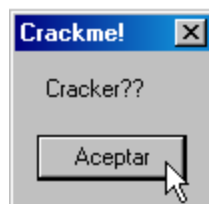




<i>Crackme</i>	<i>Crackme DRm</i>
Misión	-Remover Nag -Activar button -Registrarse
Compilado	Visual Basic 6.0
Herramientas	Olly 1.10 - RDG 0.7.5 - HxD V1.7.7.0 (Plugin Olly Hide Debugger)
Sistema Operativo	Windows Xp SP3
Cracker	QwErTy
Dedicado a	RICNAR / CLS / SoftDat Newzombie / DavicoRm y a Ismael Pérez

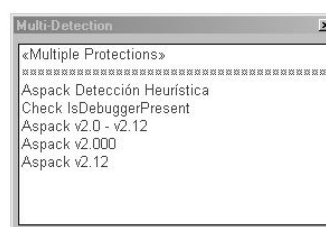
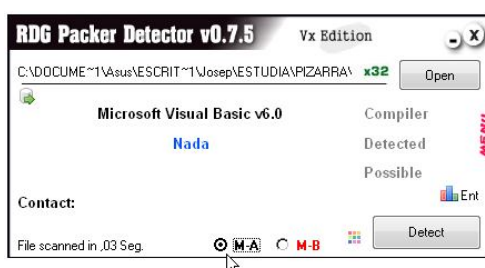
ESTUDIANDO A LA VÍCTIMA

Hacemos una copia del Crackme (Siempre es importante trabajar con una copia), lo abrimos y nos aparece una ventana en la que observamos; el nombre de la misma "Crackme!", un mensaje "Cracker??", y un button "Aceptar", le damos a "Aceptar" y nos saca fuera.



CONTINUAMOS ESTUDIANDO LA VÍCTIMA

Abrimos el Crackme con el RDG

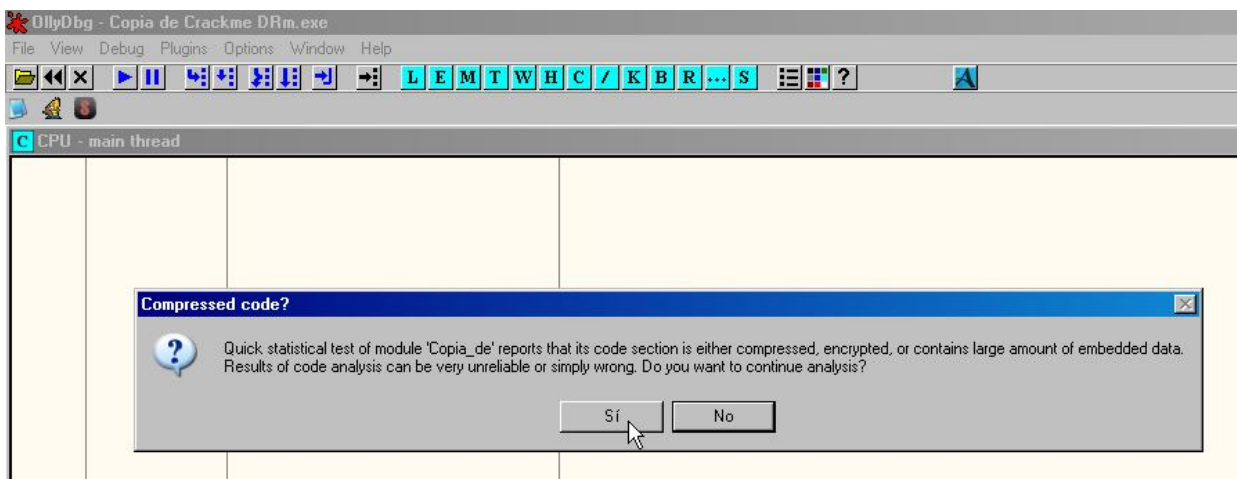


Ya sabemos algo más, el RDG Packer Detector nos informa en el escaneo rápido "M-A", que el crackme está compilado en Visual Basic V6.0, y en el escaneo profundo "M-B" de un compilado en Visual Basic 6.0 Código Nativo, de un Aspack v2.12 detectado y de un posible Check IsDebuggerPresent.

Bien...., pues lo tendremos en cuenta.

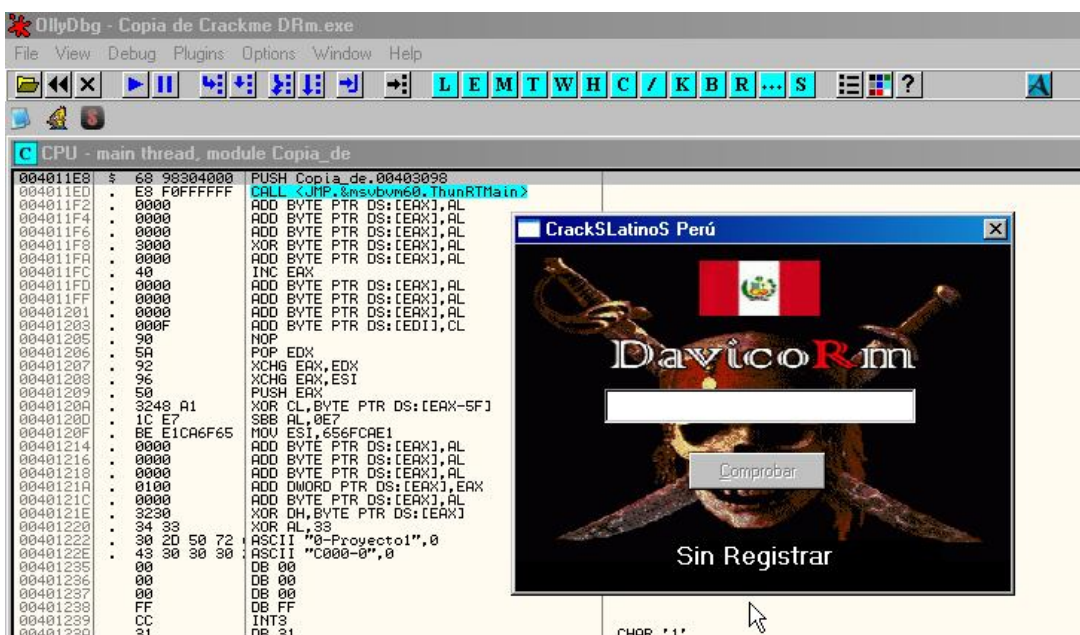
VAMOS A POR ELLA

Salimos del RDG y abrimos el crackme con Olly (importante: con los plugins antiloquetengamos desactivados), y nos sale el cartelito de "Compressed code?", avisándonos de que el crackme puede estar comprimido, encriptado, bla.bla.bla.... le decimos que "Sí" para que continúe con el análisis

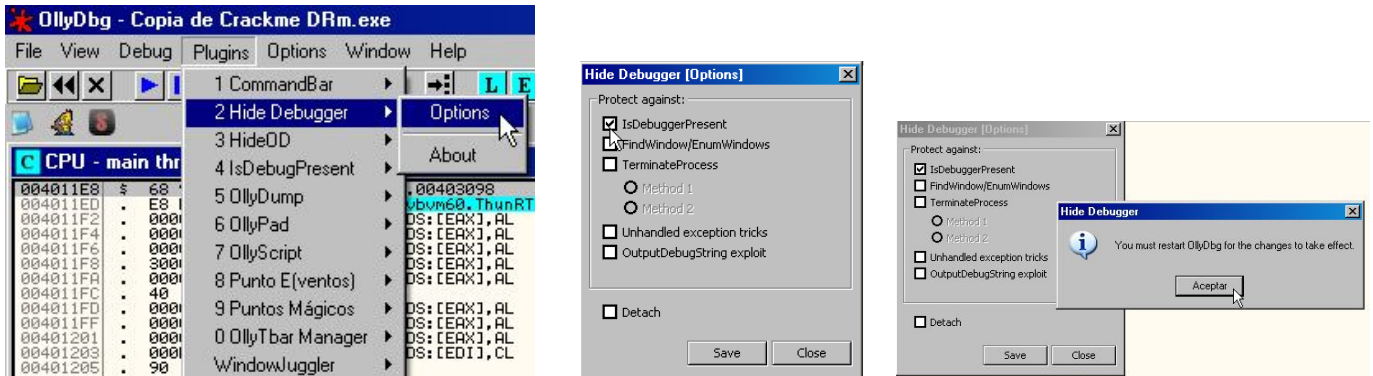


Olly lo analiza y abre correctamente. Además nos fijamos que empiece con un "PUSH" seguido de una "CALL" (detalle característico de los Visual Basic).

