ESTUDIO DE PROTECCIONES BASICO PARA PRINCIPIANTES

Impartido por Ratón (Nivel principiante)

Nota

Cada capitulo ira acompañado de su crackme correspondiente.No facilitare paginas de donde bajarse herramientas ni enlaces a paginas de crackers, la intención es que busquéis en Internet todo lo necesario. Seguro que encontráis mas paginas de herramientas, tutoriales y utilidades relacionadas con este tema que las que yo pueda deciros.
Con esto solo quiero fomentar vuestro interés, además se que la búsqueda os proporcionara gratas sorpresas.
A todos un saludo.

Capitulo 1

Victima

Abex crackme 5

Herramientas

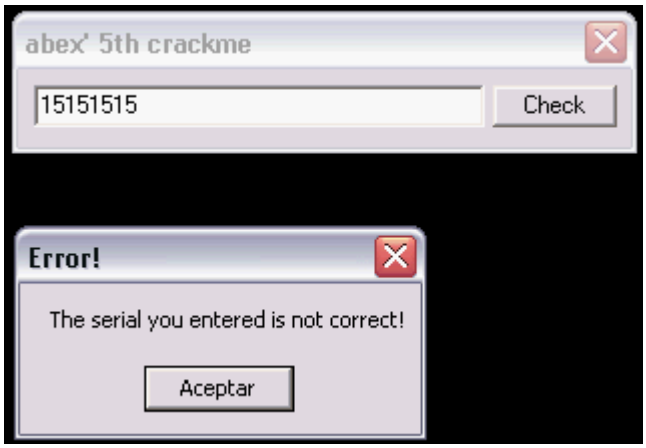
Olly Debugger.
Un mínimo de lógica y ganas de aprender.
Un PC, claro

Objetivo

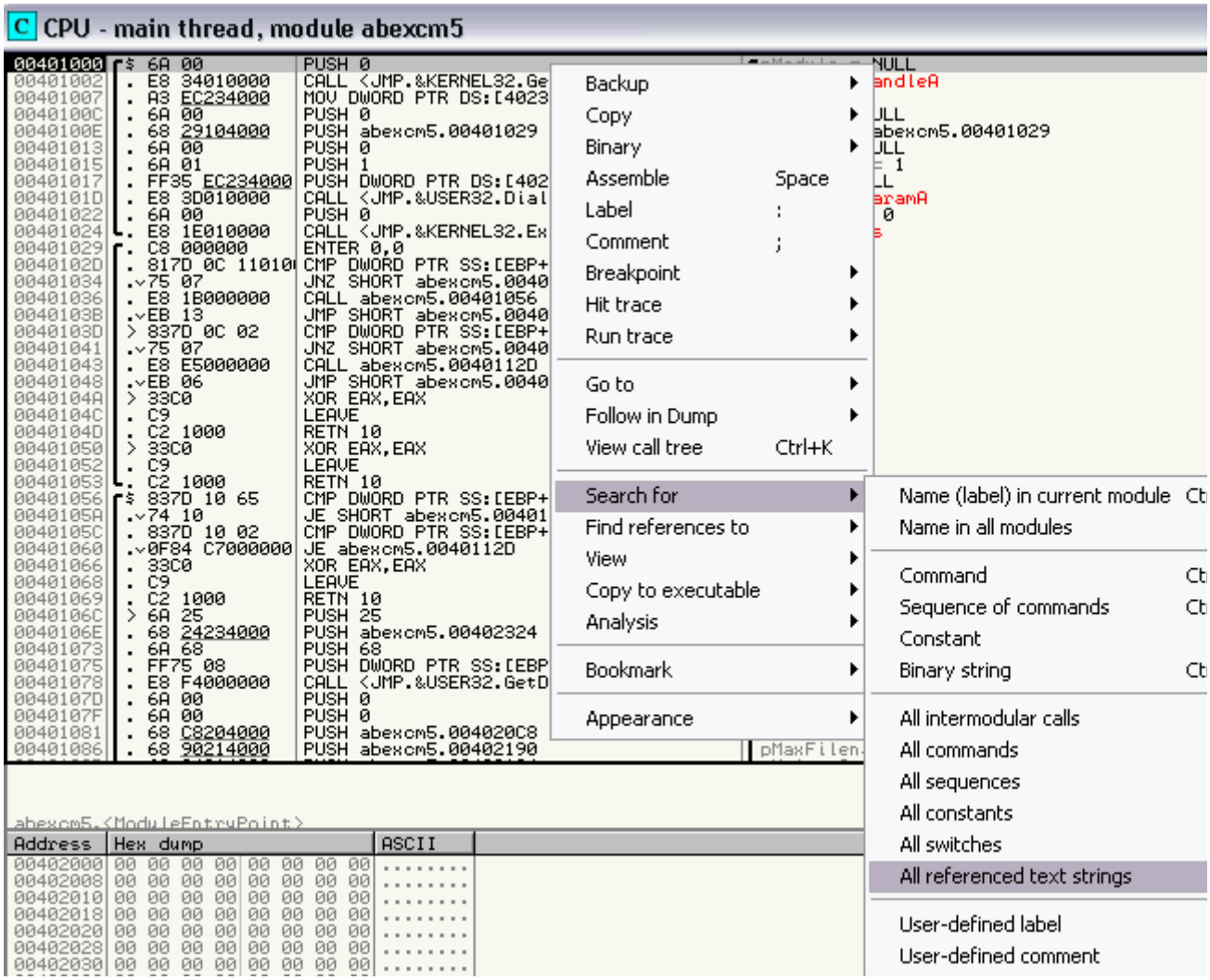
Examinar las rutinas de saltos y como invertirlos para registrarnos sin conocer el serial number del crackme.
Hallar el numero de registro (Nivel facilón).
Familiarizarnos con el Olly Debugger y algunas instrucciones básicas en lenguaje ensamblador.
Ver como no debemos proteger jamás un programa.

