Estudio completo de aplicaciones Visual C++ 6.0 (2 parte)

por

AkirA

Información

Programa:	Varios (ir leyendo)
Tamaño:	*El tamaño no importa XDDDDD*
Herramientas:	1. Ollydbg 1.09c
Dificultad:	***NewBie avanzado***

Introducción

Hola amigos!!!! Bienvenidos a la 42 entrega del curso de AkirA.

Desde que hice el tuto sobre VC++ he recibido un montón de preguntas sobre el tema, parece que este tema interesa a mucha gente. Bien, espero que con este tuto aclare conceptos que parece que no quedaron muy claros con el primer tuto.

Cualquier duda escribirme a mi email atalasa@hotmail.com

Nota: si necesitais informacoin sobre Ollydbg buscar en la pagina de Joe Cracker: www.iespana.es/ollydbg esta es la mejor página que hay sobre el tema, de hecho gracias ha ella yo hago todos es tos poryectos en olly

Comentario del Programa

Por supuesto el disclaimer de turno. Vamos a ver, no es que no me haga responsable de la utilización de esta información, es que directamente paso del tema, el único propósito de todo esto es de carácter educativo.

A fin de cuentas, si lo que quieres es crackear un programa pues te bajas el crack y punto, pero si vas a leer este tutorial es porque tu objetivo es aprender. Eso es lo que nos motiva, el comprender como funcionan las cosas o como estan hechas por dentro, y por supuesto el subidón de haberle ganado a un equipo de ingenieros diseccionando un objeto que ellos habian diseñado y del cual no sabemos nada.

.

Manos a la Obra
Comienza el asedio....

Con el compilador de Visual C ++ podemos hacer mucha clase de proyectos y apliaciones. Básicamente las mas importantes se dividen en dos: se puede utilizar las MFC, o se puede hacer un programa sin MFC.

(Hay un tercer tipo, pero sirve para hacer programas en MS-DOS, lo cual.....)

Vamos a hablar de los dos tipos, de cómo diferenciarlos y cómo analizarlos.

Lo primero que vamos a hacer es hablar de los programas hechos con MFC. MFC es un conjunto de objetos (códigos y asistentes) que Microsoft nos dan a los programadores para poder desarrollar aplicaciones de una manera mas cómoda y rápida. Así, usando MFC podemos usar los objetos Cedit, Cbutton, Cstring, etc, etc.

¿ Cómo saber si un programa está hecho con las MFC?

Fácil, los programas hechos con MFC necesitan de varias DLL para poder ejecutarse. Por ejemplo en el tuto sobre visual C++ yo hice un crackme. Cuando le doy a ejecutar, se carga la DLL MFC42.DLL, donde se encontraba la dirección 5f4373f9. Ya quedamos que esa dirección era la que decidía cual era el código del programa principal que debía ejecutarse cuando surgía un evento (pulsar un botón) Si ponemos un breakpoint justo antes de pulsar un botón, rompemos en esa dirección y si damos a f7 podremos trazar dentro de la rutina importante desde la primera línea.

Vamos a ver esto, con un nuevo ejemplo desde el principio para que podais entenderlo mejor.

Cargamos el DriveTest.exe de Wkz_Daedalus y lo primero que debemos ver y muy importante, es lo siguiente. En el EOP siempre, y subrayo lo de siempre (ya sea hecho con con o sin MFC) el código inicial es el mismo para cualquier programa en VC++.

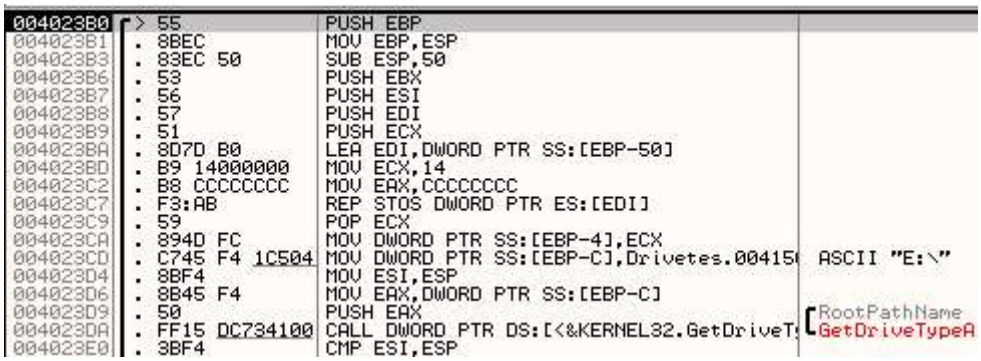
Esto es debido a que al cargar tiene que hacer unas comprobaciones y sobre todo, debe cargar las DLL y este código lo escribe el compilador, no el programador.

