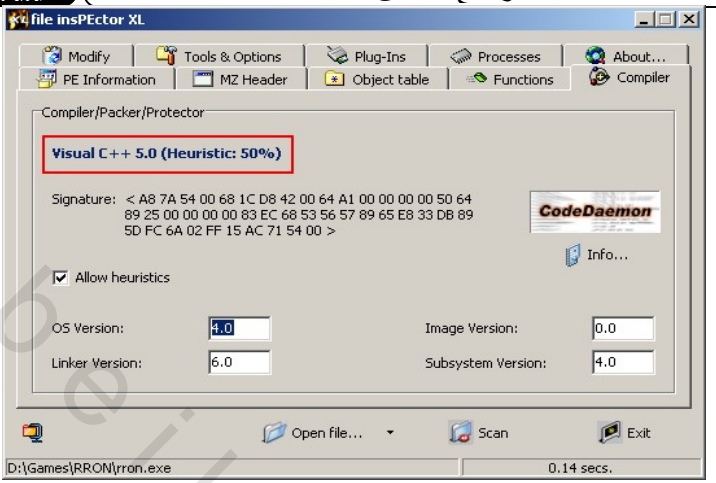


الصفحة التي تليها وهي صفحة Functions وتحتوى دوال API التي يقوم الملف باستخدامها سواء كانت دوال API الخاصة بالويندوز أو دوال خاصة بالملف

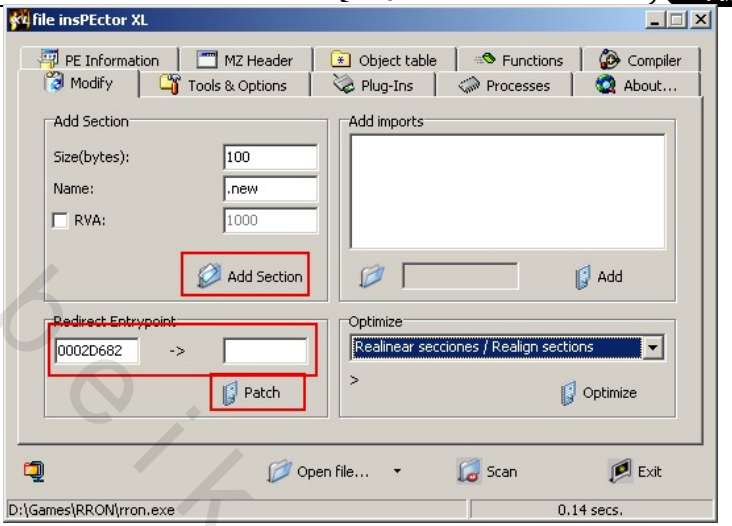


كما يظهر البرنامج أيضا ملف dll أو المكتبة الخاصة بالويندوز المسئولة عن تنفيذ دالة API مثل الملف GDI32.dll مثلا نجد به دوال التعامل مع رسوميات الملف.

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

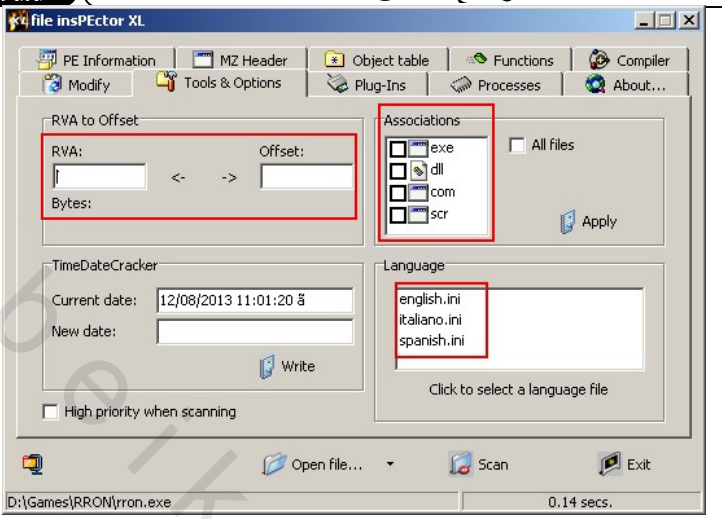


الصفحة التالية وهي صفحة Modify وتمكننا من التعديل في الملف فمثلا إذا أردنا أن نضيف جزء جديد إلى الملف وبه بيانات خاصة بنا نريد أن يقوم الملف بتحميلها قبل أن يستمر في تنفيذ باقي التعليمات الخاصة به فيمكن ذلك عن طريق تحديد EP الجديدة والضغط على مفتاح patch كما يمكنك أن ترى من الشكل:

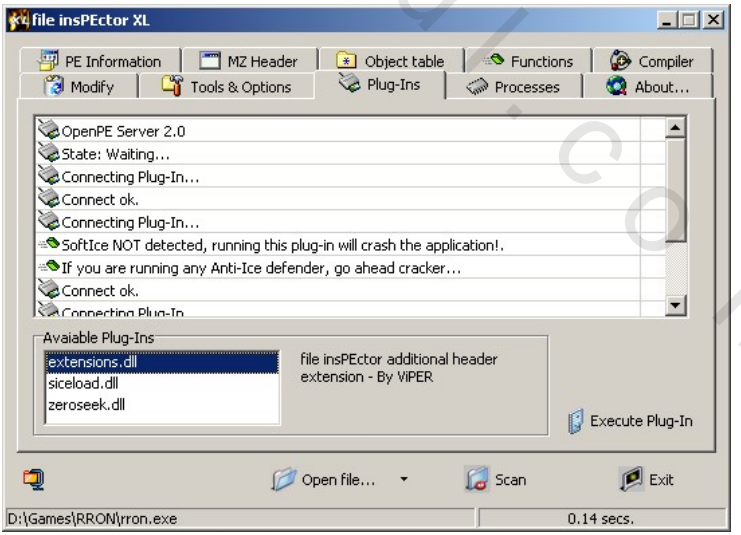


الصفحة التالية هي صفة الأدوات والاختيارات ويمكن عن طريقها تحويل عنوان من نوع RVA أي عنوان تخيلي إلى عنوان حقيقي وهو offset

كما تمكننا 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 ☺



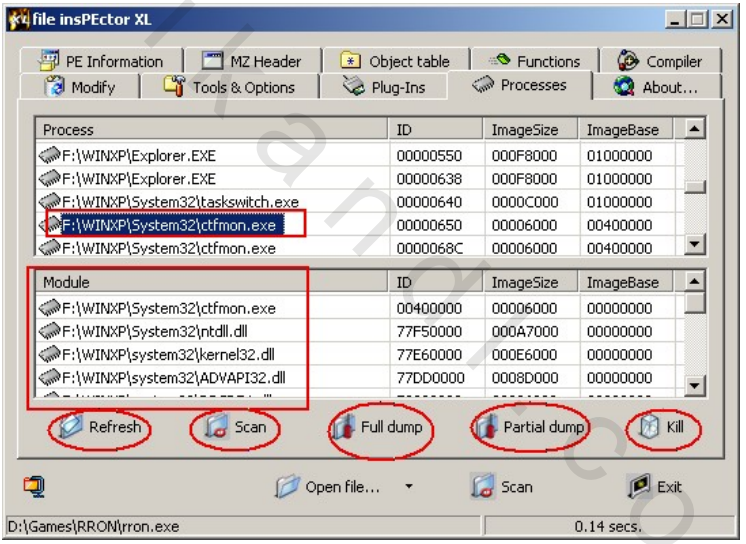
و هناك صفحة أخرى مخصصة للتعامل مع plugins والتي تزيد من الإمكانيات العادية للبرنامج مثل إدراج خاصية التحميل مع برنامج soft-ice خصوصا للبرامج المشفرة أو المضغوطة كما يمكننا من البحث داخل أي جزء من أجزاء الملف وتغيير البيانات به



أما الصفحة process فتحتوى 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 يحتوى الجزء العلوي من هذه الشاشة على البرامج المحملة بالذاكرة فإذا اخترت واحدا منها يظهر في الجزء السفلى الملفات أو المكتبات التي يعتمد عليها البرنامج المختار

كما يمكن عمل scan لبرنامج محمل بالذاكرة بدلا من اختياره بالأمر open.



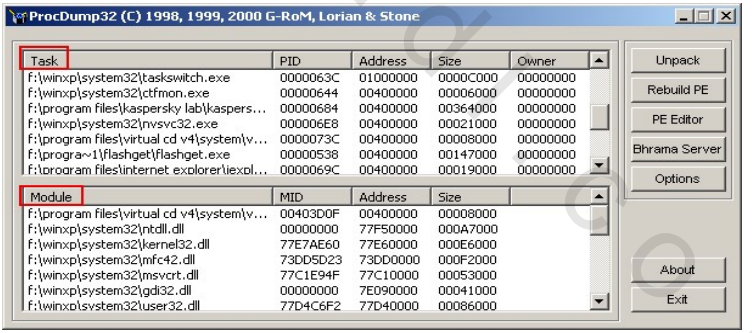
فائدة أوامر ال dump هنا هو إذا كان الملف مضغوطا يمكننا أن نقوم باستخراج الملف الحقيقي التنفيذي عن طريق هذه الأوامر كما يمكن تنفيذ ذلك بالعديد من البرامج المستقلة مثل procdump .

برنامج ProcDump 32:

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 بصوره طبيعيه أما إذا لم يجد CD فينادى المقطع على كود الخروج من البرنامج ولا يتم الذهاب أبدا إلى المقطع text.

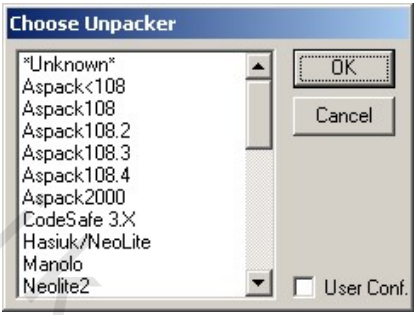
أما الملفات المضغوطة فعادة يتم فيها وضع ملف exe الحقيقي المسئول عن تشغيل البرنامج داخل ملف تنفيذي آخر بحيث يستحيل على الكراكر فك الملف ببرامج إعادة الهندسة لذلك يجب أولا استخراج الملف الحقيقي للبرنامج مثل حماية SecurRom , Safe Disk وغيرها.

الصورة الرئيسية للبرنامج:

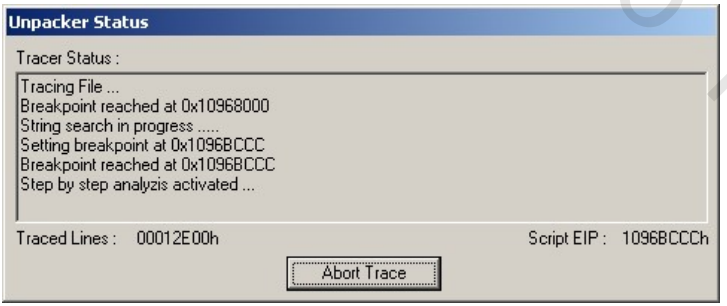


من الشكل السابق يتضح أن البرنامج يقوم بمسح الذاكرة من جميع البرامج المحملة بها ويقوم بعرض كل برنامج في القائمة task والملفات التي يقوم باستخدامها في القائمة module .

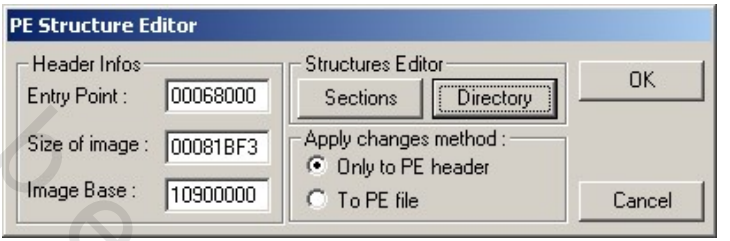
يمكننا البرنامج من التعامل مع الملفات التنفيذية بعدة طرق فالمفتاح الأول Unpack يعطينا إمكانية استخراج الملف التنفيذي الحقيقي وعند الضغط عليه يقوم البرنامج بإظهار ديالوج لاختيار نوع التشفير المستخدم :



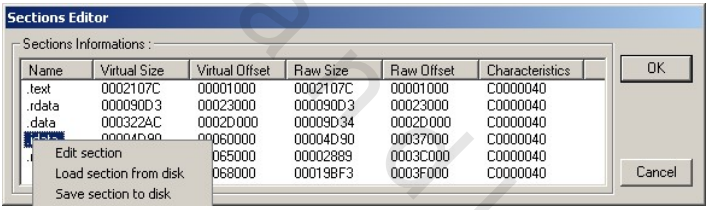
ومن أشهر الطرق التشفير استخدام النوع aspack ونرى أن البرنامج يحتوى على العديد من هذه الطرق قم الآن باختيار احد التشفيرات واضغط ok فنرى ديالوج لاختيار الملف المراد فكه قم باختيار الملف التنفيذي ثم تظهر رسالة تفيد الضغط على مفتاح ok إذا تم تحميل البرنامج بصوره صحيحة ثم يبدأ البرنامج في فحص الملف واختبار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



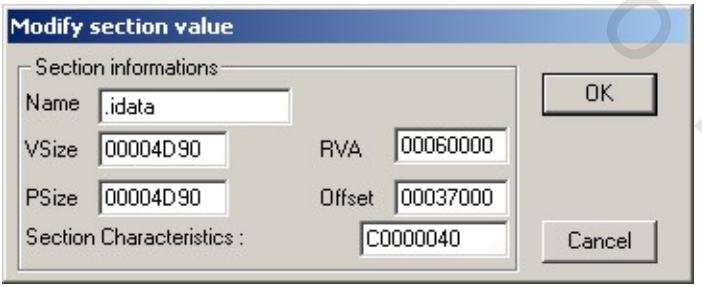
و عن طريق مفتاح PE Editor يمكننا تعديل نقطة الدخول PE الخاصة بالملف إلى نقطه دخول أخرى



كما يمكن تعديل مقاطع الملف عن طريق المفتاح Sections

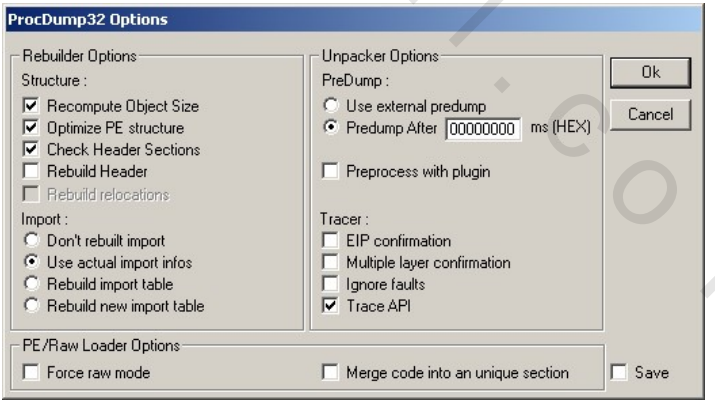


و يمكننا عن طريق قائمة الأوامر المختصرة تعديل مقطع في الملف أو إزالته تماما



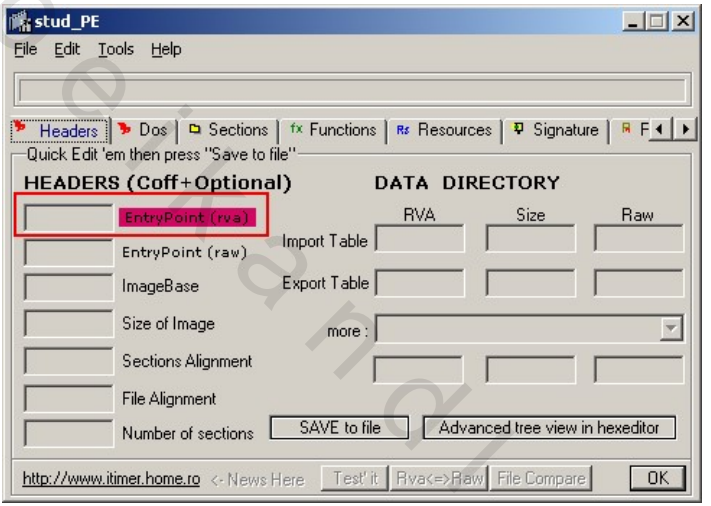
و إذا حدث خطأ يمكنك إعادة بناء الملف عن طريق المفتاح PE Rebuild .
أما المفتاح Bharma Server فيقوم بتحميل برنامج صغير في الذاكرة لاستخدامه مع اسطوانات الألعاب المحمية بطريقة securRom