Al ataque

Este es un crackme para ir tomando contacto con las herramientas y las primeras instrucciones en ensamblador.
Seria conveniente hacer una copia del crackme por si hacemos algo incorrecto no estropear el ejecutable, y aparte porque en este caso la necesitaremos mas adelante.
Lo primero es ejecutar el crackme.
Vemos que nos pide un numero de registro, introducimos un numero cualquiera, en mi caso 15151515 y pulsamos el botón check.
Como no acerté el serial correcto aparece este cartel (Message Box) diciéndome que como adivino no tengo futuro, que mi numero de serie no es valido, lo sospechaba.
Apuntamos el mensaje, en este caso The serial you entered is not correct.

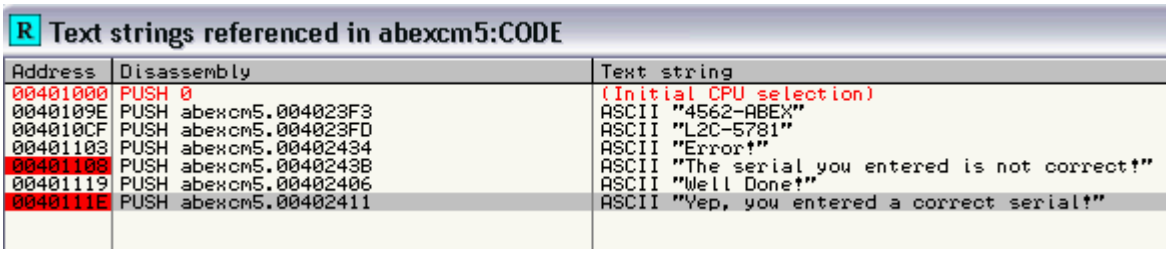


Cerramos el crackme y abrimos el Olly.
File – Open - buscamos nuestro crackme y lo abrimos.
No nos asustamos ante lo que vemos, ya lo entenderemos poco a poco.
Hemos visto que cuando introducimos un serial no valido aparece un cartel (llamado MessageBox) con la frase The serial you entered is not correct, vamos a buscarla.
Click con el botón derecho del ratón en la ventana superior izquierda y vamos a Search for se desplegara un submenú (ver imagen) y hacemos click izquierdo sobre All referenced text strings



Olly Debugger abierto y con nuestro primer crackme cargado. Opción de búsqueda de cadenas de texto (Strings)

Nos aparece esta ventana

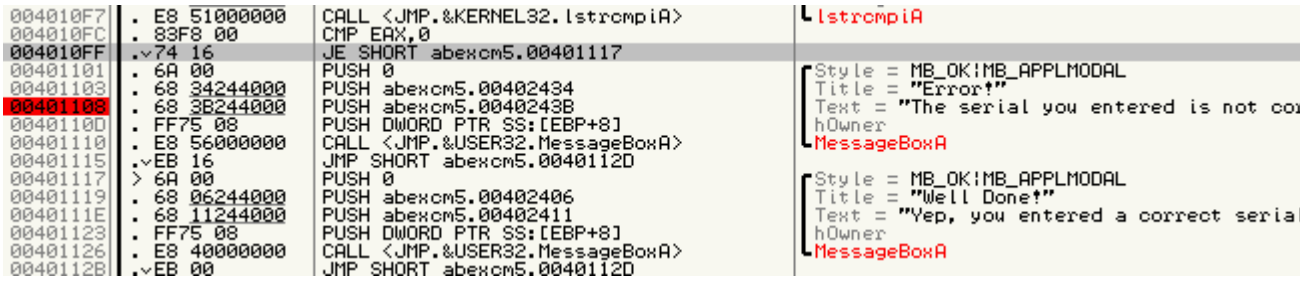


Cadenas de texto del Abex crackme 5 (Strings)

En la cual podemos ver varias frases entre ellas nos llaman la atención 2: The serial you entered is not correct (la que buscamos) y
Yep, you entered a correct serial (la que nos informa que hemos introducido un serial valido y por tanto el programa esta registrado)
Hay un par de cadenas (strings) de las que luego hablaremos ya que ahora no interesan para el propósito de este tutorial.

A la izquierda de la ventana podemos ver una serie de números asociados a cada cadena de texto. Estos números nos marcan la posición o dirección (Address) en la que se hallan cada una de las cadenas de texto dentro del ejecutable. En la parte central esta la instrucción en ensamblador que corresponde a esa dirección. En la parte derecha vemos la cadena de texto.
Tenemos pues: (imagen anterior)
Dirección 00401108 Instrucción PUSH Cadena de texto The serial you entered is not correct.