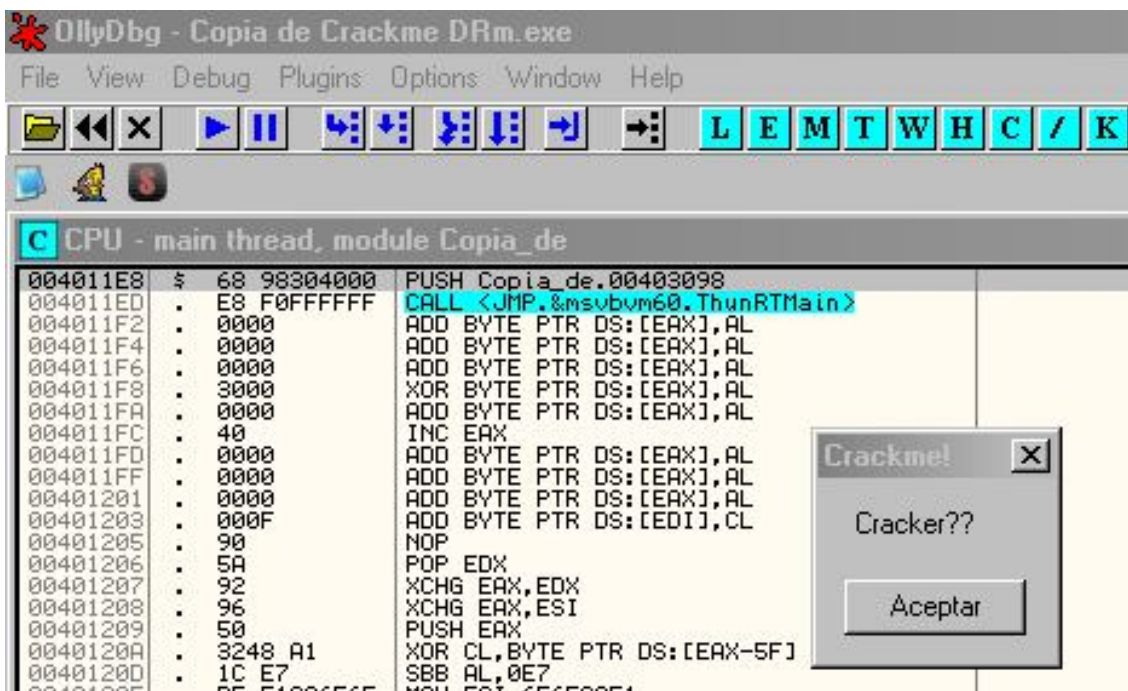
damos run (F9) para comprobar que el crackme corre y nos aparece la primera sorpresa



Que cosa más rara, cuando ejecutamos el crackme fuera de Olly nos mostró una ventana totalmente diferente a esta, aquí pasa algo..... mi instinto de cracker me dice que este crackme tiene algo metido dentro que detecta que lo estamos debuggeando. Vamos a hacer una prueba, vamos a “**Plugins - Hide Debugger**” y activamos la casilla “**IsDebuggerPresent**” le doy a “Save” para guardar cambios, después a “Close” y a “Aceptar”, Ahora salimos totalmente de Olly para que los cambios surtan efecto y volvemos a cargar de nuevo el crackme con Olly (ahora con el plugin IsDebuggerPresent activado).

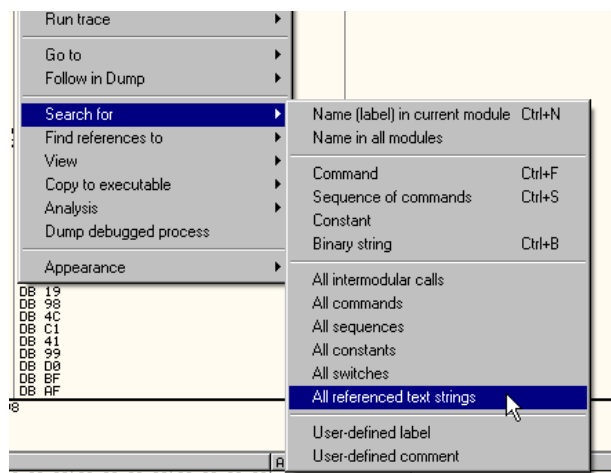


Damos “Run” para que corra y comprobamos que ahora Olly nos muestra el crackme de igual forma que cuando lo corremos fuera de el.



Osea que nuestra intuición queda demostrada, por tanto empecemos con nuestra **PRIMERA MISIÓN: Conseguir que al ejecutar el crackme fuera de Olly no aparezca esta Nag y nos muestre la segunda ventana.**

Continuamos, reiniciamos Olly, click derecho y “Search for - All referenced text strings”



y encontramos de momento dos referencias que nos interesan: "Cracker??" y "Crackme!".

NOTA: Antes de llegar a ellas tambien vemos una referencia a "IsDebuggerPresent" pero este tema ya lo hemos detectado y solucionado al activar el Plugin.

Text strings referenced in Copia_de..text		
Address	Disassembly	Text string
004039F4	ASCII "kernel32",0	
00403904	ASCII "IsDebuggerPresent"	
00403914	ASCII "t",0	
00403918	DD Copia_de.004039F4	ASCII "kernel32"
0040391C	DD Copia_de.00403904	ASCII "IsDebuggerPresent"
00403964	UNICODE "Muy bien"	
00403974	UNICODE 0	
0040397C	UNICODE "DavicoRm"	

No nos entretengamos y sigamos,

00403B84	ASCII "kObj",0	
00403B8C	ASCII "vbaStrMove",0	
00403D49	MOV DWORD PTR SS:[EBP-74],Copia_de.0040	UNICODE "DavicoRm !?"
00403D71	MOV DWORD PTR SS:[EBP-64],Copia_de.0040	UNICODE "Muy bien"
00403DC9	PUSH Copia_de.00403998	UNICODE "Registrado"
00403E10	MOV DWORD PTR SS:[EBP-64],Copia_de.0040	UNICODE "Perdido??"
00403F46	PUSH Copia_de.004039E0	UNICODE "Sin Registrar"
00403F98	MOV DWORD PTR SS:[EBP-74],Copia_de.0040	UNICODE "DavicoRm"
00403FAA	MOV DWORD PTR SS:[EBP-64],Copia_de.0040	UNICODE "Vamos NewBiet! Tu Puedes Resolverme"
00404003	MOV DWORD PTR SS:[EBP-64],Copia_de.0040	UNICODE "PeruHa"
00404020	MOV DWORD PTR SS:[EBP-64],Copia_de.0040	UNICODE "okCLS"
0040406F	MOV DWORD PTR SS:[EBP-74],Copia_de.0040	UNICODE "Crackme!"
00404081	MOV DWORD PTR SS:[EBP-64],Copia_de.0040	UNICODE "Cracker??"
00404168	ASCII "FB",0	
0040416C	ASCII "7R",0	

"Cracker??" y "Crackme!" son las referencias que salen al abrir el Crackme.exe fuera de Olly. Pues, nos posicionamos con el cursor sobre una de ellas, yo lo hago sobre "Crackme!" que es el título de la ventana, dos clicks Izquierdos de ratón y aparecemos en la address "0040406F"