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

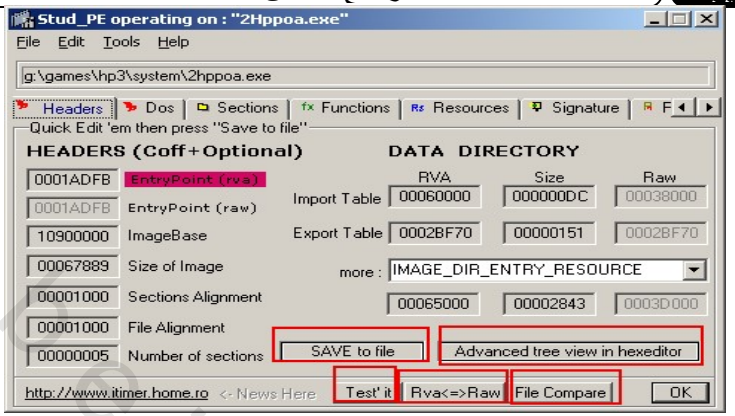


برنامج Stud_PE:

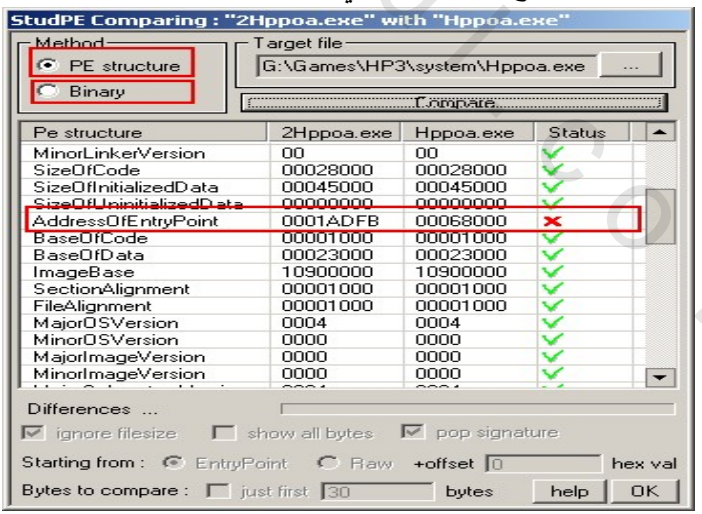
هذا البرنامج الصغير متميز جدا لاحتوائه على عدة إمكانيات توجد كل منها في برنامج منفصل بحد ذاته منها معرفة اللغة المستخدمة في البرنامج ومعرفة نوع الضغط أو التشفير الذي قد يكون مستخدم وعند فتح هذا البرنامج نرى الشاشة الافتتاحية التالية:



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

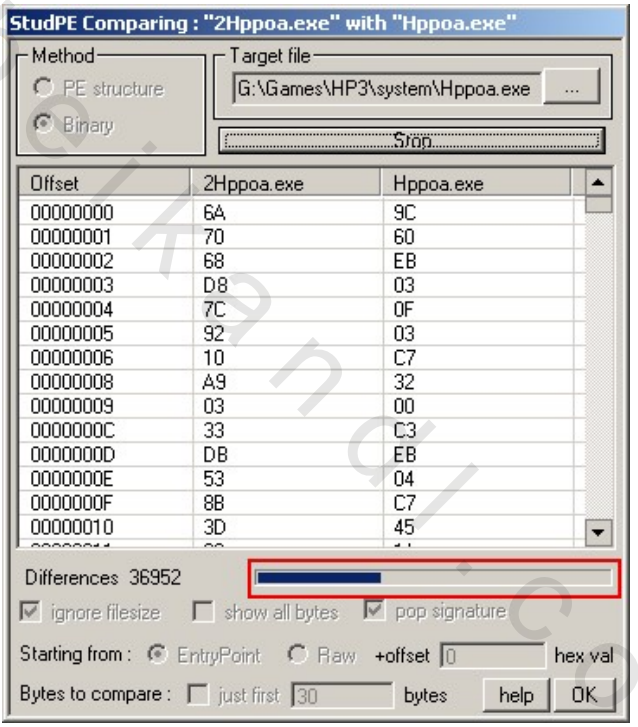


و كما ترى فمثلا هذا الملف مكون من 5 مقاطع ونقطة الدخول هي 1ADFB ويعطينا البرنامج الإمكانية للتحويل من العنوان التخيلي إلى العنوان الحقيقي عن طريق المفتاح Raw <-> Rva كما يمكنك أيضا من مقارنة ملف بآخر مقارنه على أساس هيكل PE أو على أساس ثنائي عن طريق المفتاح File Compare كما يتضح من الشكل التالي

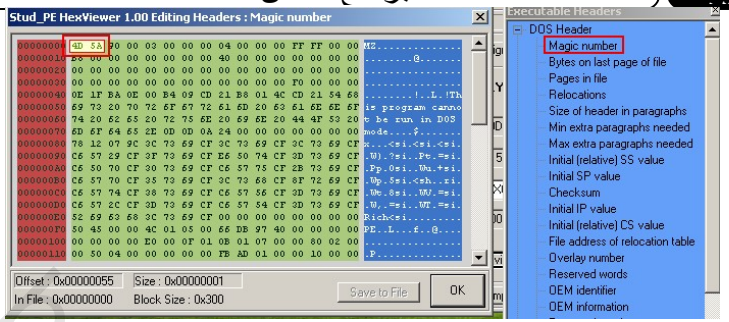


وكما ذكرنا يمكنك اختيار نوع المقارنة والضغط على المفتاح compare ويمكنك أن تشاهد علامة X أما التحاليل المختلفة مثل تغيير نقطة الدخول للبرنامج كما يتضح من الشكل السابق

أما المقارنة الثنائية فهي تأخذ بعض الوقت ولكنها أدق في التفاصيل .

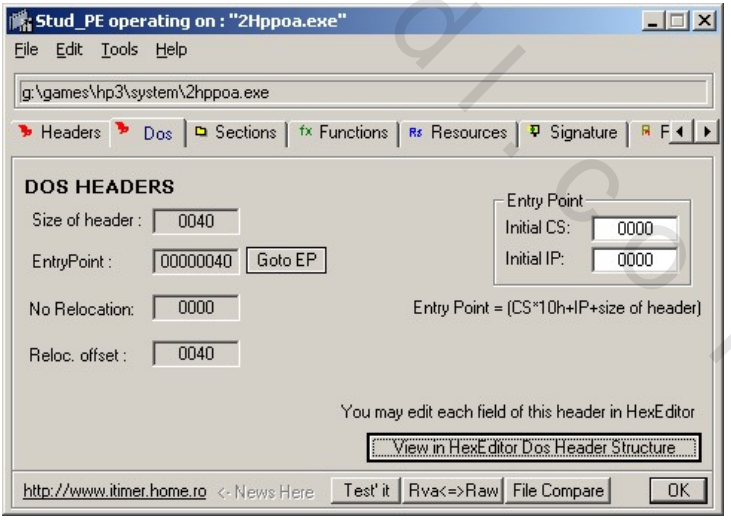


أما المفتاح Advanced tree view فيمكنك من مشاهدة الملف في صورة النظام السداسي عشر بجانب شجره توضح أهم عنوانين الملف مثل عنوان بداية احد مقاطع الملف

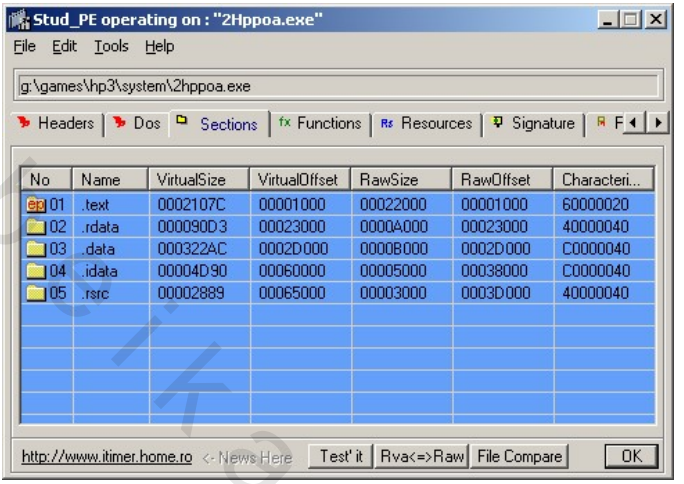


و يمكنك تغيير أي بيانات في هذه الشاشة أو في الشاشة الافتتاحية وحفظ التغييرات فورا عن طريق المفتاح Save to file

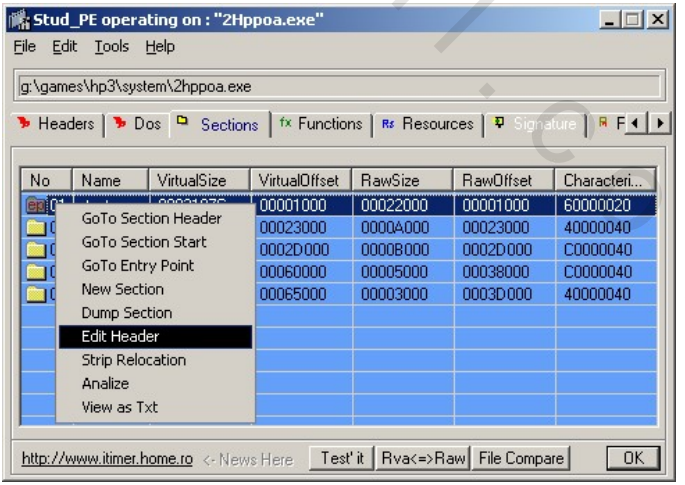
الصفحة التالية هي Dos وتحتوى على معلومات هامه عن بداية المقطع الذي يقرؤه الدوس ويمكنك أيضا مشاهدة هذه المقاطع بالصورة السداسية عشر عن طريق مفتاح view in hex editor dos header structure



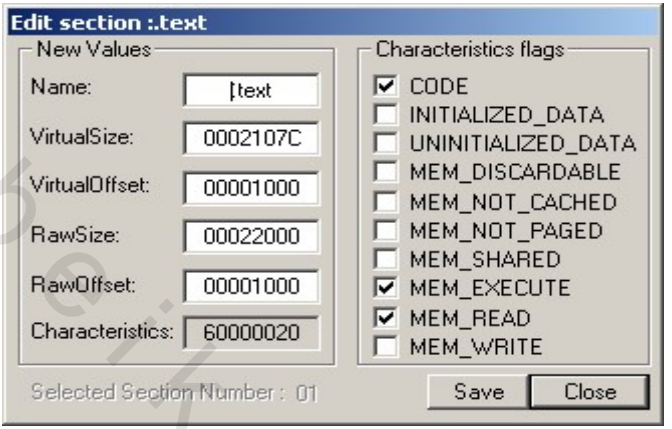
أما الصفحة sections فتحتوى على المقاطع الخاصة بالملف وتحديد أي مقطع هو ep أو نقطة البداية.



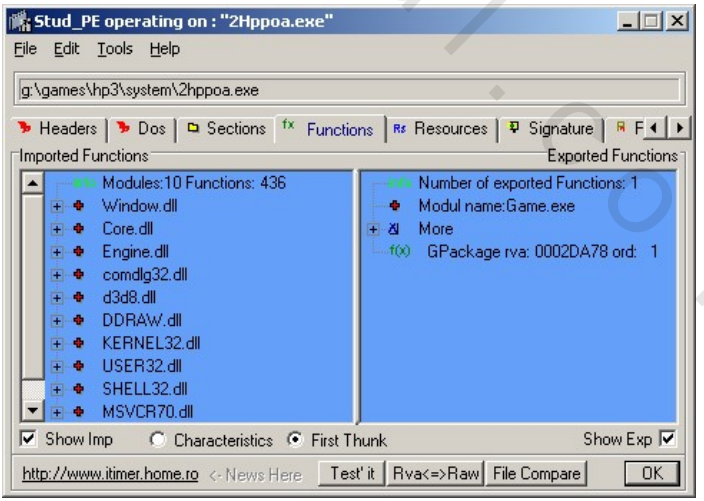
ويمكنك بالطبع اختيار احد المقاطع والتعديل فيها عن طريق أوامر القوائم المختصرة.



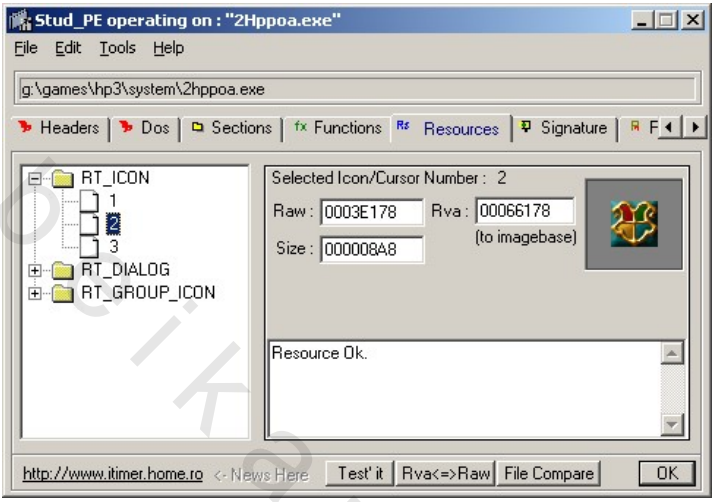
فمثلا إذا اخترت الأمر Edit Header يمكنك تغيير بايتات الخاصة برأس الملف مثل اسم المقطع والعلامات الخاصة به



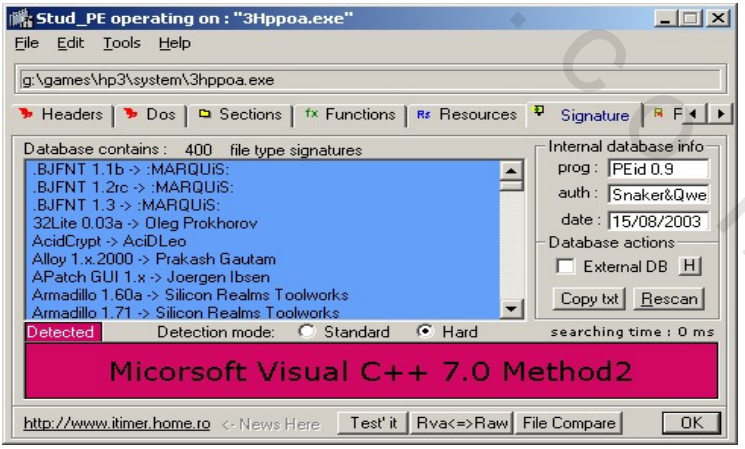
الصفحة التالية functions وهي الصفحة المعتادة التي رأيناها سابقا في العديد من البرامج وتحتوى هنا أيضا على الدوال المستخدمة في البرنامج



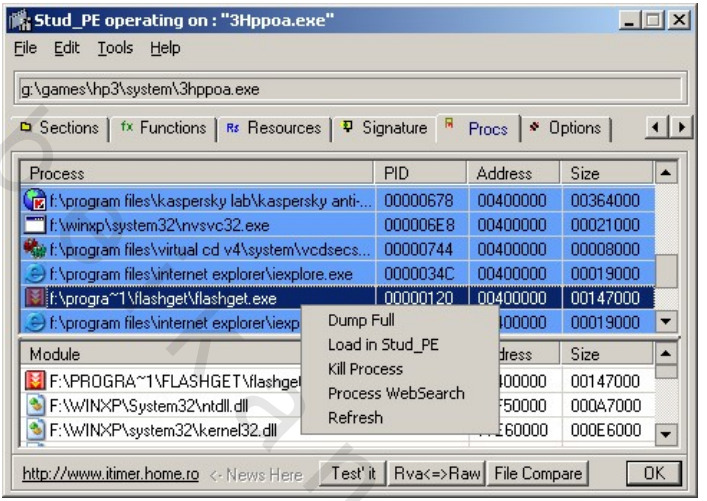
الصفحة resources تمكننا من رؤية المصادر التي يستخدمها البرنامج مثل الديالوجات والايقونات التي يحتويها الملف التنفيذي .