Hacemos doble click izquierdo sobre la cadena que nos interesa para que automáticamente nos mande a la dirección que ocupa en el ejecutable y aparecemos aquí (marcado en rojo en la imagen siguiente).



Abex crackme ventana principal de Olly Buscando la dirección de la cadena de texto (Text String)

Podemos ver que en la dirección 00401108 se encuentra nuestra frase y unas líneas más abajo (0040111E) la frase que nos aparecerá cuando introduzcamos el serial valido.

Análisis detallado de esta parte del código.

004010F7 - CALL – Llamada a un procedimiento, en este caso lstrcmpiA
004010FC - CMP - Compara algo, en este caso compara EAX con cero
004010FF – JE ---- Salta si es igual y en este caso saltaría a la dirección 00401117 si se cumpliera la condición de la
instrucción anterior (EAX = 0)
00401110 – CALL – MessageBoxA esta llamada hace aparecer la ventanita con el texto The serial … not correct

Explicación

CALL-> llamada a un procedimiento el cual tendrá alguna repercusión en la siguiente instrucción.
CMP-> compara el valor cargado en EAX con 0 (cero). El valor de EAX se genero en el CALL anterior.
JE-> si EAX es igual a 0 (cero) salta a la dirección 00401117 (JE salta si es igual).

El resultado si miramos la dirección del salto (00401117) es que si se cumple esa condición (EAX = 0) Saltamos a la zona
donde empieza a generarse la ventana que nos acepta como registrados y que nos llevaría hasta
0040111E Yep, you entered a correct serial.

Si EAX no fuese igual a 0 el programa continuaría con su siguiente instrucción (0040110D) y llegaríamos a
00401110 – CALL – MessageBoxA que es el que hace aparecer la ventana de no registrado.

Uno de los objetivos de este capitulo es que quede claro lo que acabamos de ver y comprendáis que hacen estas tres instrucciones.

Resumen de lo visto hasta ahora.
1 -Abrimos el crackme intentamos registrarnos con un serial cualquiera, lo chequeamos y nos salta una ventana (MessageBox) con unas líneas escritas (Strings) que nos indica que no es ese el código de registro (Serial).

2 -Cargamos el crackme (Ejecutable) en el Olly y buscamos las referencias a la cadena de texto The serial you entered is not correct.

3 -Una vez encontrada nos desplazamos a la dirección que ocupa dentro del ejecutable e inspeccionamos el código fijándonos en las comparaciones y saltos condicionales

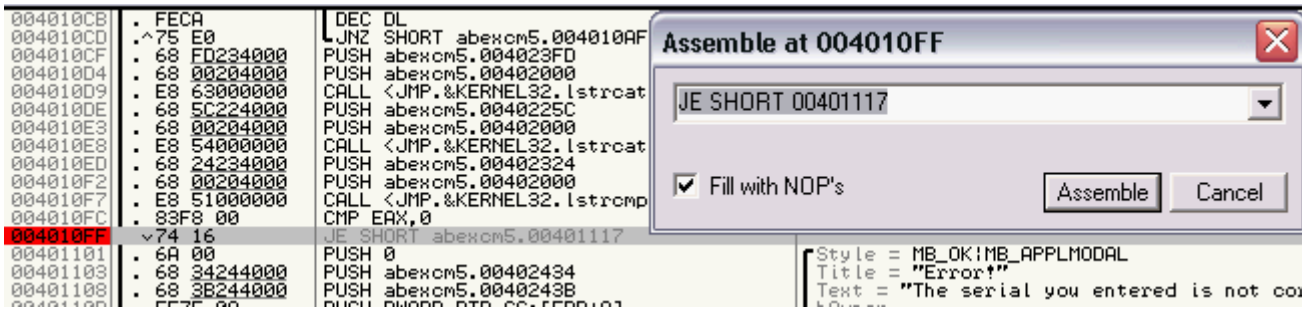
próximos a dicha dirección, y buscamos el salto que decide si estamos o no estamos registrados.

Ahora tenemos dos caminos: cambiar el valor del salto para que nos lleve a la zona donde salta la MessageBox que nos muestra el cartel de registrados Yep, you entered a correct serial (parchearlo invirtiendo el salto) o encontrar el serial valido.
Veamos las dos maneras.

1 - Parcheando.
Los maestros crackers (que la fuerza les acompañe) nos dicen que parchear debería ser la última opción, pero debéis aprender como hacerlo ya que nos servirá para cuando no seamos capaces de encontrar el serial valido (bueno también para alguna otra cosa más).
Aprender a invertir los saltos nos servirá para eliminar NAGs, esas molestas ventanas que aparecen al principio de algunos programas indicando por ejemplo que es una versión trial, shareware o algo por el estilo.
El tema de las NAGs lo veremos en el capitulo 2.