CPU - main thread, module Copia_de			
00404030	8B00	MOV EDX, EAX	
00404032	B9 24504000	MOV ECX, Copia_de.00405024	
00404037	FF15 08104000	CALL DWORD PTR DS:[C:\msvbvm60__vbaVarM	msvbvm60.__vbaVarMove
0040403D	E8 EF8FFFFF	CALL DWORD PTR SS:[EBP-A0], EAX	
00404042	8905 60FFFF	MOV DWORD PTR SS:[EBP-A0], EAX	
00404048	FF15 1C104000	CALL DWORD PTR DS:[C:\msvbvm60__vbaSetS	msvbvm60.__vbaSetSystemError
0040404E	8B85 60FFFF	MOV EAX, DWORD PTR SS:[EBP-A0]	
00404054	85C0	TEST EAX, EAX	
00404056	75 6E	JNE SHORT Copia_de.004040C6	
00404058	B8 04000200	MOV EAX, 00020004	
0040405D	8D55 84	LEA EDI, DWORD PTR SS:[EBP-7C]	
00404060	8D4D C4	LEA ECX, DWORD PTR SS:[EBP-3C]	
00404063	8945 AC	MOV DWORD PTR SS:[EBP-54], EAX	
00404066	895D A4	MOV DWORD PTR SS:[EBP-5C], EAX	
00404069	8945 BC	MOV DWORD PTR SS:[EBP-44], EAX	
0040406C	895D B4	MOV DWORD PTR SS:[EBP-4C], EAX	
0040406F	C745 8C AC39	MOV DWORD PTR SS:[EBP-74],Copia_de.0040	UNICODE "Crackme!"
00404076	8970 84	MOV DWORD PTR SS:[EBP-7C], EDI	
00404079	FFD6	CALL ESI	
0040407B	8D55 94	LEA EDI, DWORD PTR SS:[EBP-6C]	
0040407E	8D4D D4	LEA ECX, DWORD PTR SS:[EBP-2C]	
00404081	C745 9C 8839	MOV DWORD PTR SS:[EBP-64],Copia_de.0040	UNICODE "Cracker??"
00404088	8970 94	MOV DWORD PTR SS:[EBP-6C], EDI	
0040408B	FFD6	CALL ESI	
0040408D	8D55 A4	LEA EDI, DWORD PTR SS:[EBP-5C]	
00404090	8D45 B4	LEA ECX, DWORD PTR SS:[EBP-4C]	
00404093	52	PUSH EDX	
00404094	8D4D C4	LEA ECX, DWORD PTR SS:[EBP-3C]	
00404097	50	PUSH EAX	
00404098	51	PUSH ECX	
00404099	8D55 D4	LEA EDI, DWORD PTR SS:[EBP-2C]	
0040409C	6A 00	PUSH 0	
0040409E	52	PUSH EDX	
0040409F	FF15 2C104000	CALL DWORD PTR DS:[C:\msvbvm60\rtchsgBox	msvbvm60.rtcHsgBox
004040A5	8D45 A4	LEA ECX, DWORD PTR SS:[EBP-5C]	
004040A8	8D4D B4	LEA ECX, DWORD PTR SS:[EBP-4C]	
004040AB	50	PUSH EAX	
004040AC	8D55 C4	LEA EDI, DWORD PTR SS:[EBP-3C]	
004040AF	51	PUSH ECX	
004040B0	8D45 D4	LEA EDI, DWORD PTR SS:[EBP-2C]	
004040B3	52	PUSH EDX	
004040B4	50	PUSH EAX	
004040B5	6A 04	PUSH 4	
004040B7	FF15 0C104000	CALL DWORD PTR DS:[C:\msvbvm60__vbaFree	msvbvm60.__vbaFreeVarList
004040BD	83C4 14	ADD ESP, 14	
004040C0	FF15 10104000	CALL DWORD PTR DS:[C:\msvbvm60__vbaEnd	msvbvm60.__vbaEnd
004040C6	C745 FC 0000	MOV DWORD PTR SS:[EBP-4], 0	
004040C8	7C 00000000	MOV ECX, 0	

si observamos atentamente:

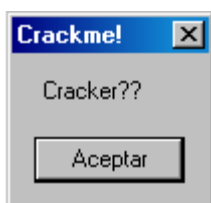
CPU - main thread, module Copia_de			
00404030	8B 00	MOV EDI, EAX	
00404032	B9 24504000	MOV ECX, Copia_de.00405024	
00404037	FF15 08104000	CALL DWORD PTR DS:[&msubvm60.__vbaVarMove	msubvm60.__vbaVarMove
0040403D	E8 EEF8FFFF	CALL Copia_de.00403930	
00404042	8985 60FFFFFF	MOV DWORD PTR SS:[EBP-A0], EAX	
00404043	FF15 1C104000	CALL DWORD PTR DS:[&msubvm60.__vbaSetS	msubvm60.__vbaSetSystemError
0040404E	8B85 60FFFFFF	MOV EAX, DWORD PTR SS:[EBP-A0]	
00404054	85C0	TEST EAX, EAX	
00404056	JNZ SHORT Copia_de.004040C6		
00404058	B8 04000200	MOV EAX, 00020004	
0040405D	8D55 84	LEA EDX, DWORD PTR SS:[EBP-7C]	
00404059	8D4D C4	LEA ECX, DWORD PTR SS:[EBP-3C]	
00404063	8945 0C	MOV DWORD PTR SS:[EBP-54], EAX	
00404066	895D A4	MOV DWORD PTR SS:[EBP-5C], EBX	
00404069	8945 BC	MOV DWORD PTR SS:[EBP-44], EAX	
0040406C	895D B4	MOV DWORD PTR SS:[EBP-4C], EBX	
0040406F	C745 8C AC3A	MOV DWORD PTR SS:[EBP-74], Copia_de.0040	UNICODE "Crackme!"
00404076	897D 84	MOV DWORD PTR SS:[EBP-7C], EDI	
00404079	FFD5	CALL ESI	
0040407B	8D55 94	LEA EDX, DWORD PTR SS:[EBP-6C]	
0040407E	8D4D D4	LEA ECX, DWORD PTR SS:[EBP-2C]	
00404081	C745 9C 883A	MOV DWORD PTR SS:[EBP-64], Copia_de.0040	UNICODE "Cracker??"
00404088	897D 94	MOV DWORD PTR SS:[EBP-6C], EDI	
0040408B	FFD5	CALL ESI	
0040408D	8D55 A4	LEA EDX, DWORD PTR SS:[EBP-5C]	
00404090	8D45 B4	LEA ECX, DWORD PTR SS:[EBP-4C]	
00404093	52	PUSH EDX	
00404094	8D4D C4	LEA ECX, DWORD PTR SS:[EBP-3C]	
00404097	50	PUSH EAX	
00404098	51	PUSH ECX	
00404099	8D55 D4	LEA EDX, DWORD PTR SS:[EBP-2C]	
0040409C	6A 00	PUSH 0	
0040409E	52	PUSH EDX	
0040409F	FF15 2C104000	CALL DWORD PTR DS:[&msubvm60.rtcMsgBox	msubvm60.rtcMsgBox
004040A5	8D45 A4	LEA ECX, DWORD PTR SS:[EBP-5C]	
004040A8	8D4D B4	LEA ECX, DWORD PTR SS:[EBP-4C]	
004040AB	50	PUSH EAX	
004040AC	8D55 C4	LEA EDX, DWORD PTR SS:[EBP-3C]	
004040AF	51	PUSH ECX	
004040B0	8D45 D4	LEA ECX, DWORD PTR SS:[EBP-2C]	
004040B3	52	PUSH EDX	
004040B4	50	PUSH EAX	
004040B5	6A 04	PUSH 4	
004040B7	FF15 0C104000	CALL DWORD PTR DS:[&msubvm60.__vbaFree	msubvm60.__vbaFreeVarList
004040BD	83C4 14	ADD ESP, 14	
004040C0	FF15 10104000	CALL DWORD PTR DS:[&msubvm60.__vbaEnd	msubvm60.__vbaEnd
004040C6	C745 FC 0000	MOV DWORD PTR SS:[EBP-4], 0	
004040C0	68 03414000	PUSH Copia_de.00404103	