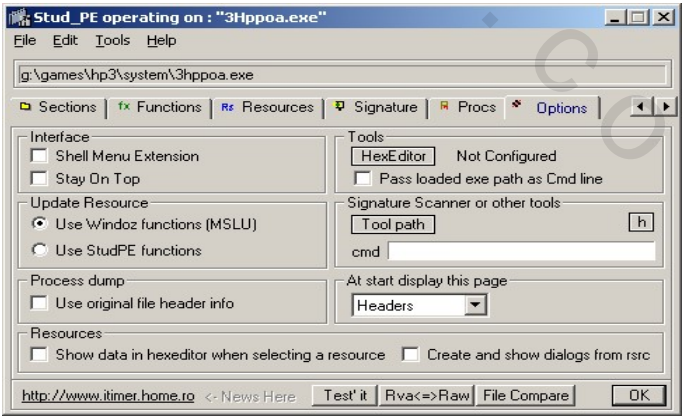
الصفحة التالية تبين اللغة المستخدمة في البرنامج أو نوع التشفير المستخدم إذا كان الملف مشفرا ويستخدم البرنامج الأداة Peid وملفات متعددة للتعرف على اللغة عن طريق وجود بيانات محدده كل منها تحدد نوع اللغة المستخدمة



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.

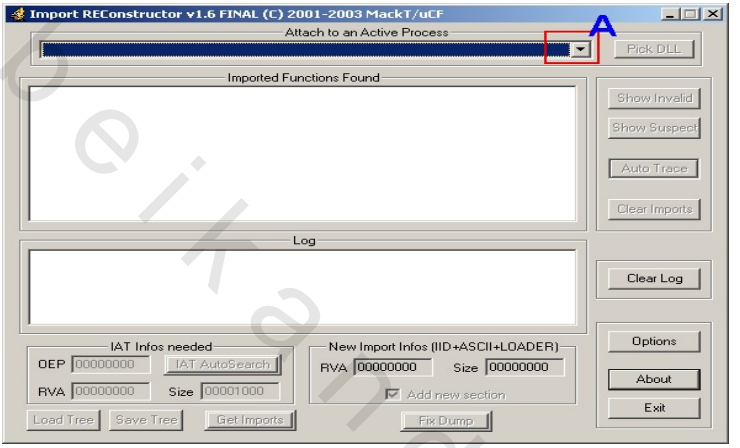


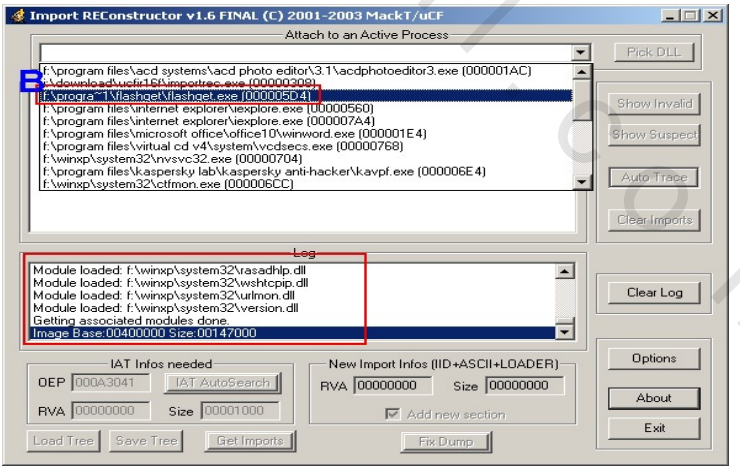
والصفحة الأخيرة وهي تحتوى على اختيارات البرنامج مثل إلحاق البرامج التنفيذية بهذا البرنامج أو اختيار محرر سداسي عشر خارجي غير المستخدم مع البرنامج



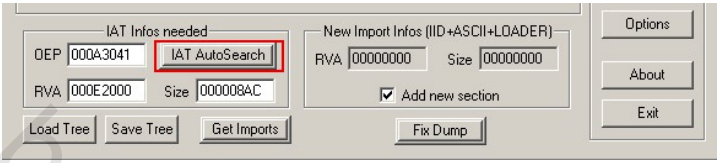
برنامج Import Reconstructure:

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 تمكننا من اختيار أي برنامج محمل حاليا بالذاكرة كما يتضح بالشكل التالي:



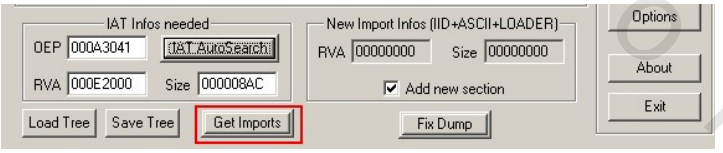


الخطوة التالية هي الضغط على المفتاح IAT Autosearch الذي يستطيع البحث عن العنوان المخصص للنداء على الدوال الخارجية كما بالشكل:

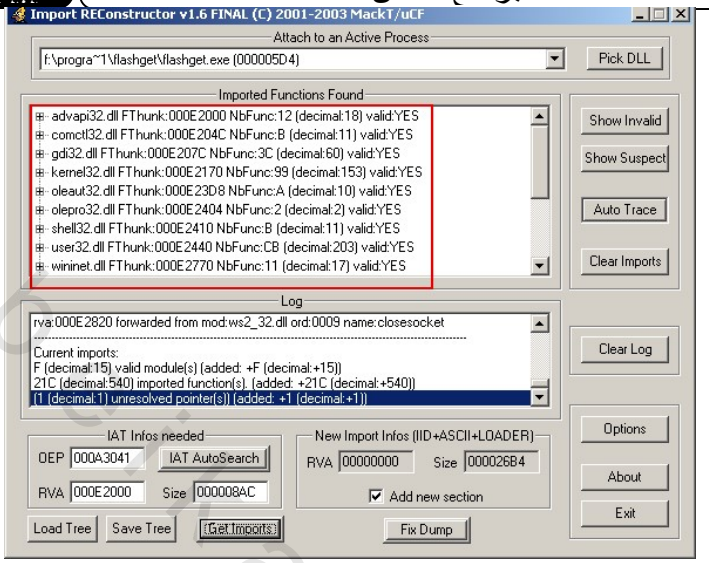


ستجد البرنامج اظهر رسالة تفيد احتمال وجود الدالة بعنوان معين

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

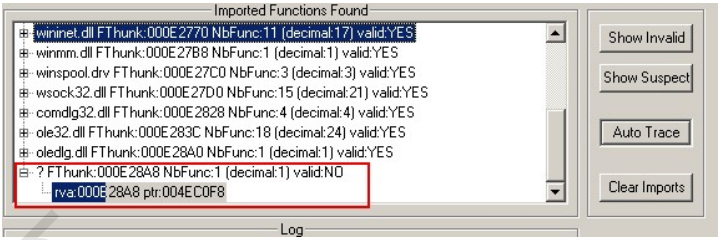


و تكون النتيجة هي إظهار الدوال المستخدمة في الشاشة الرئيسية كما يلي:

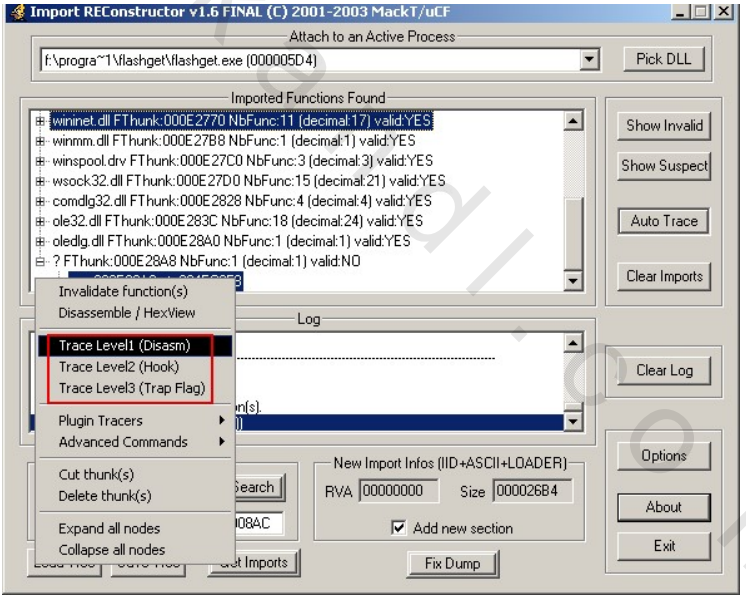


الخطوة التالية هي الضغط على المفتاح show invalide لإظهار الدوال الفاسدة فقط وتصليحها

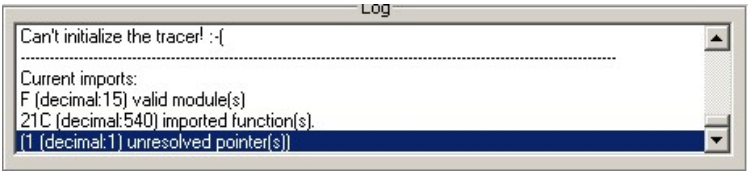
و بعد ذلك يقف البرنامج تلقائيا على الدالة الغير صالحه كما بالشكل التالي:



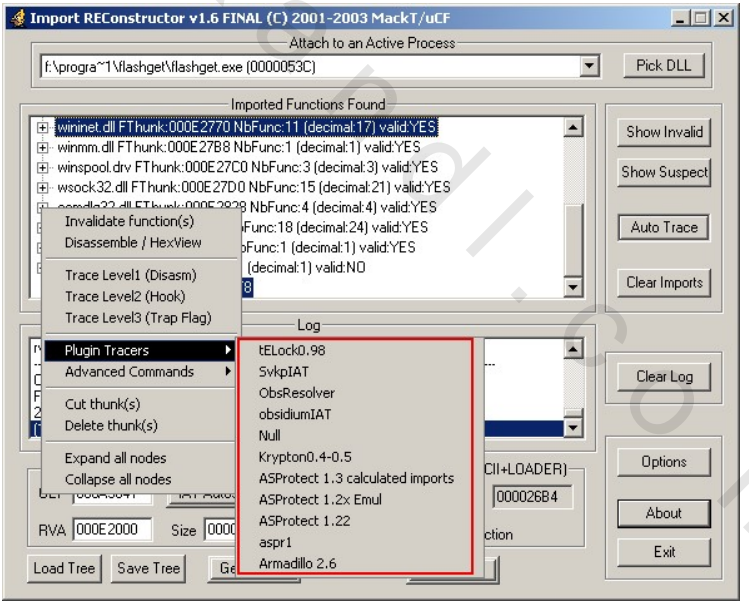
و بالضغط على المفتاح الأيمن للماوس يمكنك أن ترى قائمة الأوامر المختصرة وسنقوم هنا باستخدام احد أوامر التتبع trace لإزالة الدوال الغير صالحه



و تظهر النتيجة في الجزء Log كما يلي:

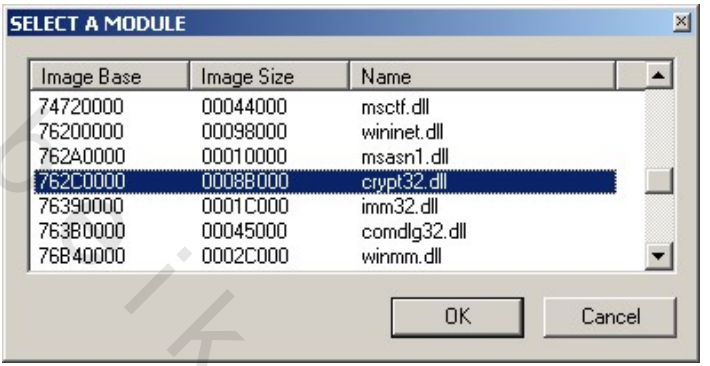


هذا ويحتوى البرنامج على عدة وظائف إضافية plugins يمكنك من خلالها تصليح الدوال التي لم تستجيب لأوامر trace وسيحدث ذلك إذا كان الملف مشفرا بأي من طرق التشفير المعروفة مثل armadillo اوaspack ولعلاج ذلك قم باختيار نوع التشفير من القائمة المختصرة كما بالشكل:

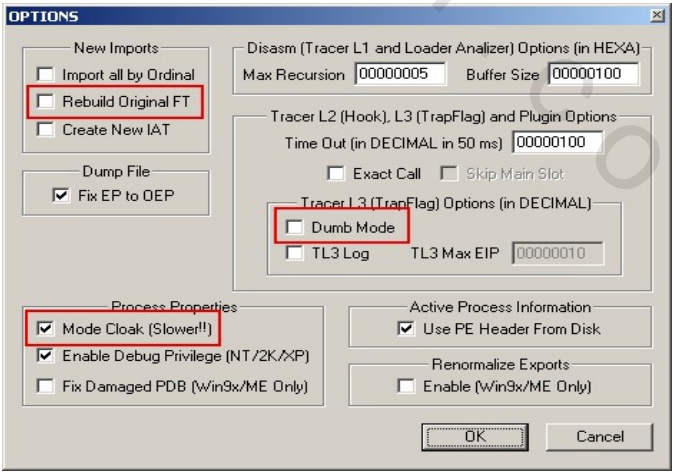


و يمكنك أيضا إصلاح باقي أجزاء الملف من المفتاح fix dump

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



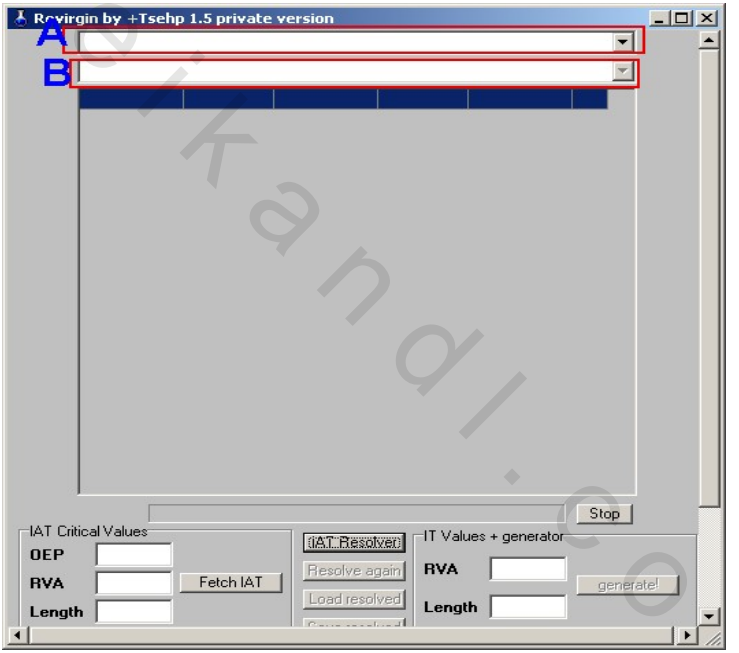
ويمكنك ضبط اختيارات البرنامج من المفتاح option ومن التضبيطات المفيدة الاختيار mode clock الذي يقوم بإخفاء البرنامج rec من الملف المحمل حتى لا يستطيع كشفه وإغلاق نفسه كما يوجد اختيارات أخرى لتغيير طريقة التتبع والفحص