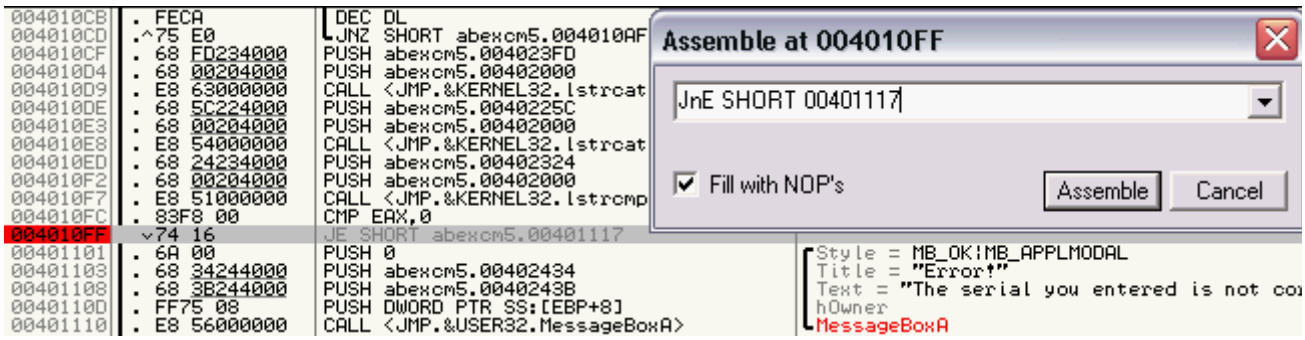
Invirtiendo el valor del salto.
Llegados al punto donde sabemos cual es el salto que decide sobre si estamos registrados o no (4010FF) lo mas lógico seria evitarlo y saltar directamente a la parte del código donde se genera la MessageBox que nos acepta como registrados (00401117).
El salto nos dice: JE salta si es igual entonces si no fuera igual no saltaría (como llegaste a esa deducción Sherlock ¿?)

Lo que haremos será cambiar JE salta si es igual por su contrario JNE salta si no igual.
Nos colocamos encima de la instrucción 00401108 que es el salto que queremos cambiar y doble click izquierdo y
aparece esta ventanita



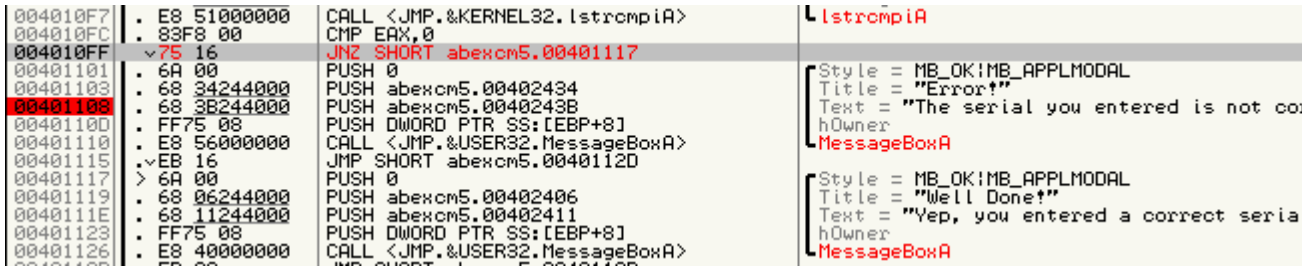
Ventana de ensamblado

Cambiamos la instrucción



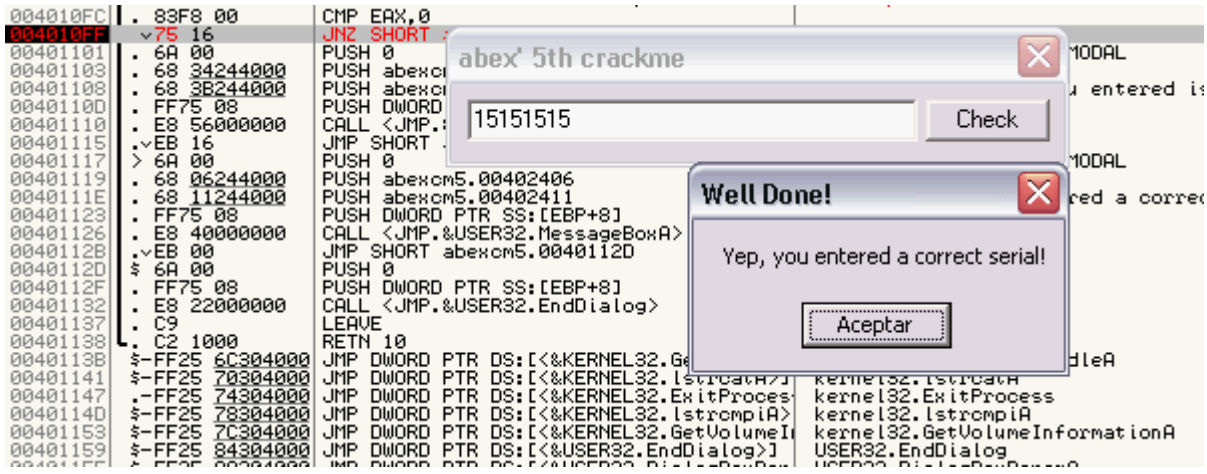
Cambiando la instrucción

Pulsamos el botón Assemble



Cambio realizado

Ya tenemos cambiado el salto no ¿? Pues si y no, me explico: lo único que hemos hecho es cambiar el salto en memoria pero no en el ejecutable, para que me entendáis: no hemos escrito nada en el ejecutable.