Vemos más abajo, en la address "00404081" la referencia "Cracker??", y todavía un poco más abajo en la address "0040409F" un "rtcMsgBox"

Y si miramos más arriba vemos en la address "00404056" el salto condicional **JNZ** (Salta si no es 0) donde se decide en este caso si el valor del registro EAX no es 0, saltará a la address "004040C6" para continuar hasta mostrarnos la ventana:

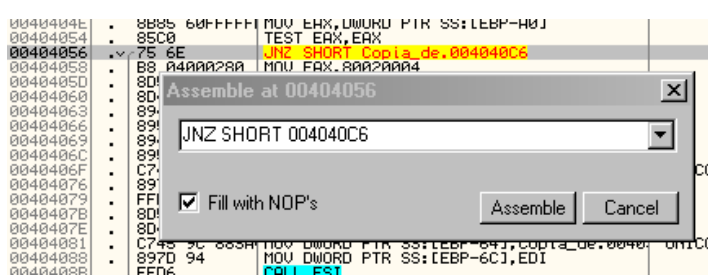


Si por el contrario el valor de EAX es 0 , continua hasta mostrarnos la ventana:

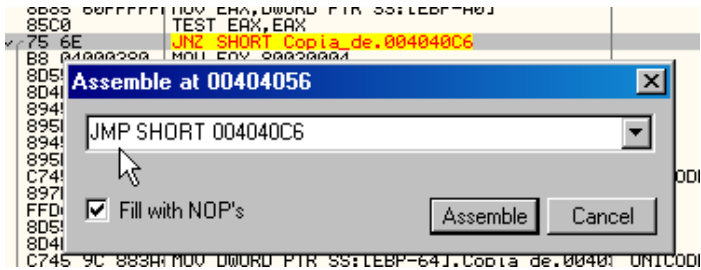


Pues voy a substituir el salto condicional **JNZ** por un salto incondicional **JMP** para que siempre salte a la address "004040C6"

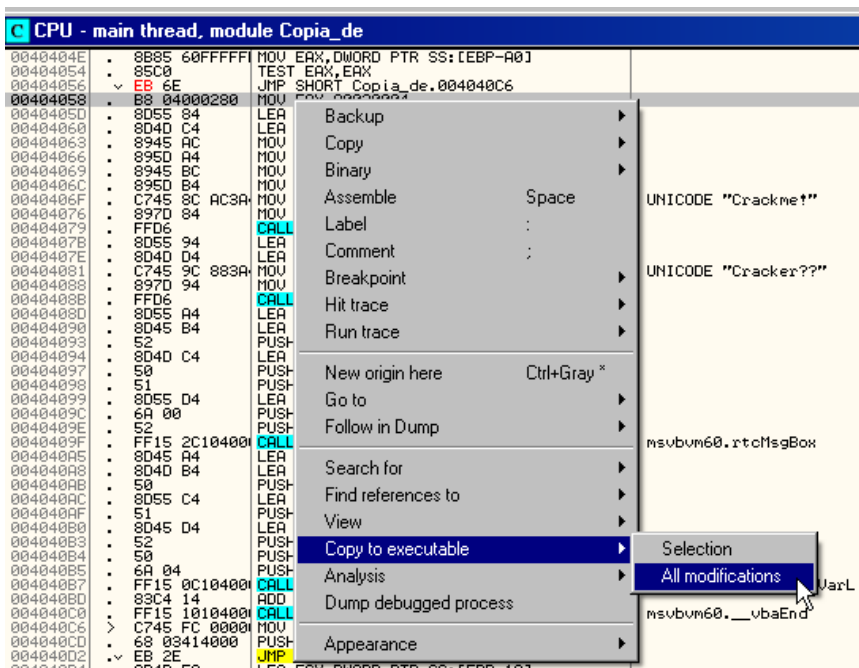
Osea esto



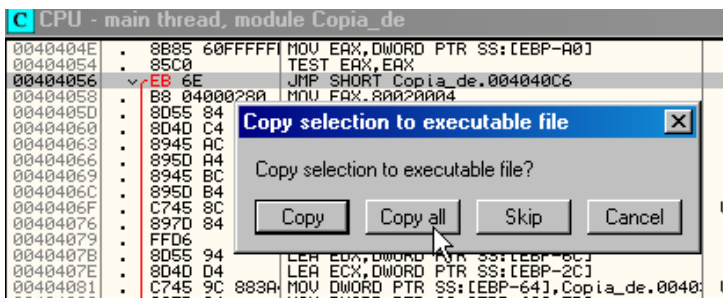
Por esto



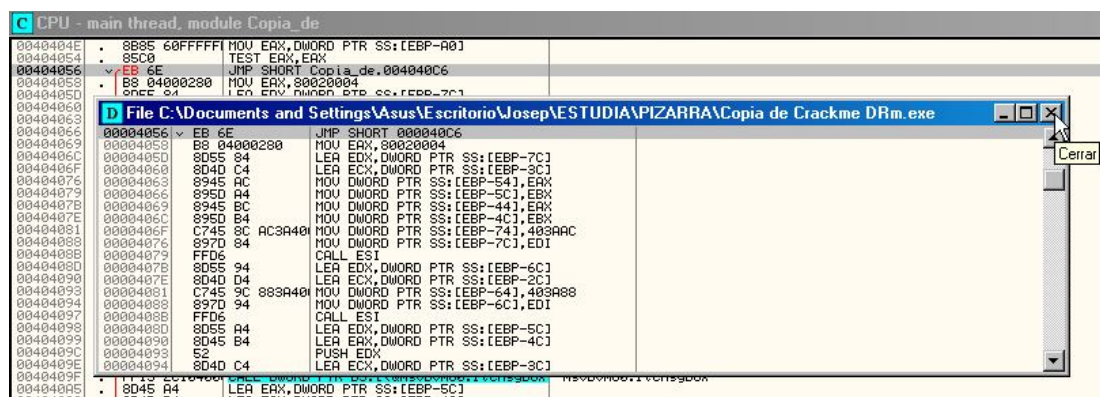
Le doy a "Assemble", guardo cambios



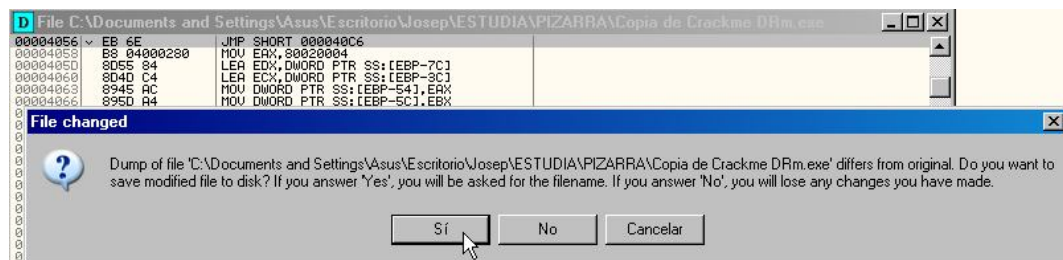
Le doy a "Copy all"



Después cierro la ventana donde me muestra los cambios



Olly me avisa de que los cambios difieren del .exe original, y que si quiero guardarlos, y le decimos que "Sí"



Por último lo guardamos con otro nombre, yo le pongo **"Copia de Crackme DRm_1.exe"**



Salimos totalmente de Olly (Si queremos antes de salir ya podemos desactivar nuestro Plugin IsDebuggerPresent por que ya ha hecho su trabajo), comprobamos que el nuevo "Copia de Crackme DRm_1.exe" se ejecuta correctamente fuera de Olly sin la Nag del principio, y vemos que sí.