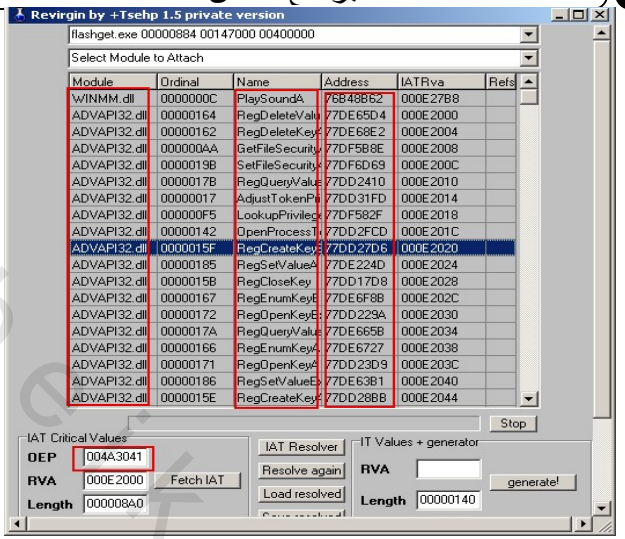
برنامج revirgin:

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

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

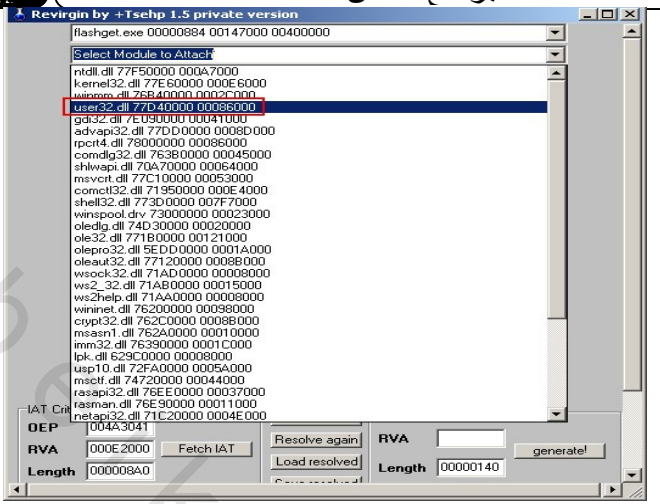


ومثل البرنامج Import REC قم باختيار الملف من الذاكرة من أول قائمه منسدلة فيقوم البرنامج بعرض كل ملفات المكاتبات (DLL Files) التي يقوم البرنامج بالنداء عليها والعنوان الخاص بكل داله واسم الدالة المستخدمة في الملف

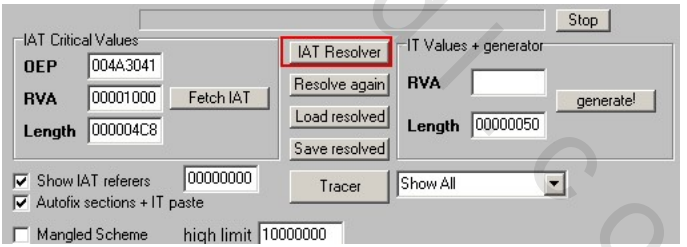


و كما يتضح من الشكل السابق نشاهد البيانات في جدول مقسم إلى أعمده ويوجد بها كل من اسم الملف واسم الدالة وعنوان النداء على هذه الدالة في الملف

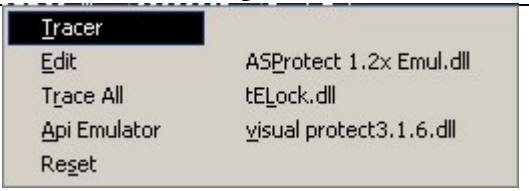
و يمكنك أيضا متابعه النداء على ملف معين باختياره من القائمة المنسدلة الثانية كما بالشكل:



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



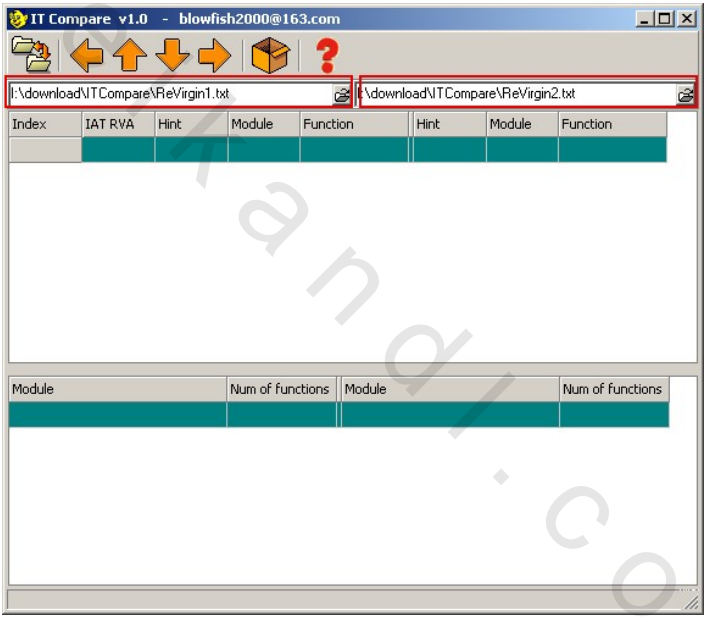
ثم بعد ذلك قم باختيار الدوال التي بها مشاكل بالماوس واختار من القائمة المختصرة الأمر Tracer:



برنامج IT Cmpare:

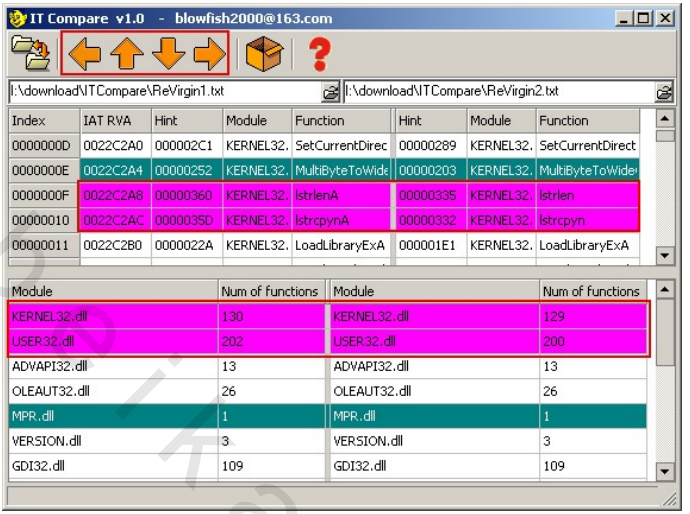
يتيح هذا البرنامج دقة أكثر وسهوله لرؤية الاختلاف بين ملفين والوصول إلى دوال API التي تحتاج إلى إصلاح فبعد استخدامك لبرنامج Import REC أو برنامج ReVirgin يمكنك عن طريق احد البرنامجين توليد ملفات مقارنه ويجب أن تكون هنا ملفات نصيه txt file

قم بتشغيل البرنامج وحدد الملفين المراد مقارنتهما كما بالشكل :



بعد ذلك قم بالضغط على مفتاح بداية المقارنة

سيبدأ البرنامج فورا مقارنة الملفين ويظهر النتائج كالتالي:

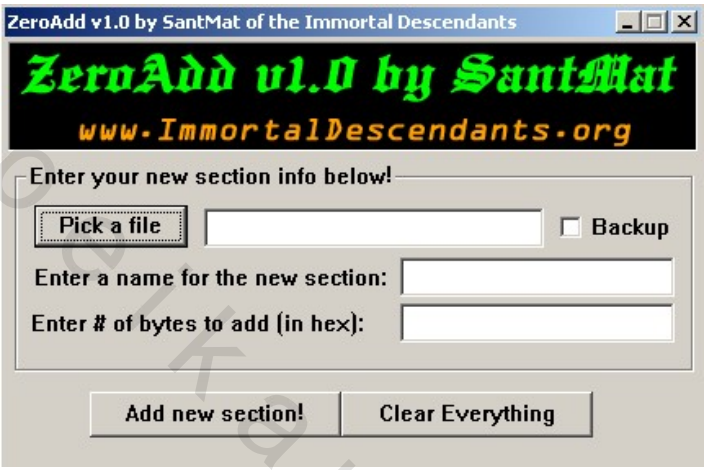


تحتوى النافذة العلوية على كل دوال API والنافذة السفلى على إحصائيات لعدد الدوال ويمكنك أن تشاهد الاختلاف باللون القرنفلي والدوال التي ليست بها اختلاف باللون الأبيض ويمكنك الذهاب بسرعة لمواقع الاختلاف عن طريق مفاتيح الأسهم

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

برنامج Zero Add:

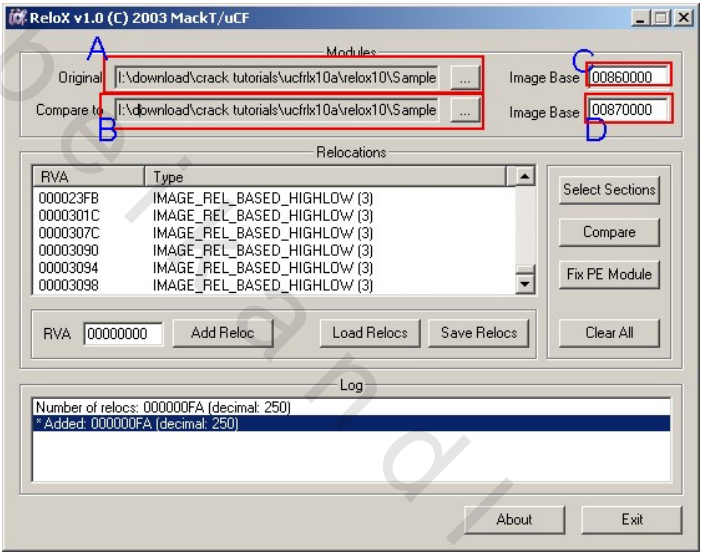
هذا البرنامج بسيط وهو يقوم ببساطه بإضافة مقطع جديد للملف



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 .

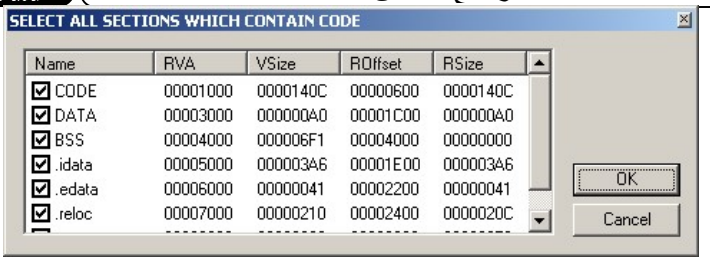
برنامج ReloX:

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

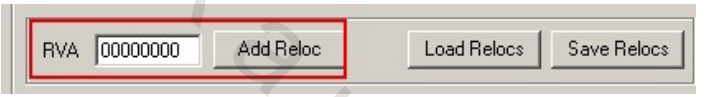


الخطوة الأولى هي تحديد ملف التفريغ للملف الأصلي ثم ملف التفريغ للملف الذي تريد مقارنته به وبعد ذلك تقوم بتحديد رقم أساس للملفين مختلف إذا لم يستطع البرنامج اكتشافهم في الخطوة C و D

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

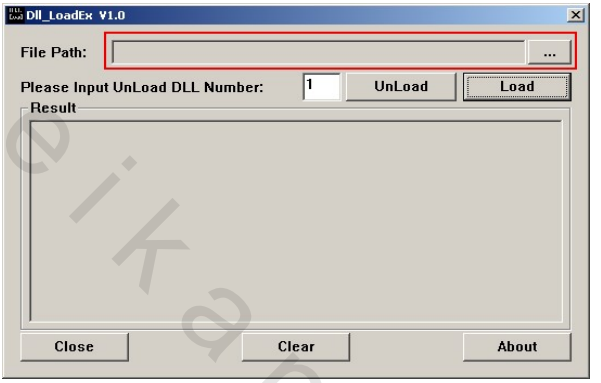


قم بعد ذلك بالضغط على المفتاح compare وإذا وجدت فروق يمكنك إصلاح ذلك بالمفتاح fix PE وسيقوم البرنامج بإنشاء مقطع .reloc جديد أو إضافة مقطع يدوي إلى الملف بالمفتاح Add Reloc

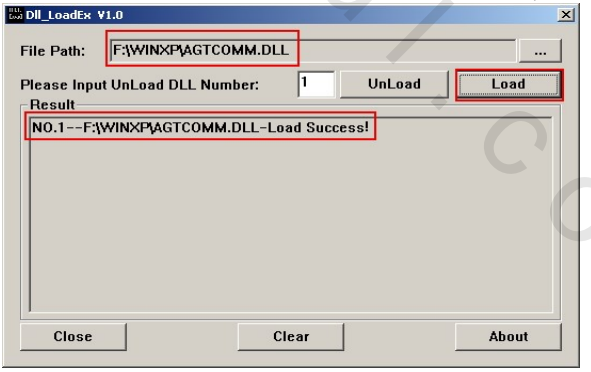


برنامج DLL_Loader:

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:



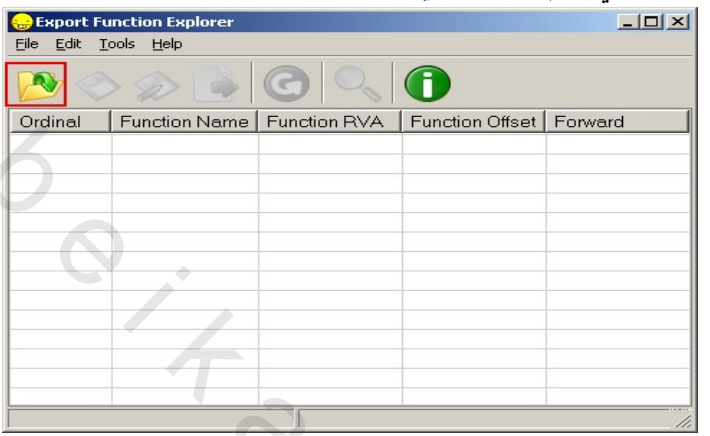
قم أولا باختيار ملف للتحميل ثم اضغط على المفتاح load الذي يقوم بتحميل ملف المكتبة في الذاكرة



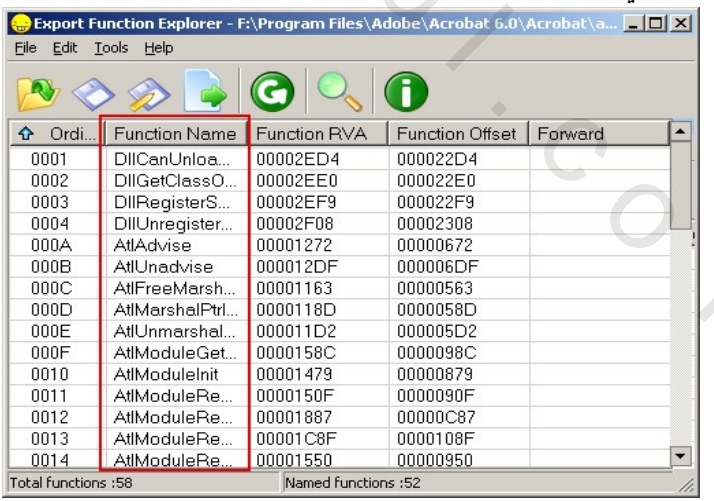
و يجب بعد الانتهاء أزاله الملف من الذاكرة حتى لا يشغل مساحه إضافية ويتم تنفيذ ذلك عن طريق المفتاح Unload

برنامج Export Function Explorer:

هذا البرنامج البسيط يستطيع معرفة كل الدوال الخاصة بملف exe أو ملف dll والتي يقوم الملف بتعريفها



قم أولا 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

أدوات Debuggers

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

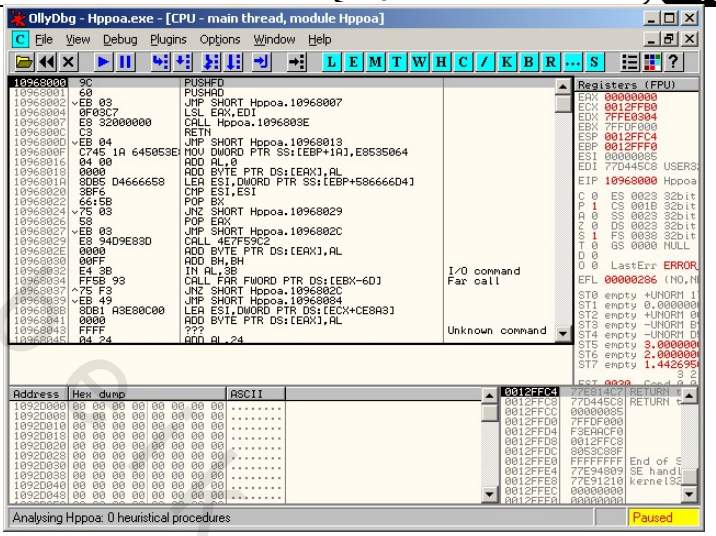
برنامج OllyDbg:

يعتمد البرنامج على البنية 32 bit لتحليل أي ملف إلى لغة الاسمبلى ويدعم كل المعالجات الحديثة تقريبا مثل

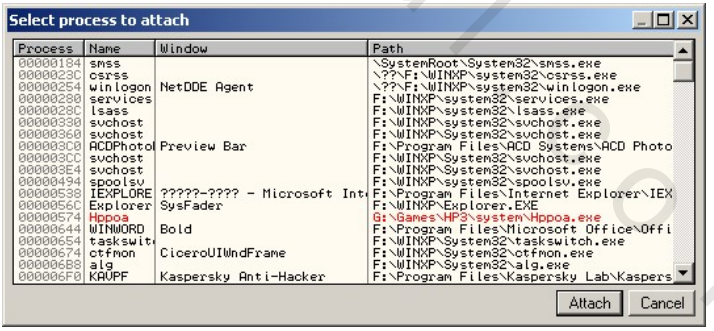
80x86, Pentium, MMX, 3Dnow!, Athlon extentions, SSE

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 .