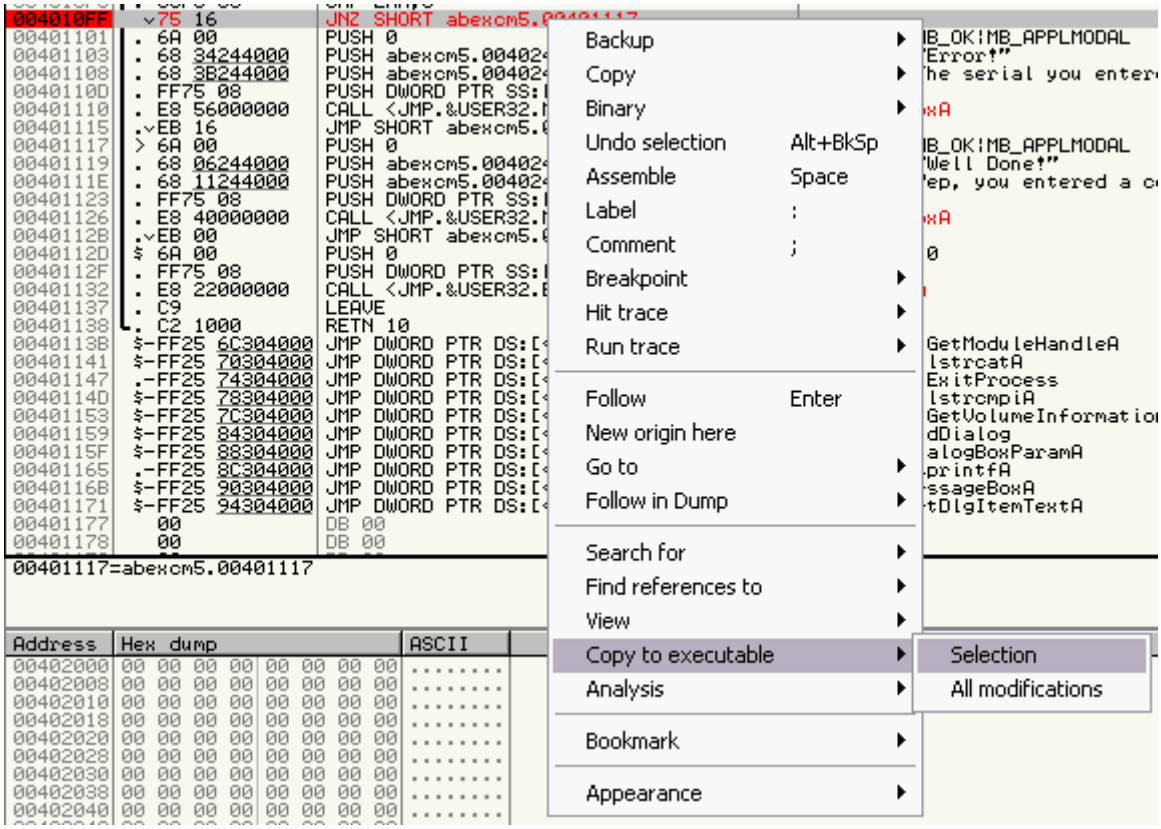


Cambio realizado en memoria lo vemos si corremos el programa pulsando F9 en el Olly

Si corriéramos el ejecutable según lo tenemos sin cerrar el Olly (Debug – Run o pulsando la tecla F9) nos aparecería el cartel de registrado, pero al cerrar el Olly y abrir el crackme si volvemos a introducir un numero cualquiera nos aparecería el cartel malo de no registrados, puesto que no hemos grabado el cambio de JE por JNE (JNZ escribió el Olly, significa lo mismo) en el ejecutable, insisto solo lo cambiamos en memoria.