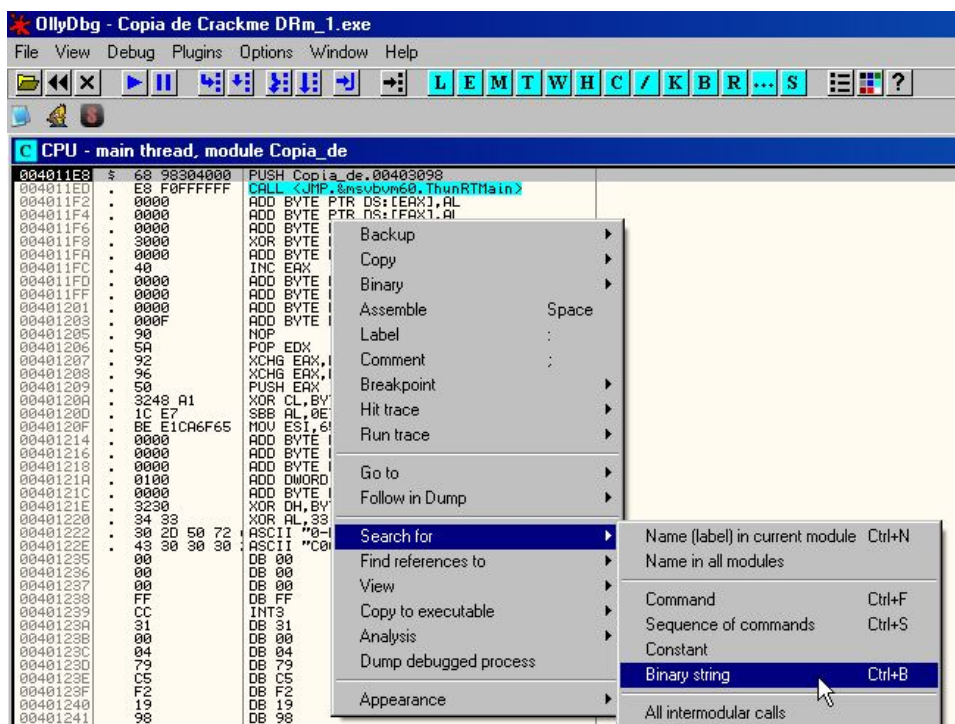
Ahora lo cargamos con Olly, lo hacemos correr y también nos sale lo mismo, o sea que con el salto incondicional hemos conseguido sortear el IsDebuggerPresent (que todavía está vivo dentro de las entrañas del crackme), ya que al pasar por el **JMP** siempre irá al mismo lugar de destino, y con ello solucionado nuestra primera misión de que la Nag no salte y de paso a la segunda ventana cuando corremos el crackme, tanto dentro como fuera de Olly.

continuamos....

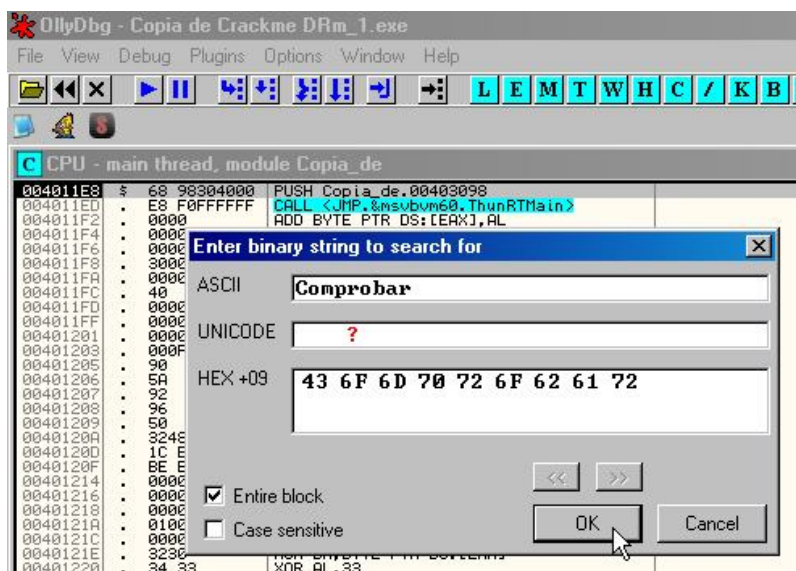
SEGUNDA MISIÓN: debemos activar el button "Comprobar"



Cargamos de nuevo nuestro crackme modificado al que hemos llamado "Copia de Crackme DRm_1.exe" con Olly

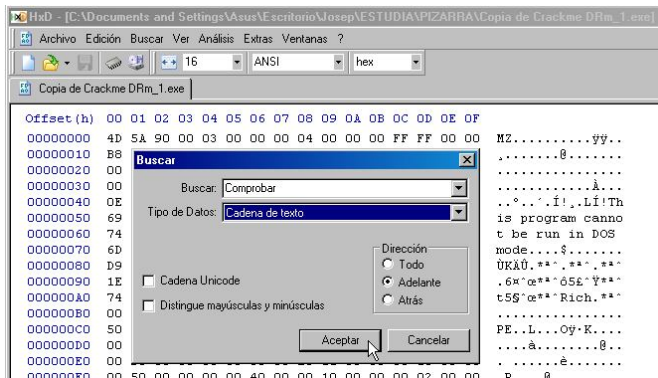


Vamos ahora a *Activar el button "Comprobar"*, y sin ejecutar nada, en la ventana principal de Olly damos clic derecho "Search for - Binary string" y en la caja de texto "ASCII" tipeamos "Comprobar" (que es el Caption del Button que buscamos), damos "OK"

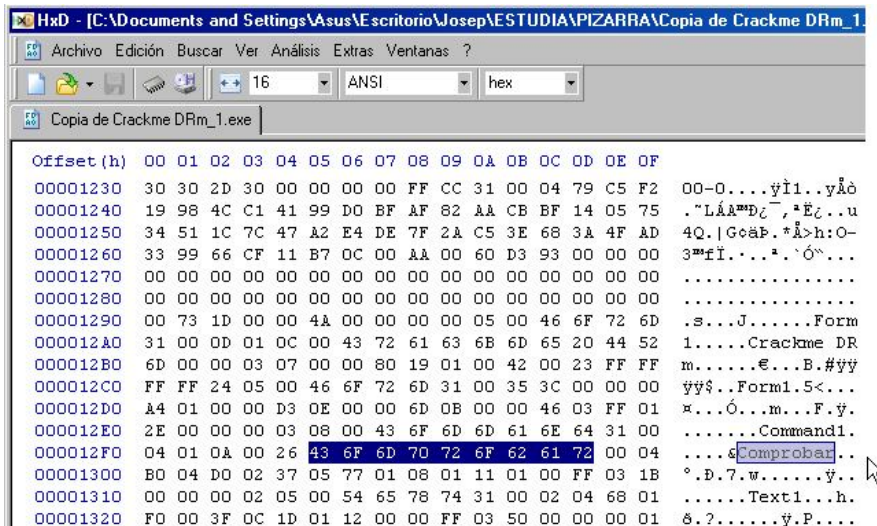


Y aparecemos en la address "004012F4"