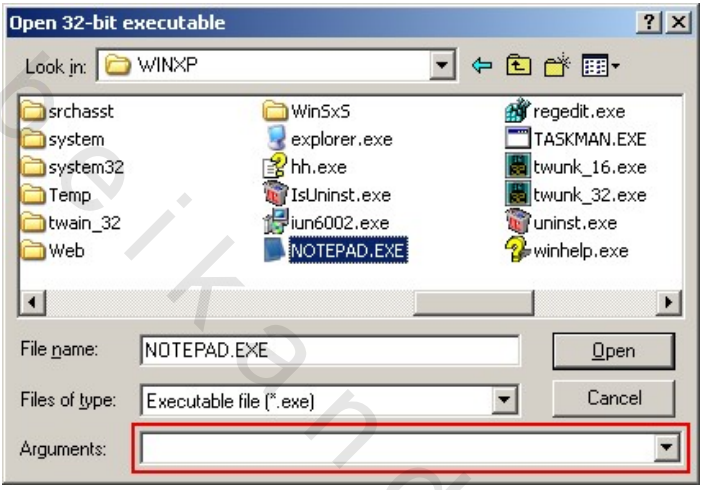


أول خطوه هي اختيار برنامج عن طريق الأمر File | Open أو اختياره من الذاكرة إذا تم تشغيل البرنامج من قبل:



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.

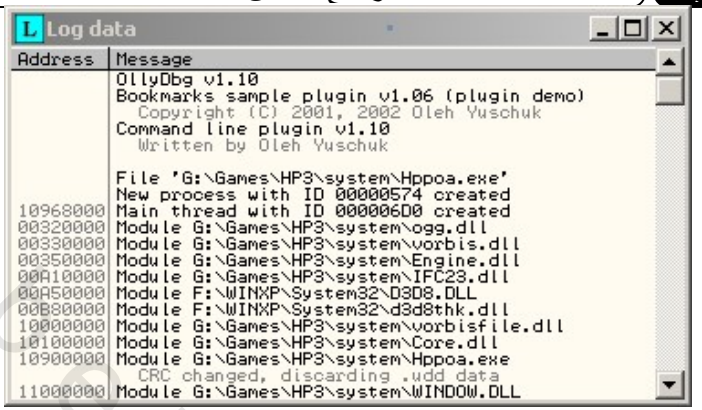
أما الأمر Open فهو يختلف في شيء واحد فقط عن أي أمر Open وهو احتوائه على سطر لأوامر المعاملات وبالتالي يمكن إعطاء الملف المراد اختباره قيمه معينه ومشاهدة كيف يتم تنفيذ البرنامج بناء على هذه القيمة



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

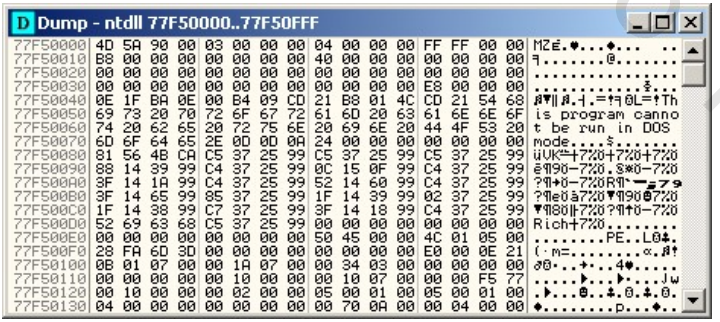
النافذة LOG:

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



و تحتوى قائمة الأوامر المختصرة على الأوامر التالية

Dump at address: لعرض محتويات الذاكرة للملف المختار في نافذة dump



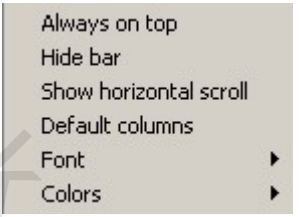
Dump in cpu: لعرض محتويات الملف في نافذة CPU

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

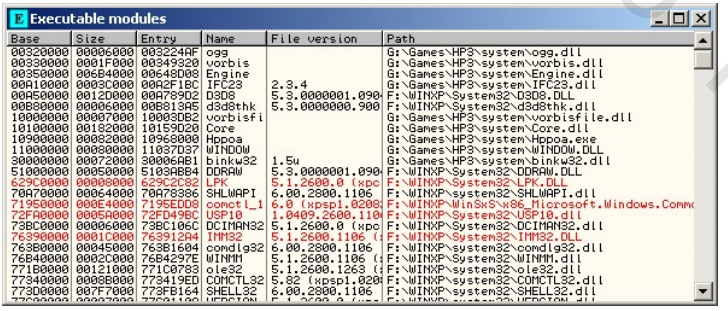
Appearance:

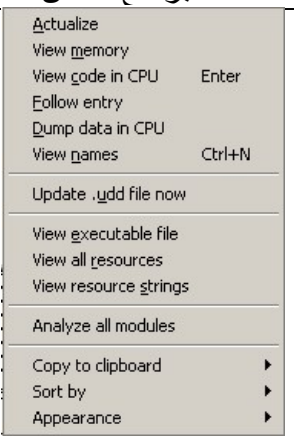


و بها الأوامر التالية : تمكين النافذة لتصبح دائما في المقدمة – إخفاء أو إظهار شريط الحالة – إظهار أو إخفاء شريط التمرير – الأعمدة الافتراضية للظهور – نوع الخط وحجمه – الألوان

نافذة Executable Modules :

تحتوى 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

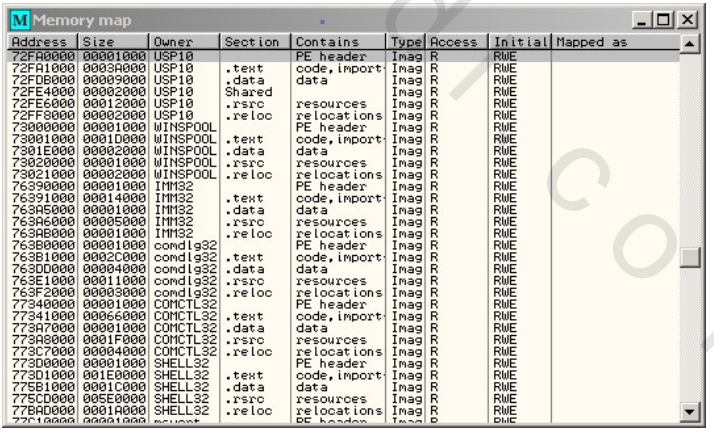




و تحتوى قائمة الأوامر المختصرة على الأوامر التالية:

Actualize: وضع علامة أن البرنامج قديم فلا يتم تمييزه باللون الأحمر

View memory: تمكننا من مشاهدة محتويات الملف الحالي بالذاكرة

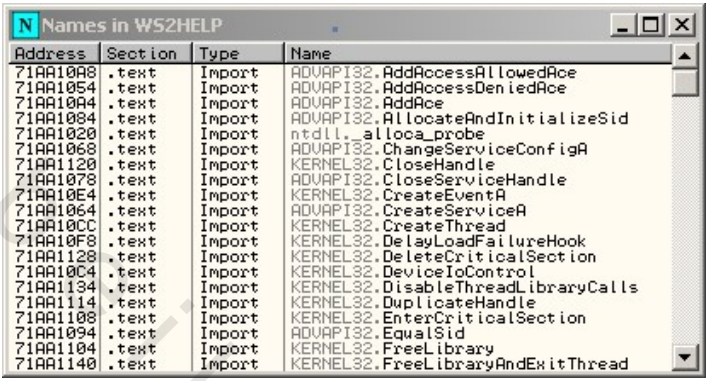


View code in cpu: يمكننا هذا الأمر من مشاهدة تعليمات الاسمبلى في نافذة CPU للملف الحالي

Follow entry: تمكننا من تتبع البرنامج المختار في نافذة cpu

Dump data in cpu: تفريغ كل البيانات للبرنامج المختار في نافذة cpu

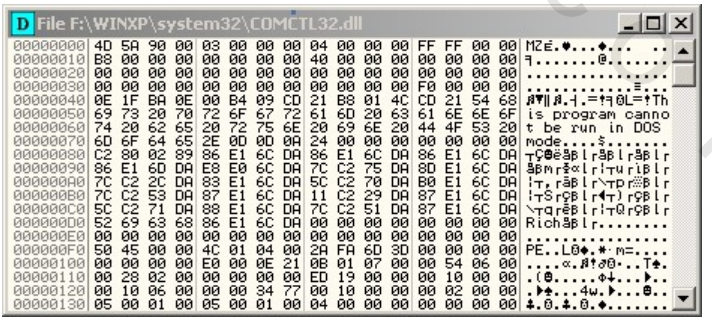
View Names:



من الأوامر المفيدة جدا حيث يمكننا هذا الأمر من مشاهدة أسماء الادله واسم المقطع الذي توجد به وعنوان الدالة ونوعها إذا كانت imported أو exported

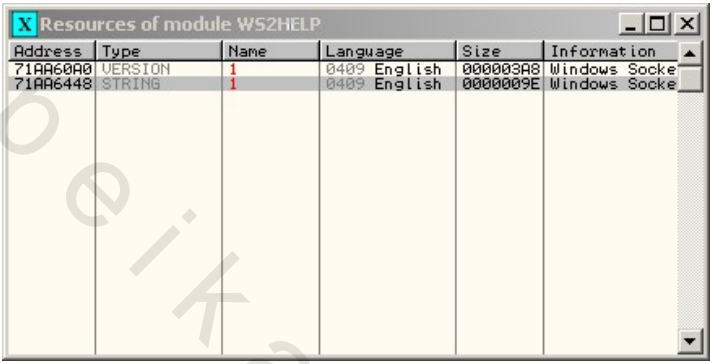
Update udd file now: برنامج ollydbg يقوم بحفظ كل بيانات البرامج أو النوافذ إلى ملفات بالامتداد udd ويستطيع استعادة هذه المعلومات مره أخرى عند تحميل البرنامج وهذه المعلومات تتضمن التعليقات والتعريفات ونقاط التوقف ونقاط المراقبة والنقاط الشرطية وبيانات تحليليه وغيرها.

View Executable File:



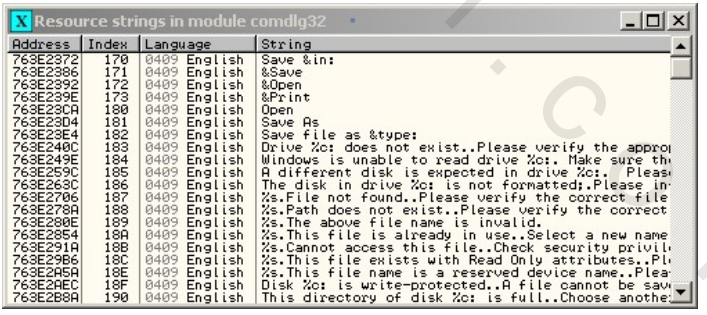
يمكننا هذا الأمر من مشاهدة محتويات الملف بالنظام السداسي عشر في نافذة dump

View All Resources:



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

View resource strings:



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

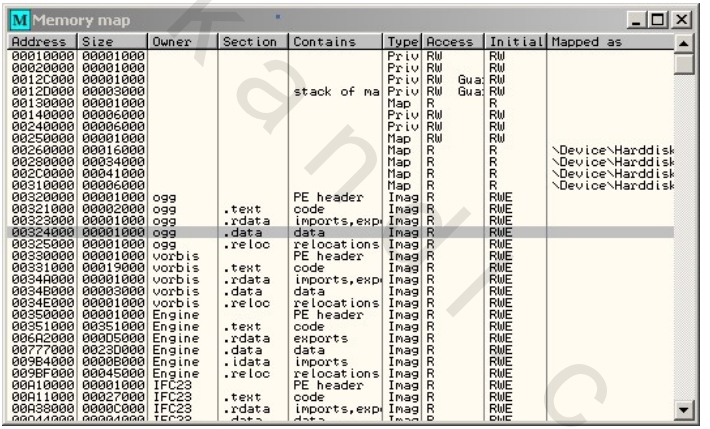
Copy to clipboard: لنسخ محتويات النافذة إلى الذاكرة ويمكنك نسخ سطر واحد أو عدة اسطر أو حتى أعمده محددة.

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

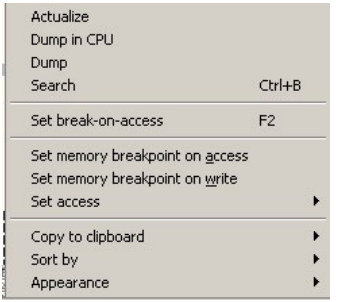
Appearance: لتغيير شكل النافذة كما سبق شرحه.