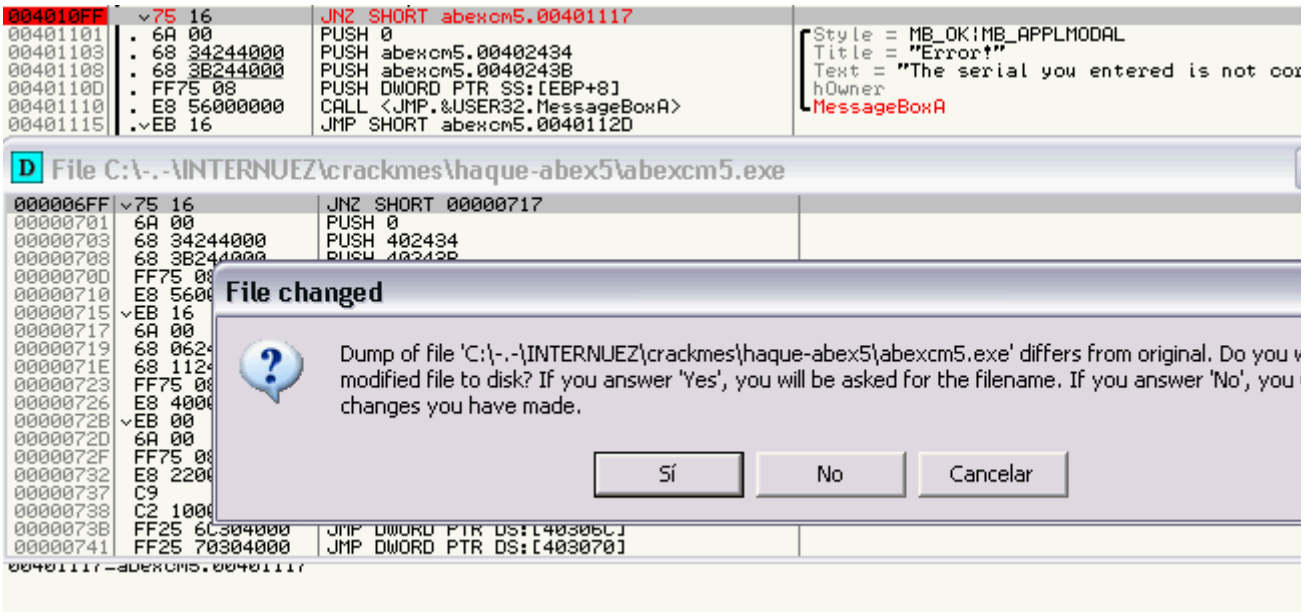
Olly nos da opción de probar en memoria los cambios sin necesidad de tocar el ejecutable si no queremos.

Tenemos que escribir los cambios en el crackme,
Como hacerlo ¿? pues muy sencillo, veréis por ahí tutoriales de otros crackers (grandes crackers) que utilizan editores hexadecimales para hacer los cambios de código en los ejecutables.
No seguiremos ese camino.
Nosotros tenemos a nuestro amigo Olly que nos hace casi todo sin tener que recurrir a programas externos.
Click derecho encima del salto al que hemos cambiado el valor, vamos a Copy to executable y en este caso elegimos Selection pues solo queremos cambiar esta línea.
Si hubiera mas líneas que cambiar adivinad la opción que escogeríamos, si la otra, la que queda: All modifications



Nos disponemos a efectuar el cambio de forma permanente en el ejecutable

Click en Selection y aparecen ventanas, cerramos la primera y aceptamos la segunda con click en Si.
Con esto sobrescribimos el ejecutable cambiando el salto.



Aceptando los cambios que nos dejaran escrito el cambio en el ejecutable de forma permanente

Cambios realizados. Ahora cualquier numero o letra (cadena de texto) que introduzcamos para registrarnos será valida.
Hemos hecho que compare nuestra cadena y si no es igual (JNE / JNZ) salte y nos registre.
Se llama invertir el salto
Antes el crackme saltaba solo si era igual (JE) la cadena introducida por nosotros que la cadena que guardaba el crackme para comparar y registrarnos (el numero de registro valido), ahora nos registrara siempre a no ser que tengamos la "suerte" de acertar el serial que esconde el crackme en sus tripas.

2 - Buscando el arca perdida: número de serie valido.

Esta es la forma de crackear que deberíamos probar primero pues es la que nos dirá el serial correcto del crackme y nos evitaríamos parchear el programa. Como dicen los maestros "la forma elegante".
Hay veces que el serial esta encriptado y como nuestros conocimientos por ahora no son muy amplios que digamos pues parchear el salto nos sacara del apuro.

Empezamos.

Si como dije habéis hecho una copia del crackme la utilizaremos ahora.

La abrimos con el Olly, al ser una copia nueva el salto no esta cambiado, trabajamos pues con el crackme original.

Ya tenemos localizado el salto y las direcciones en el ejecutable donde se generan los MessagexBox bueno y malo (los crackers llaman a esto zona de chico bueno y chico malo).

Revisémosla otra vez

004010F7 - CALL – Llamada a un procedimiento, en este caso lstrcmpiA

004010FC - CMP - Compara algo, en este caso compara EAX con cero

004010FF – JE – Salta si es igual y en este caso saltaría a la dirección 00401117 si se cumpliera la condición de la instrucción anterior (EAX = 0)

00401110 – CALL – MessageBoxA esta llamada hace aparecer la ventanita con el texto The serial … not correct

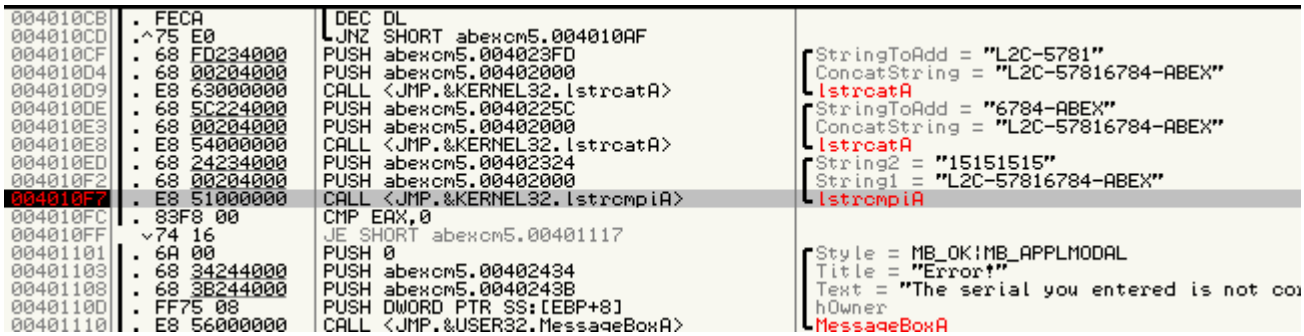
Vemos que antes de la comparación de EAX hay un CALL, nos ponemos sobre el (004010F7) y pulsamos la tecla F2, observamos que se pone en rojo, acabamos de poner nuestro primer Breakpoint. (BP)
Que significa esto ¿? Pues que cuando el programa se ejecute (F9) al llegar a esa instrucción parara.