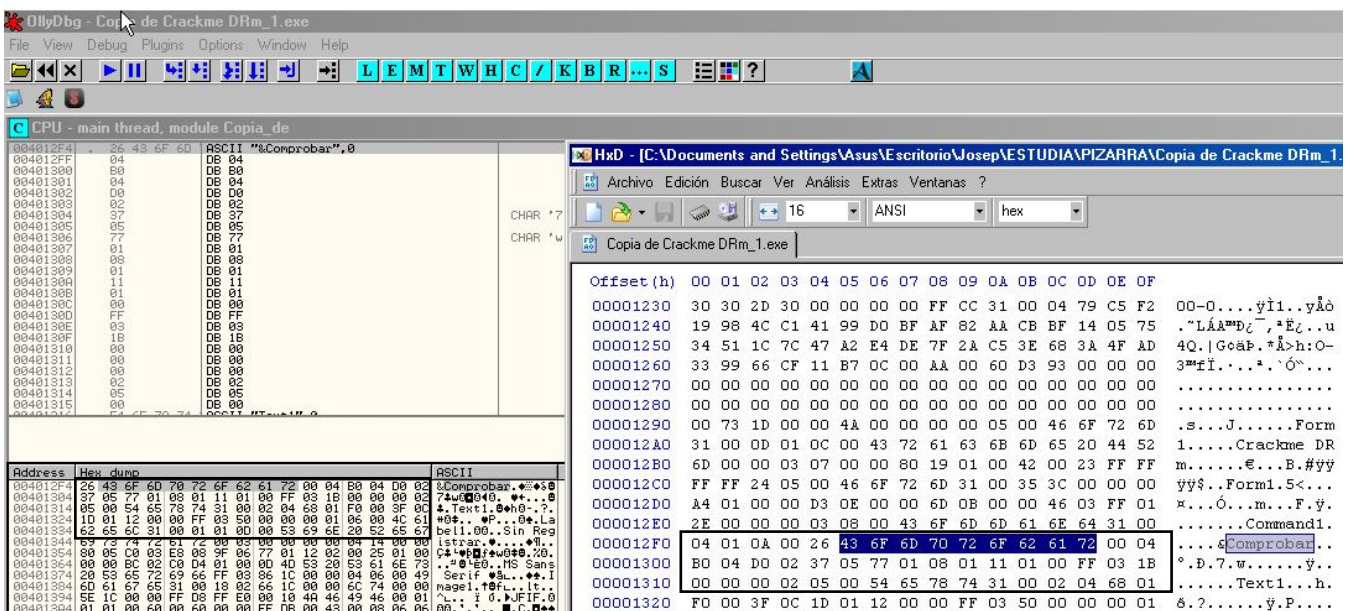
Salimos de Olly y vamos a abrir nuestro crackme modificado "Copia de Crackme DRm_1.exe" con la tool "HxD Editor Hexadecimal" y tipeamos dentro de la caja de texto "Buscar" la palabra "Comprobar", y desplegamos en la segunda caja de texto "Tipo de datos" para que busque como "Cadena de datos"



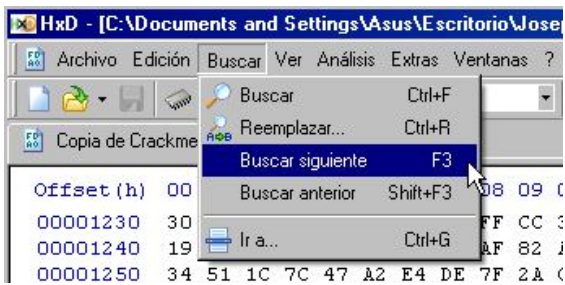
Le damos a "Aceptar" y aparecemos en



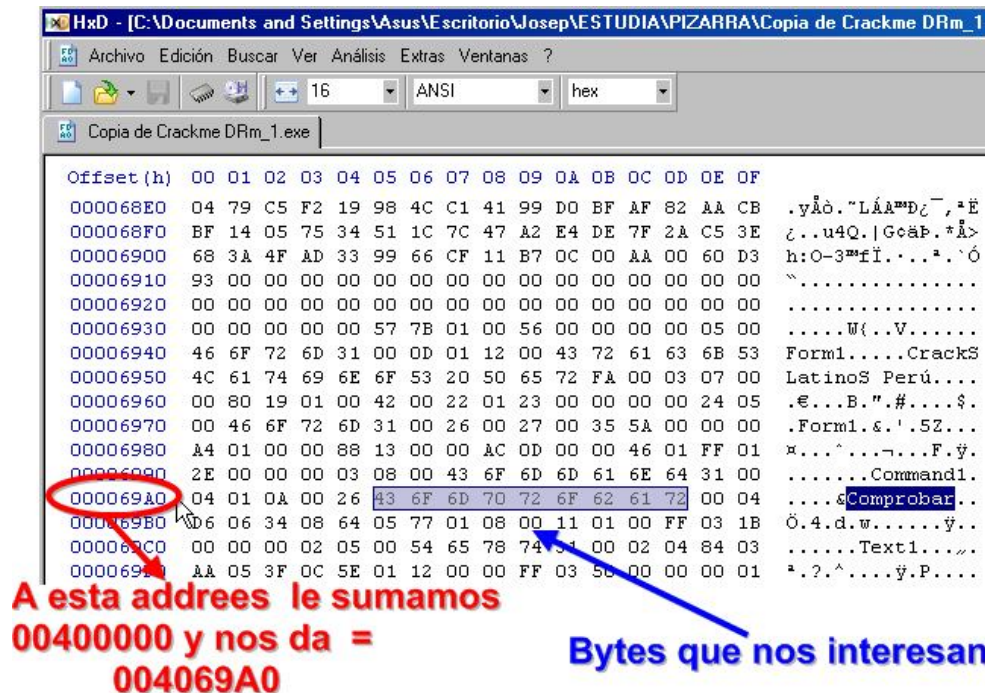
Que si recordamos (He sobrepuesto las dos imágenes para que se vea mejor) son exactamente los mismos valores hexadecimales que obtuvimos con Olly en la ventana "Dump"



Continuamos con la tool **HxE**, y vamos a ver si encuentra alguna otra cadena de texto con la misma referencia "Comprobar" dándole a la tecla "F3"



Y efectivamente encuentra otra. (que Olly no encontró). Ahora estamos parados aquí:



Creo que ahora si que estamos en el sitio adecuado para cambiar los recursos que conforman el button "Comprobar" y que Olly no nos mostraba. (Si continuamos buscando más referencias con la tecla "F3", el HxD nos dice que no hay más.

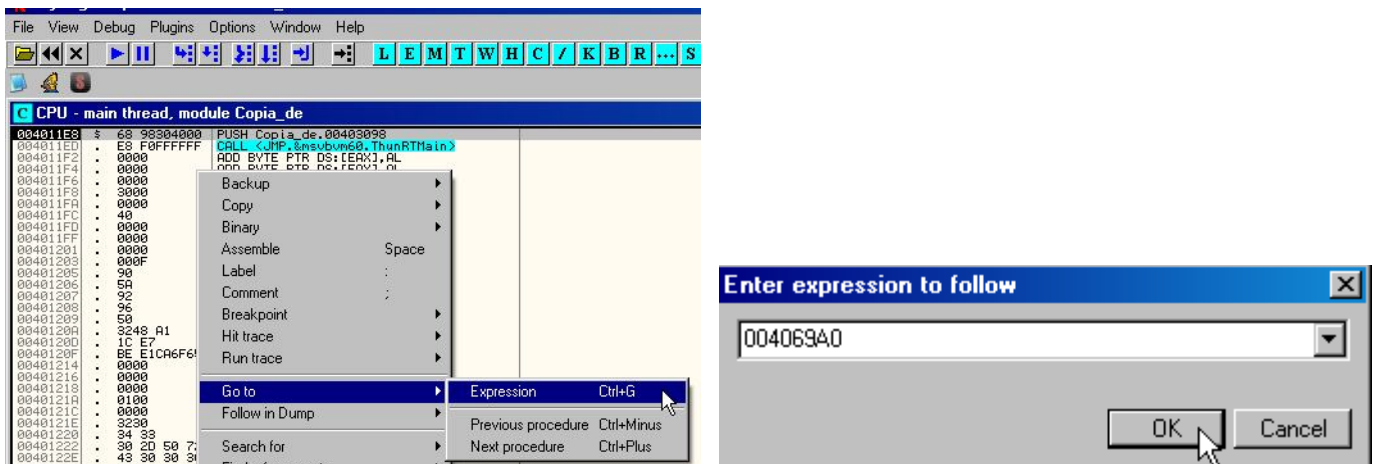


En este caso nuestra intuición de cracker nos dice que los Bytes que nos interesa cambiar están en la posición 12 después de la cadena de texto "Comprobar" de esta segunda parada, recordemos que **00=False=Desactivado** y que **01=True=Activado**, y yo había dicho que normalmente estaban en la posición 11 (pido disculpas, esta vez me he equivocado de una je.je.je.., pero normalmente están por estas posiciones).