Mas cosas: cualquier código que escribimos nosotros, se escribirá siempre por encima del EOP y presumiblemente la gestión de eventos empezara en 401000.

Pero lo más importante es lo siguiente: tanto si el programa tiene MFC, como si no, siempre se llama a GetStartUpInfoA y después a GetModuleHandleA y la primera función CALL que haya después de esta última API, será la llamada a el principio de lo que nosotros ya si hemos programado.

Ahora Run y dejar que cargue tranquilo el programa y se carguen las DLL. Poned el breakpoint en 5f4373f9 y presionad el botón de OK y romperá en nuestra dirección.

Con un par de f7 llegareis a un Push EBP, más abajo vereis GetDriveTipeA, bueno pues ya sabeis en que consiste la gracia de este crackme.



Vamos a aprender mas cosas sobre MFC.

Cargar ahora mi crackme, el del primer tuto de VC++. Era un simple crackme que cogía una cadena y la comparaba con otra.

Si repetis el proceso anterior rompereis dentro de la función que gestiona el evento de que se pulsó el botón Registrar. Vale, lo importante es que vereis que en toda esa función, a pesar de ser muy fácil no se llama a ninguna API.

Eso tiene una explicación: cuando se trabaja con MFC, el propio asistente te da la posibilidad de asociar variables a las cajas de texto para manejarlas de forma sencilla y rápida. Suelen ser dos objetos, o Un Cstring o un Cedit. Este crackme en concreto está hecho con Cstring, y como os podeis imaginar las funciones del Cstring están implementadas dentro de la DLL. Por eso vemos llamadas a funciones 389, 255, etc, etc... esto es muy parecido a cuando en Visual Basic vemos la _vbaStrcmp o _vbaStrcat, etc. Pero Olly no nos pone ningun nombre.

Haced una cosa: cargad el crackme con el IDA y cuando termine de analizar, dais a File -Produce– Create MAP y poneis un nombre a ese archivo.map

Ahora en el Olly debereis tener un Plugin llamado MapConv. Lo pinchais y elegid sustituir comentarios por los del archivo.map Ahora esas CALL 389 ya tiene un nombre. Lo único que debeis hacer es hacer vuestras propias pruebas e ir descubriendo para que sirve cada función.

Por ejemplo:

String (+) es la función que compara dos String, equivale en Visual Basic a _vbaStrCmp, así que de ahora en adelante cualquier string(+) sabréis que sirve para comparar dos Strings XD.

(No pongo más sobre estas funciones, porque estoy haciendo una lista que espero sea grande y definitiva, para un tercer tuto, aunque tu mismo puedes hacerlo XD)

Seguimos aprendiendo cosas sobre las MFC. Estan implementadas sobre DLL y existen varias versiones. Por ejemplo, nosotros hemos visto la MFC42.DLL veamos un ejemplo con otra versión.

Cargamos el VCCrackMe10 Eternal y vemos que carga las MFC40.DLL . Atentos a que también, en el comienzo se llama a las dos APIs de las que hablamos antes.

Si aplicamos el mismo proceso que hicimos en el primer tuto sobre VC++ descubriremos que el salto importante en la versión 4.0 de las MFC está en 61f026f1 en vez de 5f4373f9, como era en 4.2

```
61F026BE|  8B4D 08          |MOV ECX,DWORD PTR SS:[EBP+8]
61F026C1|  FF55 14          |CALL DWORD PTR SS:[EBP+14]
61F026C4|v EB 41            |JMP SHORT MFC40.61F02707
```

Os ayudo: poned breakoint en 401526 y cuando pulseis el botón mirad el valor de la pila XD