Por que hemos hecho esto ¿?
Por que queremos que pare el programa en ese CALL y no en otra parte del código ¿?
Pues por un poco de lógica: si la instrucción siguiente a ese CALL es la comparación de EAX y luego viene el salto que cambiamos anteriormente pues intuimos que algo se debe de estar cociendo en ese CALL y vamos a ver que es.

Con el Breakpoint puesto en el CALL 004010F7 pulsamos F9 (Debug run) y arrancamos el ejecutable (A veces aparece minimizado en la barra de tareas otras veces salta a primer plano).
Introducimos un numero de registro (yo sigo con mi 15151515) pulsamos el botón check y Olly para en la dirección del CALL donde habíamos puesto el Breakpoint.

Fijaros en la diferencia entre las dos imágenes siguientes

Breakpoint marcado sin correr el ejecutable antes de pulsar F9



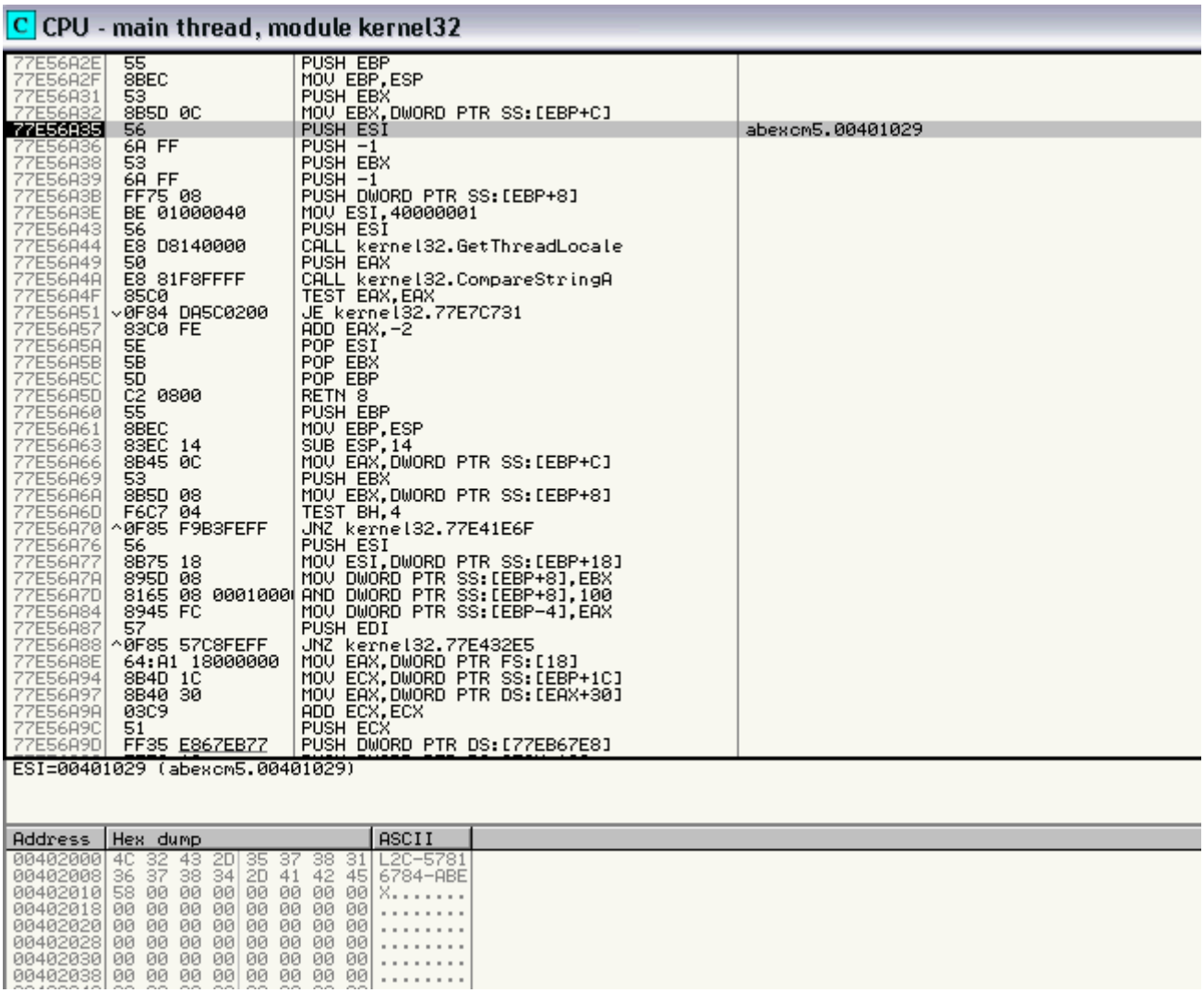
Programa en ejecución. Al introducir el serial 15151515 y pulsar check para Olly en el CALL Observad la ventana derecha del Olly (registers)