النافذة Memory Map:

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



و تحتوى قائمة الأوامر المختصرة للنافذة على التالي:

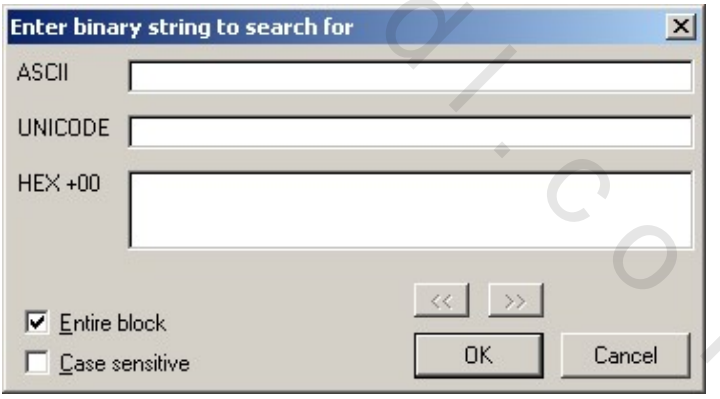


الأمر Actualize: كما شرحنا سابقا يضع علامة لكل المدخلات على أنها قيمه حتى لا يتم تمييزها باللون الأحمر

dump in cpu : يقوم بتفريغ محتويات النافذة في نافذة cpu dump:

يعرض هذا الأمر محتويات الذاكرة بنافذة التفريغ بشكل نصي

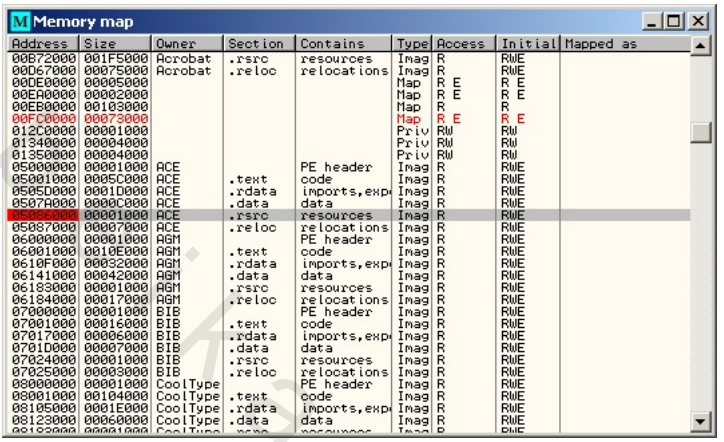
Search:



يمكننا هذا الأمر من البحث في محتويات الذاكرة إما بنص أو بنص قياسي بنظام Unicode أو بالنظام السداسي عشر.

Set Break point on access:

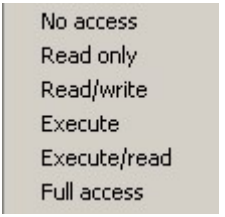
يقوم هذا الأمر بوضع نقطة إيقاف عند مكان معين بالذاكرة حتى يتوقف البرنامج عندما يصل إلى هذه النقطة.



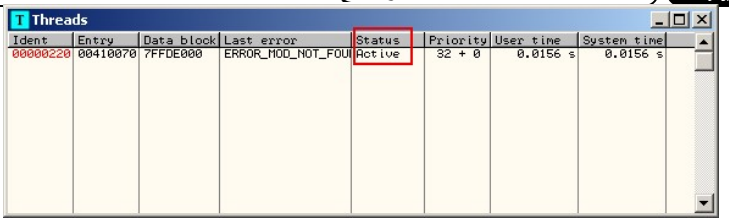
لاحظ مكان نقطة الإيقاف باللون الأحمر

أوامر Memory Breakpoints تمكننا من وضع نقطة إيقاف عند قراءة أو كتابة على جزء محدد من الذاكرة

set access: يمكننا من تحديد نوعيه التعامل مع هذا الجزء من الذاكرة فيمكننا من وضع القيم التالية



النافذة Threads :



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 محدده

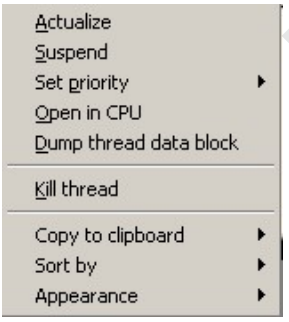
Suspended : البرنامج متوقف

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:

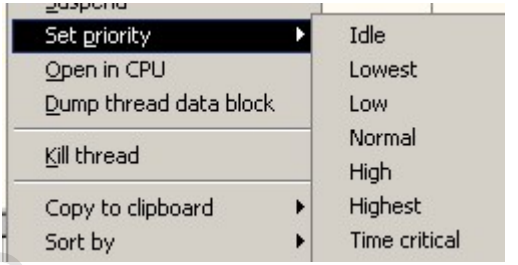


Actualize: وضع علامة على كل البرامج أنها تم تنفيذها من قبل

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

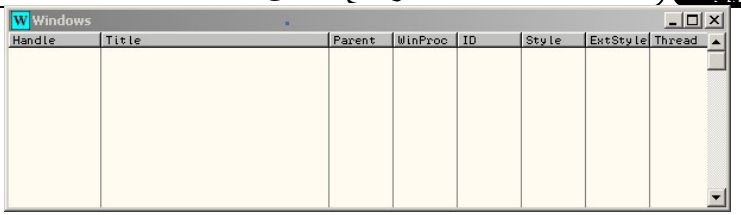
Copy to clipboard : يمكننا هذا الأمر من نسخ محتويات الذاكرة إلى ذاكرة clipboard

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.

النافذه Windows:



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 مجموعه 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

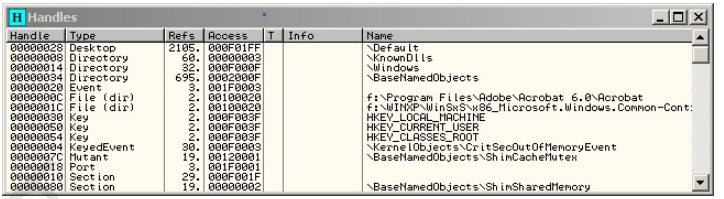
[ESP+00] Return address
[ESP+04] Window's handle
[ESP+08] Message
[ESP+0C] wParam
[ESP+10] lParam

و هذا يعنى انه يتم التوقف عند تعريف رقم 00001234 للنافذه والرسائل التي يتم التعرف عليها هي

BM_GETCHECK…BM_SETIMAGE,
WM_CTLCOLORBTN

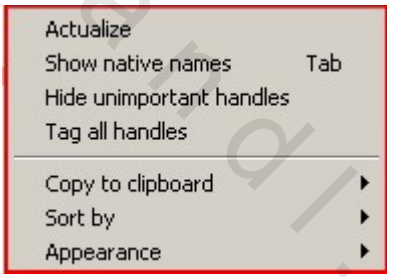
لاحظ أن رقم تعريف النافذة (window handle) يتغير في كل مره نقوم بها بتنفيذ البرنامج لذلك لن يكون شرط نقطة التوقف صالح بعد إغلاق النافذة.
إذا أردت أن تعرف المزيد عن الرسائل راجع الملحق أ

النافذة Handles:



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

و تحتوى 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

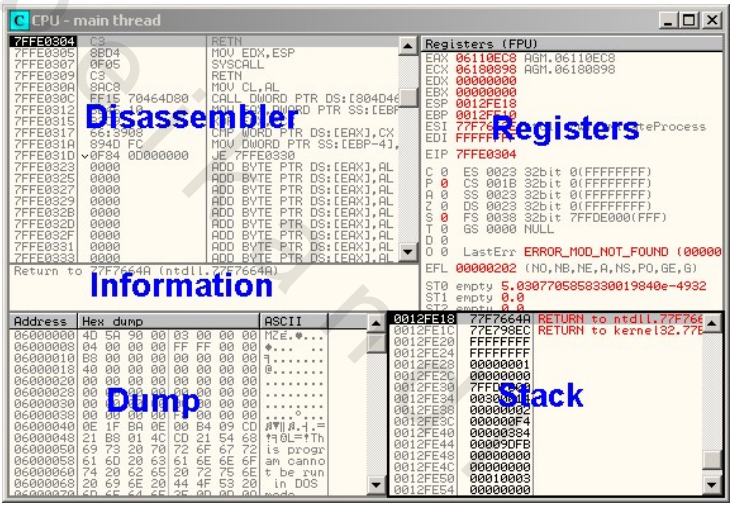
Copy to clipboard: لنسخ محتويات النافذة للذاكرة

Sort by: لتحديد ترتيب البيانات حسب عمود محدد

Appearance: لتغيير شكل وهيئة النافذة

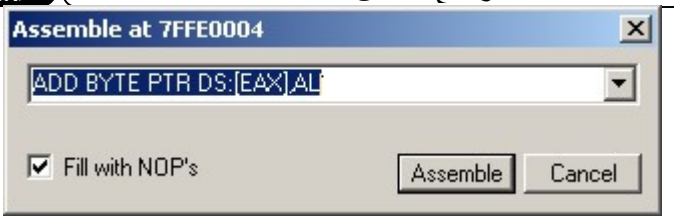
النافذة CPU:

وهي النافذة الرئيسية للبرنامج وتنقسم إلى عدة نوافذ داخليه هي:



Disassembler:

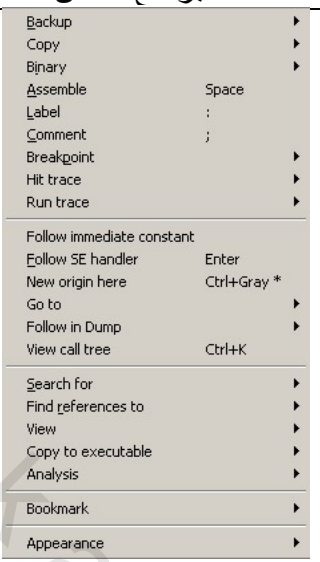
وتحتوى هذه النافذة على كود البرنامج الذي يتم اختباره وتتضمن أربع أعمده هي العنوان والتفريغ السداسي عشر للأمر الحالي وأوامر الاسمبلى وأخيرا التعليقات الضغط المزدوج على أي أمر اسمبلى في العمود assembly يظهر ديالوج يمكننا من تعديل الأمر:



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

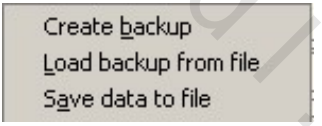
عندما يتم إيقاف البرنامج المراد اختباره فيتم التوقف عند الأمر الذي يعتبر EIP أو نقطة الدخول للبرنامج

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



:Backup

و يحتوى على الثلاث أوامر التالية:



Create backup يقوم بإنشاء ملف احتياطي لبيانات الاسمبلى

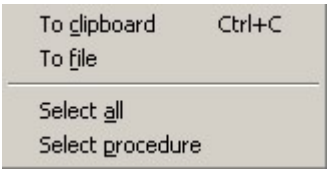
Load backup from file يستخدم عند إجراء تعديل لأوامر الاسمبلى وتريد الرجوع عنه

Save data to file يقوم البرنامج بتخزين بيانات الذاكره في ملف بالامتداد *.mem

وتظهر أوامر أخرى بعد إجراء backup مثل update لتحديث النسخة الاحتياطية و delete لحذف النسخة الاحتياطية.

:Copy

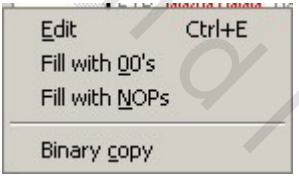
يحتوى الأوامر التالية:



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

:Binary

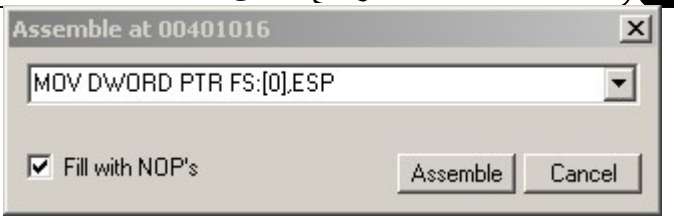
يحتوى الأوامر التالية:



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:

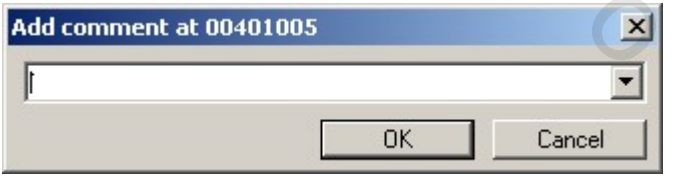


:Label

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

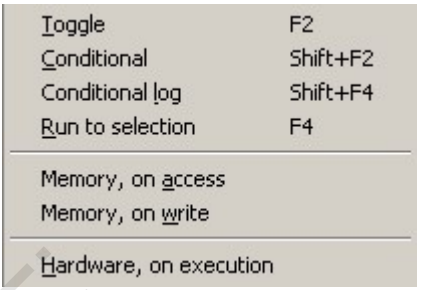
:Comment

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 الديالوج الذي سيظهر



:Breakpoint

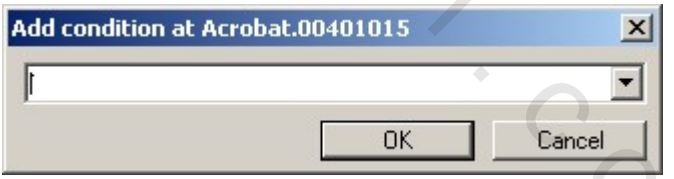
تظهر قائمه فرعيه للتعامل مع نقاط التوقف وتحتوى الأوامر التالية:



: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 x0111

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

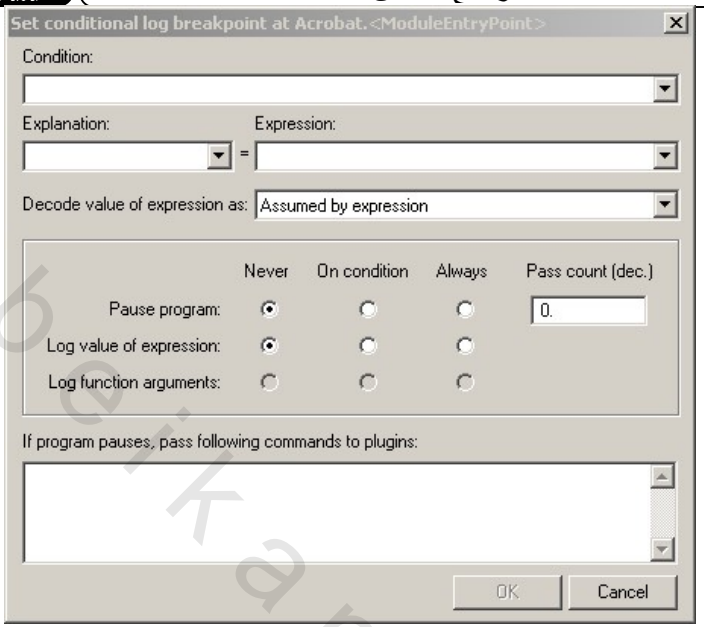
أما

(2+3)*4 = 20

يمكنك تغيير الترتيب إذا استخدمت الأقواس.

Conditional log:

يمكن عن طريق هذا الأمر وضع نقطة إيقاف شرطيه 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.

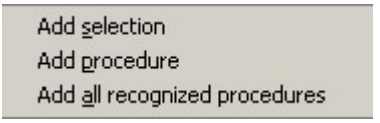
الاختيار pass count يمكنك من التوقف عند عدد محدد من المرات مثلا إذا كان هناك حلقه تكراريه تتكرر 100 مره فيمكنك وضع قيمة الاختيار 99 فيتم التوقف فقط عند آخر حلقه .