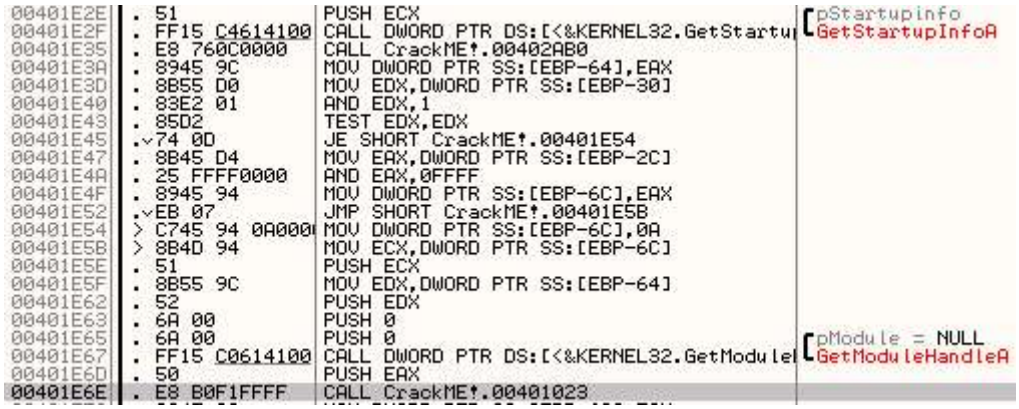
```
00401526|. 55              |PUSH EBP
00401527|. 8BEC            |MOV EBP,ESP
00401529|. 6A FF          |PUSH -1
0040152B|. 68 9A164000    |PUSH VCCrackM.0040169A
00401530|. 50              |PUSH EAX
00401531|. 64:8925 00000( |MOV DWORD PTR FS:[0],ESP
00401538|. 83EC 20         |SUB ESP,20
0040153B|. 894D F0         |MOV DWORD PTR SS:[EBP-10],ECX
0040153E|. 53              |PUSH EBX
0040153F|. 56              |PUSH ESI
00401540|. 8D4D E0         |LEA ECX,DWORD PTR SS:[EBP-20]
00401543|. 68 58304000     |PUSH VCCrackM.00403058          |ASCII "Wrong! Try Again"
00401548|. E8 79030000     |CALL <JMP.&MFC40.#483>
0040154D|. 68 50304000     |PUSH VCCrackM.00403050          |ASCII "Error"
```

Además vereis que las funciones que gestionan nuestro serial son las mismas que antes.

Bueno, segunda parte, analicemos los programas en VC++ que no utiliza MFC.

Cargamos el Crackme! vc++ de Demian.

Fijaos que aquí también al empezar se llama a GetStartUpInfoA y después a GetModuleHandleA. Y que el código normal está por encima de esta dirección.

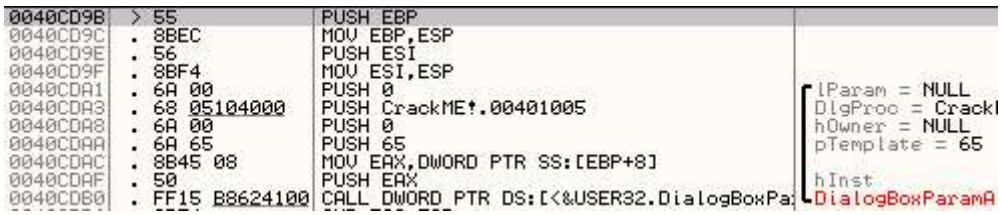


Situaos en GetModuleHandleA. El siguiente CALL es el que llama al programa que sí programa el cracker.

Cuando no se utilizan MFC hay que programar con las APIs verdaderas.

Ahora, veamos un posible método para encontrar el punto en que empieza una función al presionar un botón.

En 401e6e está la call famosa. Trazamos una vez allí con f7 y otro y vemos que llama a DialogParamA.

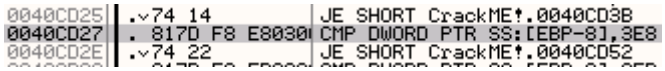


Aquí hay dos posibilidades. Podemos ir a la dirección del DlgProc, osea la función que gestiona los eventos del cuadro de diálogo (401005) y buscar allí el botón, o buscar el botón directamente.