La diferencia es clara verdad ¿? Aparecen mas Strings, estas Strings estaban en la ventana que nos mostraba las Strings referenced al principio de este capitulo, pero dije que no le prestáramos atención puesto que como ahora estaréis imaginando pertenecen al serial valido para resolver este crackme, si lo hubiera dicho al principio el ejercicio habría dejado de tener interés y no hubiéramos visto como llegar hasta aquí que es lo que nos interesa para ir comprendiendo este tipo de protecciones y lo sencillo que es burlarlas.

Asegurémonos.
Con el CALL donde ha parado Olly sombreado pulsamos F7 (Debug – Step into) una sola vez para entrar en el CALL e investigar (esto se conoce como trazar o tracear)

Al pulsar F7 hemos entrado en el CALL en la dirección 77E56A2E.

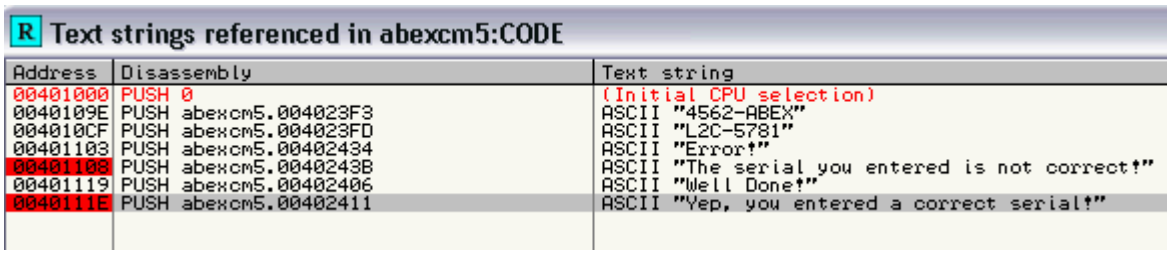
Pulsamos ahora F8 (Debug – Step over) para ir bajando línea por línea (cada vez que pulsemos F8 avanzaremos una línea de código) desde 77E56A2E hasta 77E56A35 por ejemplo y vemos en la ventana derecha del Olly (Registros) nuestro numero 15151515 y otra cadena de letras y números L2C-57816784-ABEX.
En sucesivos capítulos veremos que hacen F7 y F8 mas detenidamente, no os preocupéis, por el momento quedaros con los pasos necesarios para llegar hasta la solución de este crackme, pues son pasos clave para la resolución de muchos de ellos.



Vemos el serial valido y nuestro serial en la ventana de registros

Dentro de este CALL el programa mueve y compara nuestra cadena (15151515) con el serial correcto.
Si, ese es el número de registro de este crackme: L2C-57816784-ABEX

Cualquier cracker experimentado lo hubiera visto nada mas abrir el programa (incluso yo lo vi), pero parto de la base que este curso es para gente que nunca se ha acercado al tema del crack, reversing o como queráis llamarlo, aparte lo que nos importa es seguir estos pasos para saber como llegar a este punto, además no encontrareis protecciones tan facilitas cuando os estudiéis la protección de un programa comercial (o quizás si ¿?).



Miramos de nuevo las strings references Vemos el serial valido separado Estuvo todo el tiempo delante

de nuestras narices

Aquí el serial esta separado, pero en el CALL donde nos paro el Breakpoint lo vemos unido y tal y como debemos introducirlo en la casilla de registro.
Cerramos el Olly y abrimos el crackme para registrarlo con nuestro número.
Lo escribiremos sin las comillas. L2C-57816784-ABEX
Pulsamos check y…

Registrado Serial encontrado

Bien espero haberlo explicado al menos decentemente para los que se acercan por primera vez a este tema, he intentado darlo lo mas masticado posible aun a riego de parecer pesado pues esta será la base desde la que partiremos para siguientes capítulos.

Intentad resolverlo sin el tutorial, es la mejor forma de comprobar si os vais quedando con lo explicado.
Buscad paginas de crackmes e intentadlo con crackmes de este tipo (Serial crackme)
Practicad e intentad comprender lo que hemos visto aquí: comparaciones y saltos.
Leed a los maestros.
Bajad al menos el DeDe, Peid y W32 dasm (herramientas).

Gracias
A todos los crackers y programadores de los cuales he aprendido y sigo aprendiendo.
A los creadores de crackmes
En especial y sin menospreciar a nadie a Ricardo Narvaja por su aportación y su trabajo sobre el estudio de las protecciones y sus tutoriales en castellano y a Makkakko por sus tutoriales con Olly Debugger (Recomendados 100%).
A ti que me estas leyendo.

Ratón Enero 2004