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

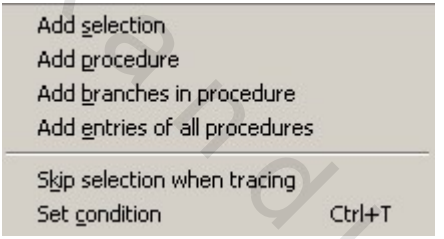
كما يمكن التوقف عند عمل اتصال بأي جزء من مكونات الكمبيوتر بالأمر hardware on execution

: Hit trace



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

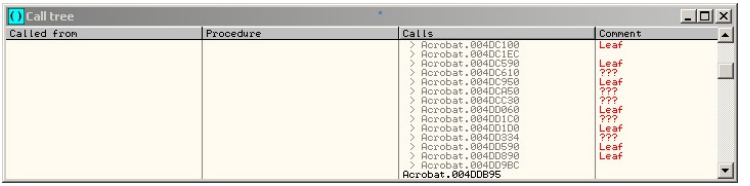
:Run trace



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



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

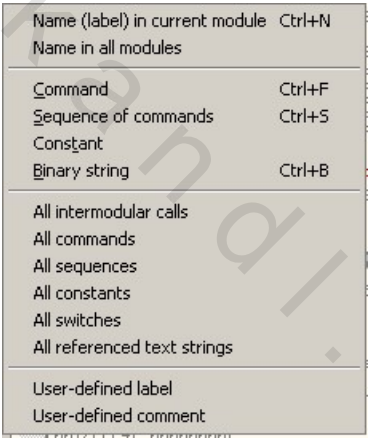
Pure لا يتم النداء على أية داله أخرى مع عدم وجود آثار جانبيه

(الإجراء لا ينادى نفسه بنفسه)

RETN يحتوى على أمر الرجوع

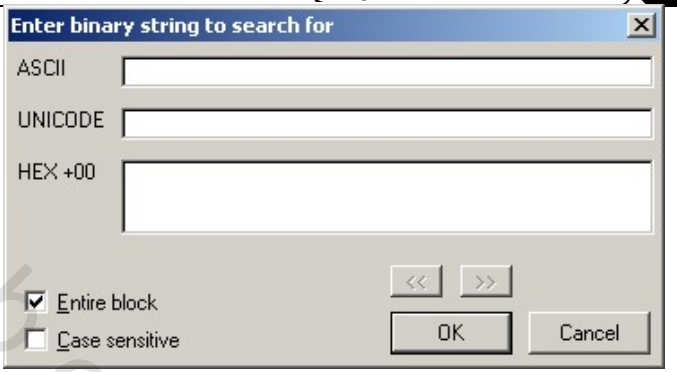
Sys الدالة هي من دوال النظام

:Search for



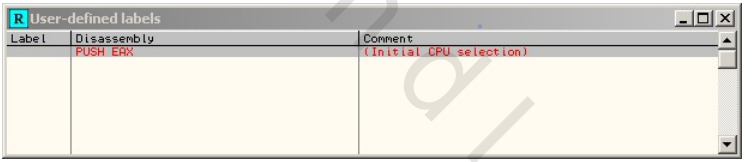
المجموعة الأولى للبحث عن معرف في الإجراء الحالي أول كل البرنامج.

المجموعة الثانية للبحث عن أمر أو مجموعه من الأوامر أو ثابت أو نص أو كود بالنظام السداسي عشر.



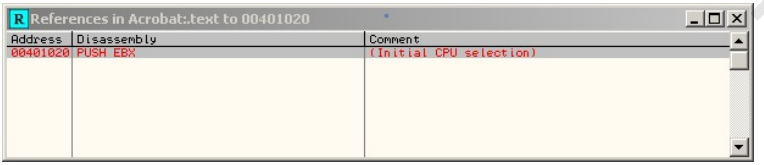
المجموعة الثالثة للبحث في فئات محدد مثلا كل الثوابت all constant أو كل النصوص المرجعية all refrenced text

المجموعة الأخيرة للبحث في التعليقات أو المعرفات التي قمت بإدخالها.



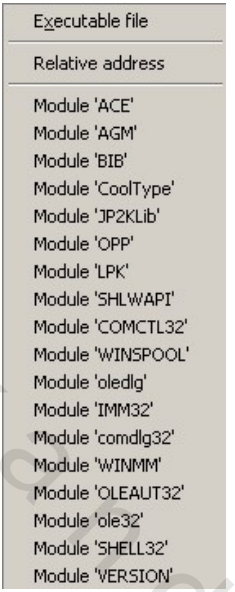
:Find reference to

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



:View

يمكنك عن طريقها اختيار احد الإجراءات المعرفة بالبرنامج والذهاب إليه:

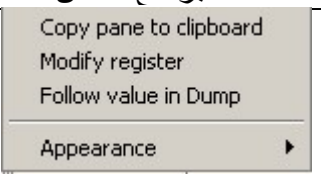


Analyze code: يقوم البرنامج بإجراء تحليل إضافي للكود ومن فوائد هذا الأمر انه يستطيع التفريق بين أجزاء البيانات وأجزاء الكود في الملف والتعرف على أماكن القفز والنصوص والحلقات التكرارية.

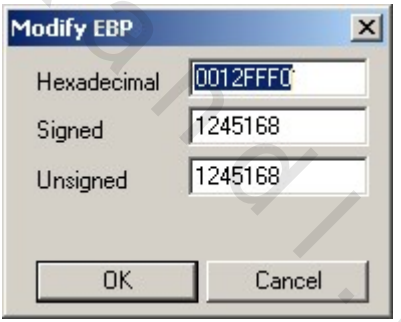
Bookmark: لتعليم جزء من الكود والرجوع إليه بسرعة عن طريق أمر goto bookmark

Appearance: للتحكم بشكل النافذة

نافذة Information:

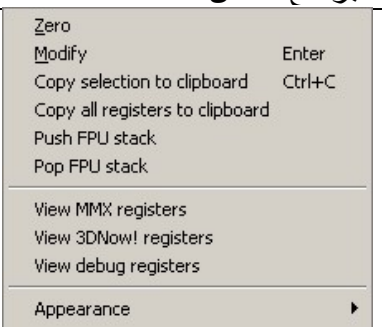


أثناء التوقف على سطر محدد في نافذة disassembly يمكنك رؤية قيمة ومعاملات المسجل الحالي في هذه النافذة وإذا قمت بعمل analyze code يمكنك رؤية الأوامر التي أدت للسطر الحالي وإذا كانت معلومات الديبج متضمنة في الملف يمكنك رؤية سطر الكود المصدر المكافئ للسطر الحالي.
ومن هذه النافذة يمكنك تتبع مسجل معين في نافذة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



نافذة Registers:

هذه النافذة تحتوى على المسجلات الخاص بالبرنامج ويمكن عن طريق هذه النافذة متابعة التغير في قيم المسجلات وتتغير 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 و3Dnow الخاص بمعالجات AMD عن طريق الضغط مره واحد على شريط الحالة أو باختيار نوع المعالج من القائمة المختصرة.

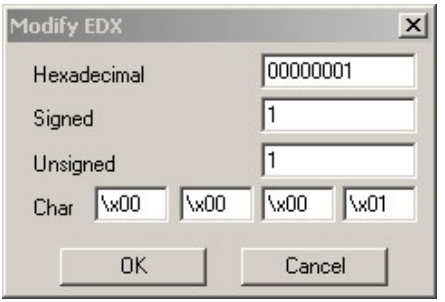
Zero: يمكنك من تصفير قيمة مسجل

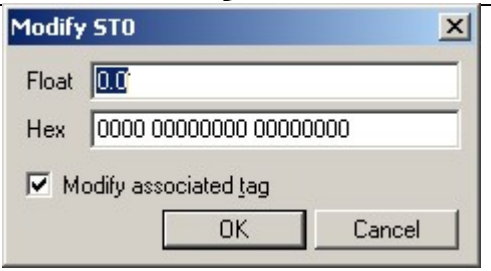
Increment: لزيادة قيمة المسجل بواحد والعكس decrement

Set to 1: لوضع القيمة 1 للمسجل

نافذة Modify:

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





نافذة Dump:

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

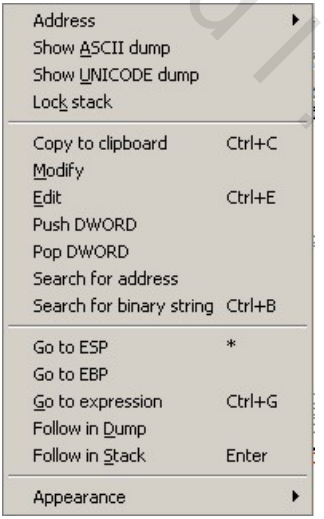
كما يمكنك رؤية كود نقطة الدخول في رأس الملف عن طريق الأمر special | PE header

Address	Hex dump	Data	Comment
00ACA000	00	DB 00	
00ACA001	00	DB 00	
00ACA002	00	DB 00	
00ACA003	00	DB 00	
00ACA004	40	DB 40	
00ACA005	C9	DB C9	
00ACA006	98	DB 98	
00ACA007	00	DB 00	
00ACA008	6F	DB 6F	
00ACA009	C9	DB C9	
00ACA00A	98	DB 98	
00ACA00B	00	DB 00	
00ACA00C	A1	DB A1	
00ACA00D	C9	DB C9	

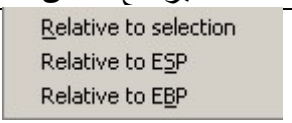
نافذة Stack:

هذه النافذة تظهر محتويات الذاكرة stack للتشغيل الحالي للبرنامج ويتم دائما الوقوف عند بداية الديبج 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



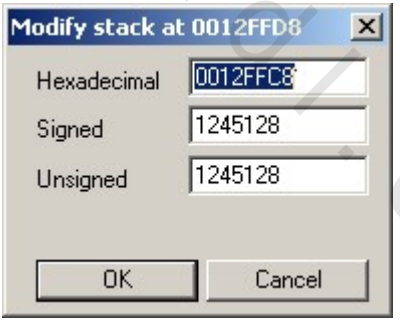
يمكنك رؤية العنوان الذي له علاقة بالاختيار الحالي أو بالمسجل ESP أو المسجل EBP في حالة المسجلات تتغير العناوين بتغير المسجلات

Show ASCII dump: يقوم بإظهار عمود إضافي يحتوى على تفريغ نصي

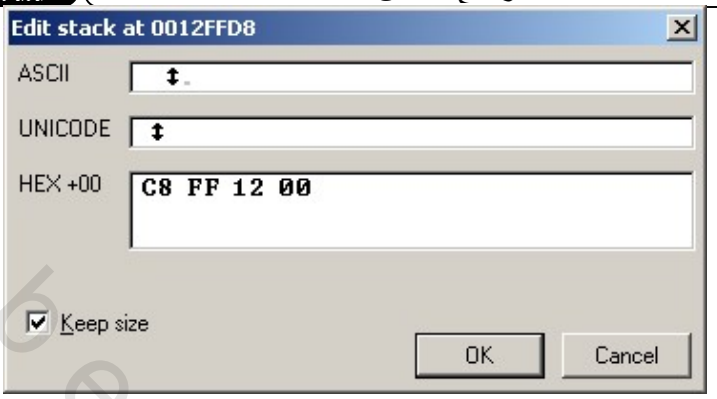
Unicode Dump: يظهر عمود إضافي يظهر نص قياسي من نوع Unicode

Lock stack: يقوم بإيقاف الانتقال التلقائي إلى المسجل ESP في كل مره يتم فيها إيقاف البرنامج.

Modify: تمكنك من تعديل أول رقم من نوع doublword



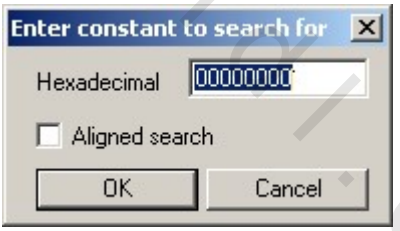
Edit: يمكنك من تعديل المحتويات بالنظام السداسي عشر أو نصوص ASCII أو Unicode



Push DWORD: إنقاص المسجل ESP بمقدار 4

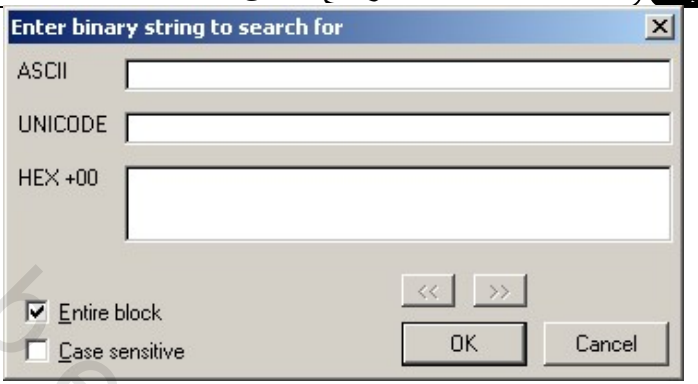
Pop DWORD: زيادة المسجل ESP بمقدار 4

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

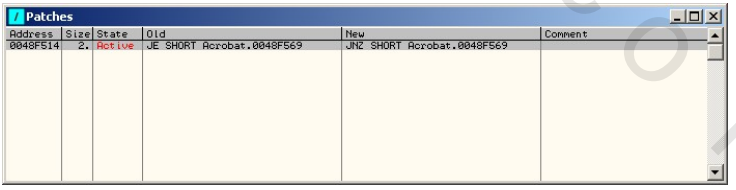
Goto EBP: للذهاب إلى محتويات المسجل EBP

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

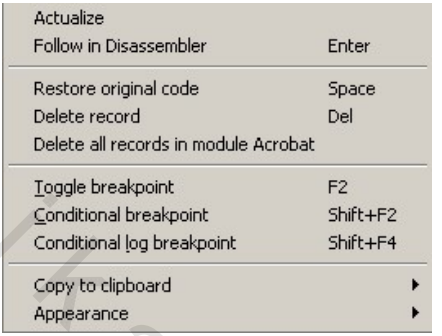
و بذلك نكون انتهينا من نافذة CPU وهي أهم نافذة ببرنامج OllyDBG

نافذة Patches:



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 Dissasembler بتغيير يمكنك أن تراه في نافذة Patches عن طريق الأوامر View Patches وعن طريق قائمة الأوامر المختصرة يمكنك التعامل مع هذا التعديل:



Follow in disassembler : لتتبع التعديل في نافذة disassembler

Restore original code: لإلغاء التعديل واستعادة الكود الأصلي

Delete record: لحذف كل القيم بنافذة patches

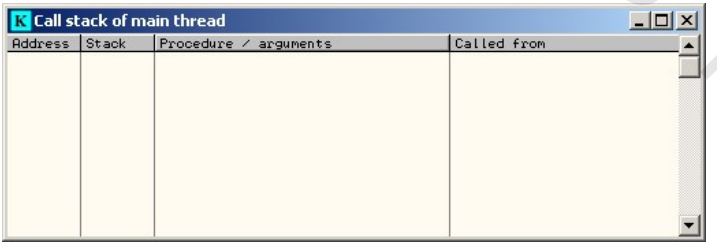
Toggole breakpoint: لوضع أو حذف نقطة إيقاف

Conditional breakpoint: لوضع نقطة إيقاف شرطيه

Copy to clipboard: للنسخ إلى الذاكرة

Appearance: للتحكم بشكل النافذة

النافذة Call Stack:



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:

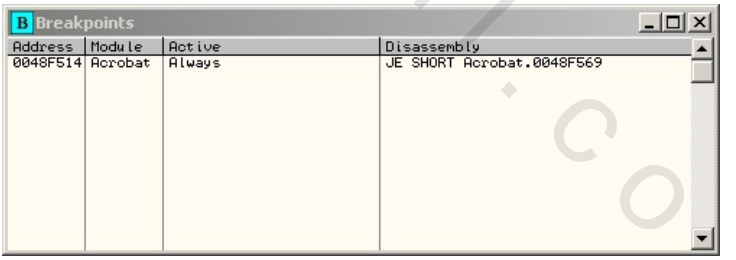
نقطة الدخول لا يعتمد عليها ?