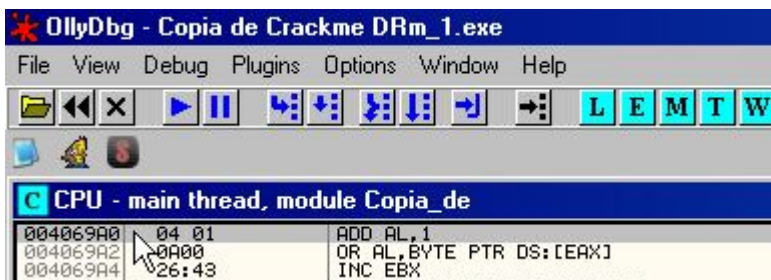
Bien, ahora podríamos cambiar directamente el valor hexadecimal 00 por 01 con este magnífico editor hexadecimal y guardar los cambios, pero vamos a intentarlo hacerlo con nuestro Olly.

Apuntamos la address **"000069A0"** que es la segunda parada obtenida con el HxD al buscar la cadena de texto "Comprobar", le sumamos el "ImageBase" del crackme que en este caso es **"00400000"** dándonos como resultado = **"004069A0"**, lo anotamos y salimos del HxD (recordemos que no hemos hecho ningún cambio con esta tool).

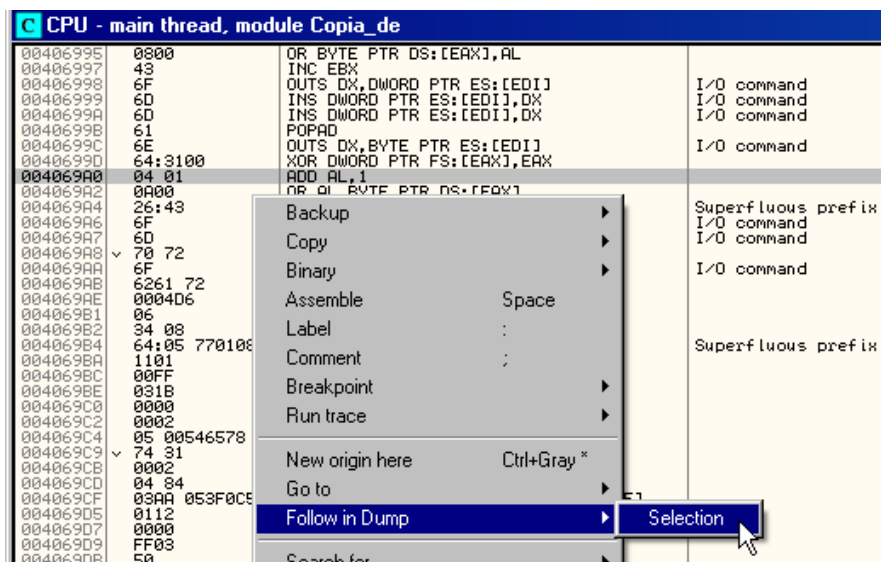
Cargamos nuestro **"Copia de Crackme DRm_1.exe"** de nuevo con Olly y sin ejecutar nada, parados en el punto de entrada, damos click derecho "Goto to", "Expresión " e introducimos el resultado del cálculo de la address que realizamos anteriormente **"004069A0"** para que Olly la busque



Damos "OK", y aparecemos en



Nos posicionamos encima con el cursor, clic derecho y "Follow in Dump" - "Seleccction" para que nos muestre en la ventana "Dump" lo que lleva dentro



Ahora nos fijamos en la ventana "Dump", Olly nos muestra lo que estabamos buscando, que es exactamenete lo que nos mostró el Editor Hexadecimal HxE en su

segunda parada de búsqueda, nos posicionamos marcando el texto entero "Comprobar" con el ratón, y en la posición número 12 tenemos los bytes que debemos cambiar.

Address	Hex dump	ASCII
004069A0	04 01 0A 00 26 43 6F 6D 70 72 6F 62 61 72 00 04	0...&Comprobar.0
004069B0	06 06 34 08 64 05 77 01 08 00 11 01 00 FF 03 1B	i4d#w0.0.0
004069C0	00 00 00 02 05 00 54 65 78 74 31 00 02 04 84 03	...0..Text1.000
004069D0	AA 05 3F 0C 5E 01 12 00 00 FF 03 50 00 00 00 01	~?..00..P...0
004069E0	06 00 4C 61 62 65 6C 31 00 01 00 00 53 69 0E	..Label1.00..Sin
004069F0	20 52 65 67 69 73 74 72 61 72 00 03 00 00 00	Registrar.0....
00406A00	04 14 00 00 80 05 54 06 A4 0B 82 A7 01 12 02	0..0T#00.w000
00406A10	00 25 01 00 00 00 BC 02 C0 04 01 00 00 4D 53 20	.%0...0000..MS

Nos posicionamos sobre este Byte "00", clic derecho de ratón y "Binary" - "Edit"

Address	Hex dump	ASCII
004069A0	04 01 0A 00 26 43 6F 6D 70 72 6F 62 61 72 00 04	0...&Comprobar.0
004069B0	06 06 34 08 64 05 77 01 08 00 11 01 00 FF 03 1B	i4d#w0.0.0
004069C0	00 00 00 02 05 00 54 65 78 74 31 00 02 04 84 03	...0..Text1.000
004069D0	AA 05 3F 0C 5E 01 12 00 00 FF 03 50 00 00 00 01	~?..00..P...0
004069E0	06 00 4C 61 62 65 6C 31 00 01 00 00 53 69 0E	..Label1.00..Sin
004069F0	20 52 65 67 69 73 74 72 61 72 00 03 00 00 00	Registrar.0....
00406A00	04 14 00 00 80 05 54 06 A4 0B 82 A7 01 12 02	0..0T#00.w000
00406A10	00 25 01 00 00 00 BC 02 C0 04 01 00 00 4D 53 20	.%0...0000..MS
00406A20	53 61 6E 73 20 53 65 72 69 66 00 00 00 00 00	316E732053657269660000000000000000
00406A30	04 06 00 49 6D 61 67 65 31 00 00 00 00 00 00	040600496D616765310000000000000000
00406A40	6C 74 00 00 36 7A 01 00 42 40 00 00 00 00 00	6C740000367A0100424000000000000000
00406A50	00 00 36 04 00 00 28 00 00 00 00 00 00 00 00	0000360400002800000000000000000000
00406A60	00 00 01 00 08 00 00 00 00 00 00 00 00 00 00	0000010008000000000000000000000000
00406A70	00 00 00 00 00 00 00 00 01 00 00 00 00 00 00	0000000000000000000000000000000000
00406A80	04 00 4C 84 38 00 0C 02 93 00 00 00 00 00 00	04004C8438000C02930000000000000000
00406A90	9C 00 7C 9C 00 04 05 45 00 00 00 00 00 00 00	9C007C9C00040545000000000000000000

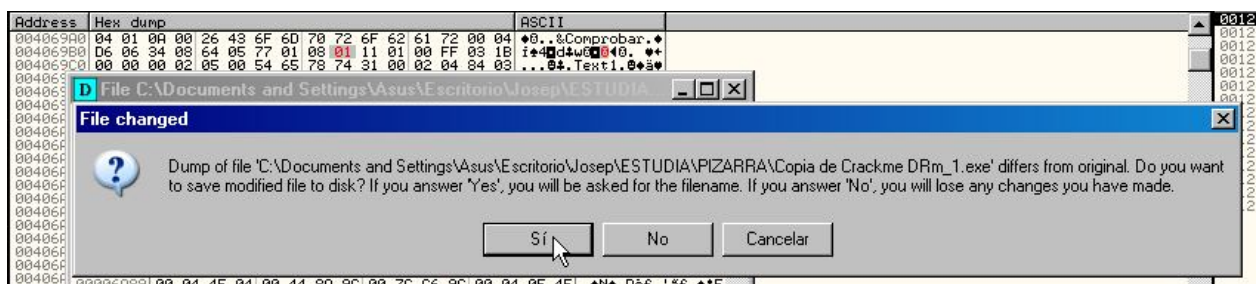
Y cambiamos su valor "00" por "01"