Vereis:

Dad a run y cuando cargue pincad e el botón Window del Olly. Allí vereis que el botón Check tiene como ID 3E8.

vais al desemsamblador y con el derecho search for – all constant, y poneos 3E8 y os saldrá una única línea. Lla pinchais dos veces y vereis que es una comparación con [ebp-8] (en la linea 40cd27 ☺) y que si es verdad salta a 40cd52.



Bien, por cualquiera de los dos métodos hubiesemos llegado hasta aquí, porque lo que ocurre es que en la DlgPrc se mira el evento que se genera. Si es el de se pulso un botóin 201 se compara con el Id de cada botón para saber cual es el que se pulso.

Osea que en 40cd52 esta la línea que nos lleva a la rutina importante del crackme que está en 40ca88 (trazar un poquito)

Pero en fin, como podeis ver, cuando se trabaja sin MFC, se trabajan con APIs directamente, así que cualquier metodo del mundo es válido XD

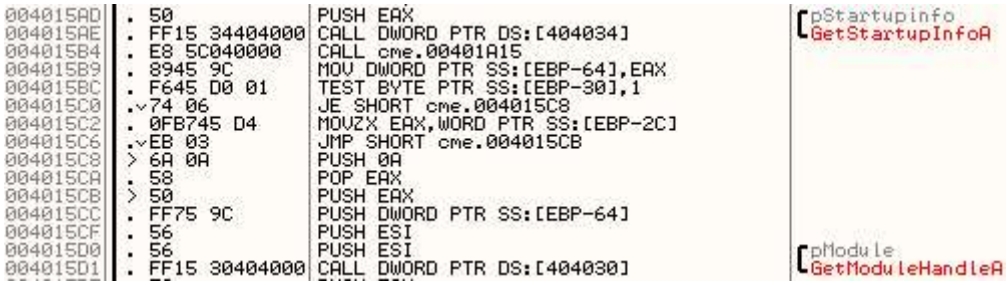
Ahora, la tercera parte del tuto y posiblemente la más importante:

¿Se puede empaquetar un programa hecho en VC++?

Como poder se puede, pero es una grandísima estupidez XD. Veamos porque.

Carguemos el crackme CME.exe de K3nny. Sabemos que es un crackme en VC++ que está empacado. No tengo ni idea de cual es el empaquetador pero es que me da igual.

Sabemos que el principo de cualquier programa en VC++ es igual y que se llaman a dos funciones al empezar a GetStartUpInfoA y después a GetModuleHandleA, bien, pues si dejais que se descomprima, lo único que teneis que hacer es mirar en el código y encontrarlas.



Subir un par de líneas hasta que encontreis el tipico Push EBP y punto XDD el EOP está en 40150f,

(también se puede poner un BPX en las APIs antes de que se desencripte y con un ctrl.+f9 y será mas rápido que buscar XD)

Ahora se dumpea con lo que querais (con el pluging del Olly funciona) y ya está.

La verdad es que este mismo “fallo” le ocurre a Visual Basic.

Cualquier programa hecho en VB empieza con estas dos lineas

Push (valor numérico)
Call ThuRMain (no está bien escrito, mirarlo en cualquier crackme VB)

Y justo encima hay un montón de Jmp[funciones de VB], osea de tipo ff25.

Ademas ese valor numérico apunta a una cadena Ej: VB5 que le dice la Dll a cargar.

El EOP es siempre el Push (valor numérico) y si no está se escribe XD

Yo se porque packer lo digo XDD.

Bueno espero que os haya quedado clar,o como eso de los programa de VC++ y que os hayais divertido con vuestros nuevos conocimientos.

Hasta la proxima ¡!!

Nota:

Bueno espero que hayas aprendido mucho que es de lo que se trata, que te sirva de ejemplo para que tu también puedas hacerlo, desde luego no hay comparado como coger un programa, abrirlo, ver un montón de código por todas partes y manejar lo que otros ingenieros han hecho, tu sólo, por ti mismo.

Bueno , si quieres comentarme algo escribeme a atalasa@hotmail.com

Chao!!

Espero que hayan disfrutado leyendo este tutorial y que les sirva para incrementar sus habilidades, pero recuerden, lean muchos tutoriales, practiquen, estudien y CRACKEAR será mucho más