العنوان تم تخمينه لعدم التوصل لنقطة دخول Maybe

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

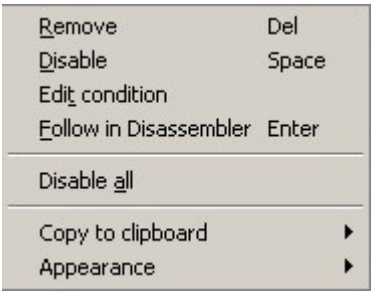
Called from: هو العنوان الخاص بالأمر الذي قام بالنداء على هذا الدالة

النافذة Breakpints:



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

العمود address يحتوى 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

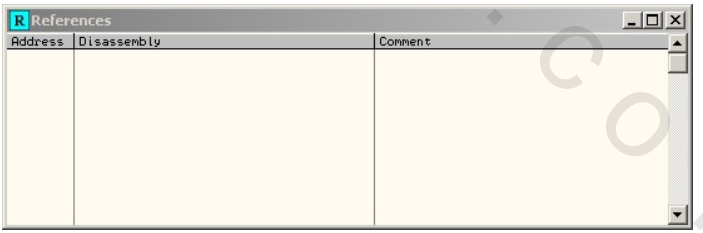
Disable: يقوم برنامج ollydbg بتجاهل نقطة الإيقاف

Edit condition: للتعديل في شروط نقطة التوقف

Follow in disasembleer: لتتبع نقطة الإيقاف في نافذة disassembler

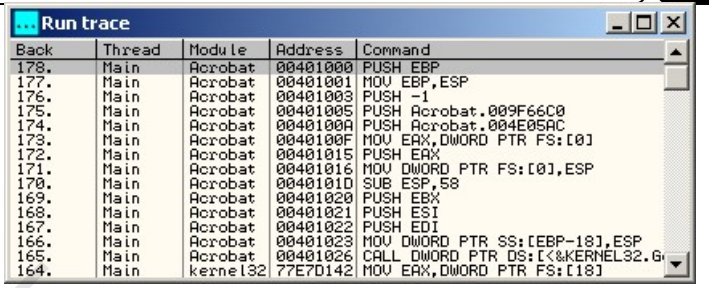
Disable all: لإبطال جميع نقاط التوقف

النافذة References:



تتيح لك هذه النافذة التنقل بسرعة بين العناوين والنصوص في نافذة disassembler

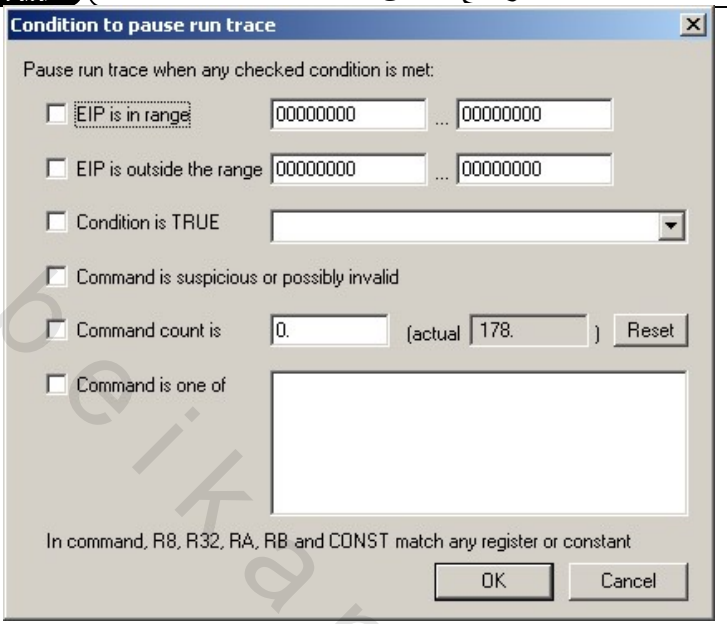
النافذة Run Trace:



يمكن عن طريق هذه النافذة تتبع عمليات البرنامج قبل حدوث بعض الإحداث 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 المفتاحين ctrl+f11 وهذا النوع من التتبع يقوم بالدخول في كل إجراء فرعى يجده أو الضغط على المفتاحين ctrl+f12 وهذا النوع لا يقوم بالدخول في الإجراءات الفرعية.

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

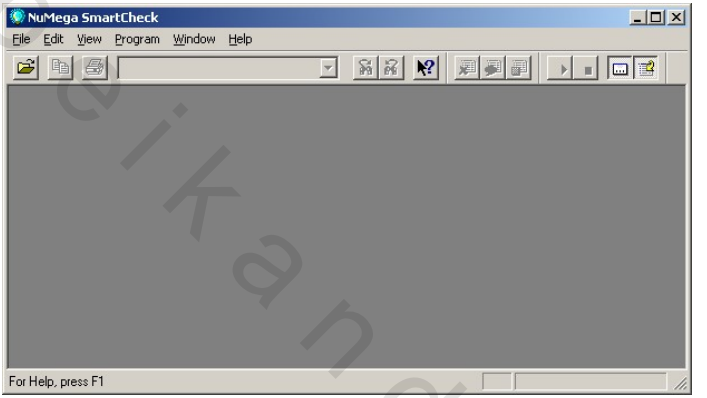


و يمكنك عن طريق النافذة السابقة تحديد التوقف مثلا عندما يصل المسجل EIP إلى مدى محدد من العناوين أو خارج هذه العناوين أو عندما يتحقق شرط معين وهكذا.

برنامج SmartCheck :

هذا البرنامج من أفضل برامج الديبجر والتي تلي ollydbg في القوه ويمكن عن طريقها اختبار البرامج المكتوبة بلغة فيجوال بيزيك .

تحتوى النافذة الافتتاحية للبرنامج على قوائم ومفاتيح اختصار يمكن عن طريقها تتبع البرنامج بسرعة وبسهوله.



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

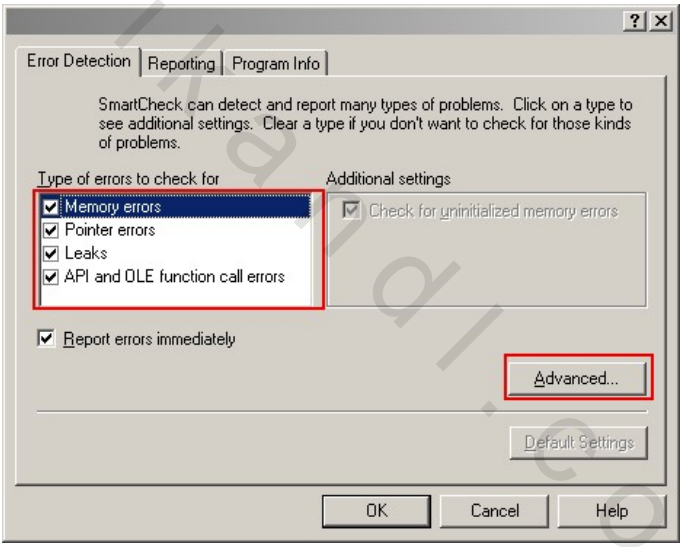
افتراضيا يقوم البرنامج بعمل التالي:

- ✦ 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

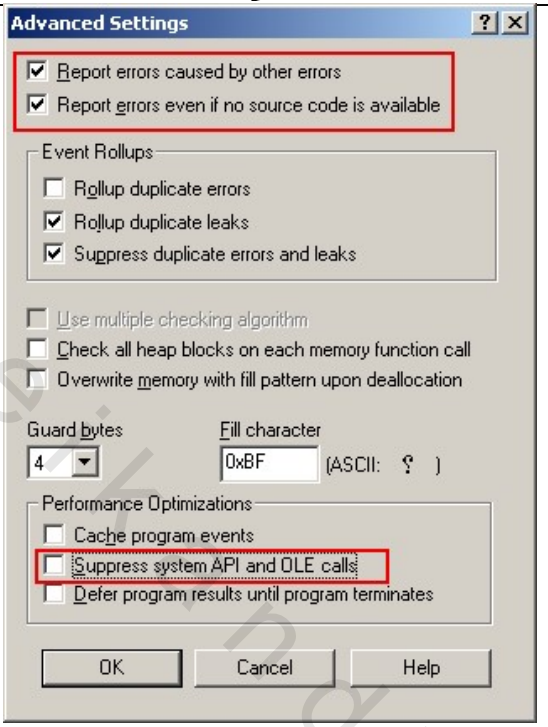
- يقوم بفتح نافذة result عند الانتهاء والتي تحتوى على جميع الأحداث والأخطاء التي وقعت

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

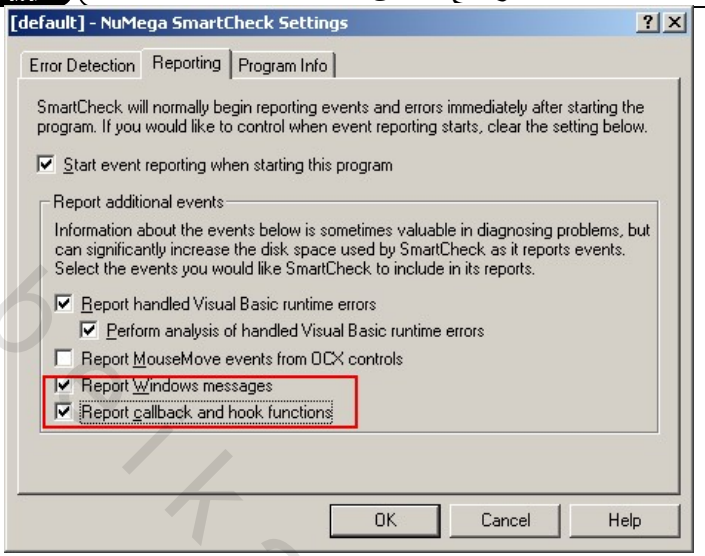
1. قم بفتح نافذة settings بالضغط على الأوامر | Program settings:



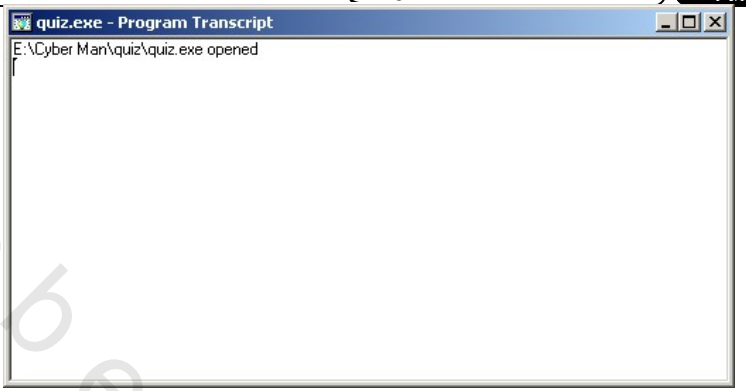
كما ترى الاختيارات الافتراضية للبرنامج هي اكتشاف جميع أنواع الأخطاء قم بالضغط على المفتاح advanced:



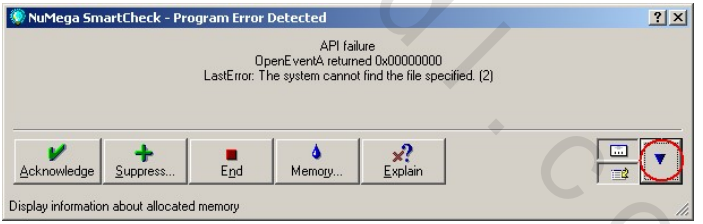
2. قم بالتأكد من اختيار report errors cused by other errors والاختيار report error even if no source code حتى يقوم البرنامج باكتشاف الأخطاء للبرامج التي نريد كسر حمايتها فلن يكون متاح لنا بالطبع الكود الخاص بتلك البرامج
3. تأكد أيضا من إزالة الاختيار suppress system API calls حتى لا يتجاهل البرنامج النداء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



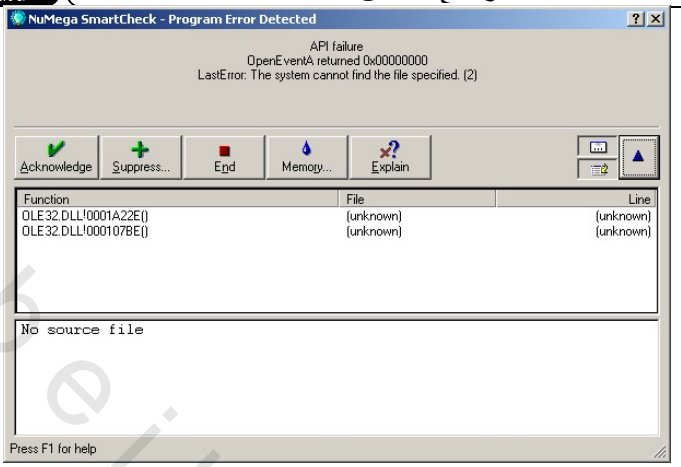
5. تأكد من اختيار الاختيارين report windows messages حتى يقوم البرنامج بتتبع رسائل الويندوز الناتجة عن النداء لبعض الدوال وأيضا الاختيار report callback and hook functions حتى يمكننا تتبع جميع أنواع الدوال
6. لكي تقوم بتتبع برنامج اختار الأوامر File | Open أو المفتاح المختصر ثم قم باختيار الملف التنفيذي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



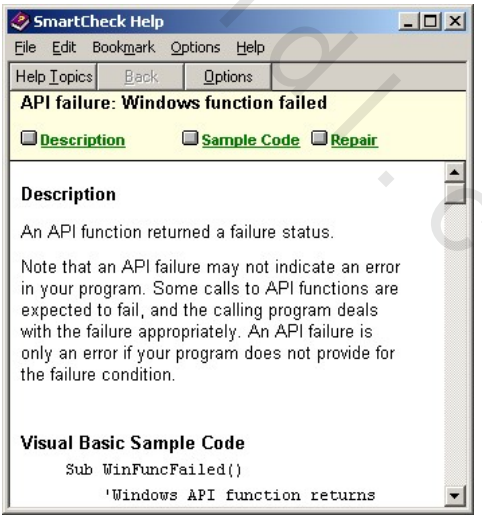
يقوم البرنامج الآن بفتح النافذة result وإذا حدث خطأ يقوم البرنامج بإظهار الرسالة التالية:



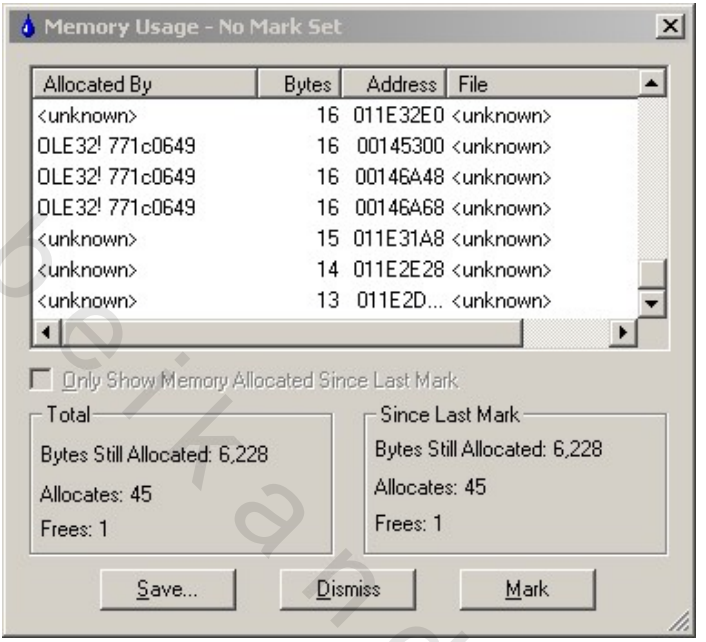
و يمكنك معرفة اسم الدالة المسببة للخطأ ورقم السطر إذا كان الكود الأصلي للبرنامج متوفر عن طريق الضغط على المفتاح إلى يوجد به سهم لأسفل.



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

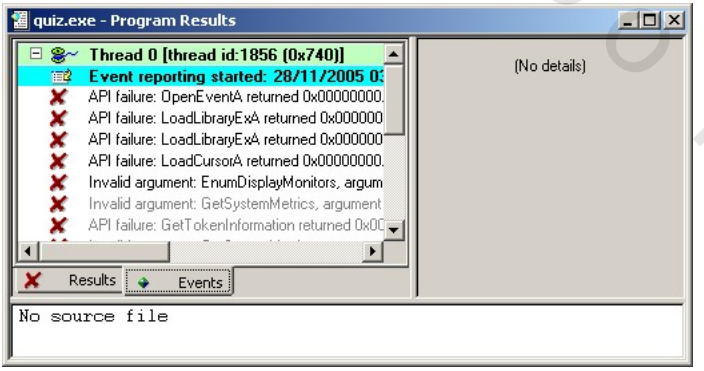


أما المفتاح memory فيمكنك من رؤية الملفات المحملة بالذاكرة عند هذه النقطة وعنوان كل داله تم النداء عليها من الملف



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