Address	Hex dump	ASCII
004069A0	04 01 0A 00 26 43 6F 6D 70 72 6F 62 61 72 00 04	0...&Comprobar.0
004069B0	06 06 34 08 64 05 77 01 08 00 11 01 00 FF 03 1B	i4d#w0.0.0
004069C0	00 00 00 02 05 00 54 65 78 74 31 00 02 04 84 03	...0..Text1.000
004069D0	AA 05 3F 0C 5E 01 12 00 00 FF 03 50 00 00 00 01	~?..00..P...0
004069E0	06 00 4C 61 62 65 6C 31 00 01 00 00 53 69 0E	..Label1.00..Sin
004069F0	20 52 65 67 69 73 74 72 61 72 00 03 00 00 00	Registrar.0....
00406A00	04 14 00 00 80 05 54 06 A4 0B 82 A7 01 12 02	0..0T#00.w000
00406A10	00 25 01 00 00 00 BC 02 C0 04 01 00 00 4D 53 20	.%0...0000..MS
00406A20	53 61 6E 73 20 53 65 72 69 66 00 00 00 00 00	316E732053657269660000000000000000
00406A30	04 06 00 49 6D 61 67 65 31 00 00 00 00 00 00	040600496D616765310000000000000000
00406A40	6C 74 00 00 36 7A 01 00 42 40 00 00 00 00 00	6C740000367A0100424000000000000000
00406A50	00 00 36 04 00 00 28 00 00 00 00 00 00 00 00	0000360400002800000000000000000000
00406A60	00 00 01 00 08 00 00 00 00 00 00 00 00 00 00	0000010008000000000000000000000000
00406A70	00 00 00 00 00 00 00 00 01 00 00 00 00 00 00	0000000000000000000000000000000000
00406A80	04 00 4C 84 38 00 0C 02 93 00 00 00 00 00 00	04004C8438000C02930000000000000000
00406A90	9C 00 7C 9C 00 04 05 45 00 00 00 00 00 00 00	9C007C9C00040545000000000000000000
00406AC0	AE 00 54 66 54 00 00 00 00 00 00 00 00 00 00	AE00546654000000000000000000000000
00406AD0	86 00 4F 0A 00 00 00 00 00 00 00 00 00 00 00	86004F0A00000000000000000000000000
00406AE0	06 00 1B 04 00 00 00 00 00 00 00 00 00 00 00	06001B0400000000000000000000000000
00406AF0	45 00 6D 71 00 1E 07 F3 00 00 00 00 00 00 00	45006D71001E07F3000000000000000000
00406B00	71 00 1E 07 F3 00 00 00 00 00 00 00 00 00 00	71001E07F3000000000000000000000000
00406B10	2A 00 3C 52 68 00 00 00 00 00 00 00 00 00 00	2A003C5268000000000000000000000000
00406B20	86 00 34 38 6A 0A 00 00 00 00 00 00 00 00 00	860034386A0A0000000000000000000000
00406B30	29 00 02 22 00 00 00 00 00 00 00 00 00 00 00	2900022200000000000000000000000000

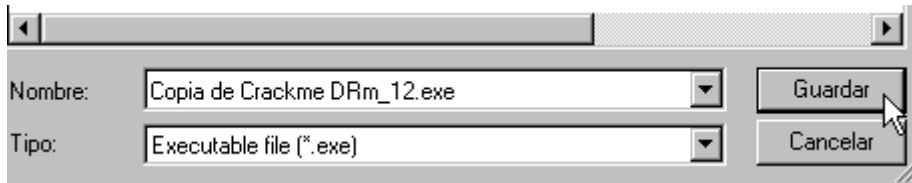
Le damos a "OK". Ahora solo queda guardar los cambios, que deberemos hacerlo posicionados dentro de la misma ventana "Dump", click derecho de ratón "Copy to executable file"

Address	Hex dump	ASCII
004069A0	04 01 0A 00 26 43 6F 6D 70 72 6F 62 61 72 00 04	0...&Comprobar.0
004069B0	06 06 34 08 64 05 77 01 08 00 11 01 00 FF 03 1B	i4d#w0.0.0
004069C0	00 00 00 02 05 00 54 65 78 74 31 00 02 04 84 03	...0..Text1.000
004069D0	AA 05 3F 0C 5E 01 12 00 00 FF 03 50 00 00 00 01	~?..00..P...0
004069E0	06 00 4C 61 62 65 6C 31 00 01 00 00 53 69 0E	..Label1.00..Sin
004069F0	20 52 65 67 69 73 74 72 61 72 00 03 00 00 00	Registrar.0....
00406A00	04 14 00 00 80 05 54 06 A4 0B 82 A7 01 12 02	0..0T#00.w000
00406A10	00 25 01 00 00 00 BC 02 C0 04 01 00 00 4D 53 20	.%0...0000..MS
00406A20	53 61 6E 73 20 53 65 72 69 66 00 00 00 00 00	316E732053657269660000000000000000
00406A30	04 06 00 49 6D 61 67 65 31 00 00 00 00 00 00	040600496D616765310000000000000000
00406A40	6C 74 00 00 36 7A 01 00 42 40 00 00 00 00 00	6C740000367A0100424000000000000000
00406A50	00 00 36 04 00 00 28 00 00 00 00 00 00 00 00	0000360400002800000000000000000000
00406A60	00 00 01 00 08 00 00 00 00 00 00 00 00 00 00	0000010008000000000000000000000000
00406A70	00 00 00 00 00 00 00 00 01 00 00 00 00 00 00	0000000000000000000000000000000000
00406A80	04 00 4C 84 38 00 0C 02 93 00 00 00 00 00 00	04004C8438000C02930000000000000000
00406A90	9C 00 7C 9C 00 04 05 45 00 00 00 00 00 00 00	9C007C9C00040545000000000000000000
00406AA0	C7 00 05 2D 45 50 00 00 00 00 00 00 00 00 00	C700052D45500000000000000000000000
00406AB0	9C 00 2D 45 50 00 00 00 00 00 00 00 00 00 00	9C002D4550000000000000000000000000
00406AC0	AE 00 54 66 54 00 00 00 00 00 00 00 00 00 00	AE00546654000000000000000000000000
00406AD0	86 00 4F 0A 00 00 00 00 00 00 00 00 00 00 00	86004F0A00000000000000000000000000
00406AE0	06 00 1B 04 00 00 00 00 00 00 00 00 00 00 00	06001B0400000000000000000000000000
00406AF0	45 00 6D 71 00 1E 07 F3 00 00 00 00 00 00 00	45006D71001E07F3000000000000000000
00406B00	71 00 1E 07 F3 00 00 00 00 00 00 00 00 00 00	71001E07F3000000000000000000000000
00406B10	2A 00 3C 52 68 00 00 00 00 00 00 00 00 00 00	2A003C5268000000000000000000000000
00406B20	86 00 34 38 6A 0A 00 00 00 00 00 00 00 00 00	860034386A0A0000000000000000000000
00406B30	29 00 02 22 00 00 00 00 00 00 00 00 00 00 00	2900022200000000000000000000000000

Oly nos mostrará el cambio y le decimos que "Sí"



Le ponemos otro nombre, yo lo he renombrado como **"Copia de Crackme DRm_12.exe"** y le damos a "Guardar"



Salimos totalmente de Olly, vamos a buscar nuestro nuevo Crackme por segunda vez modificado, lo ejecutamos y vemos que el button "Comprobar" está activado con lo cual también hemos conseguido nuestra segunda misión.

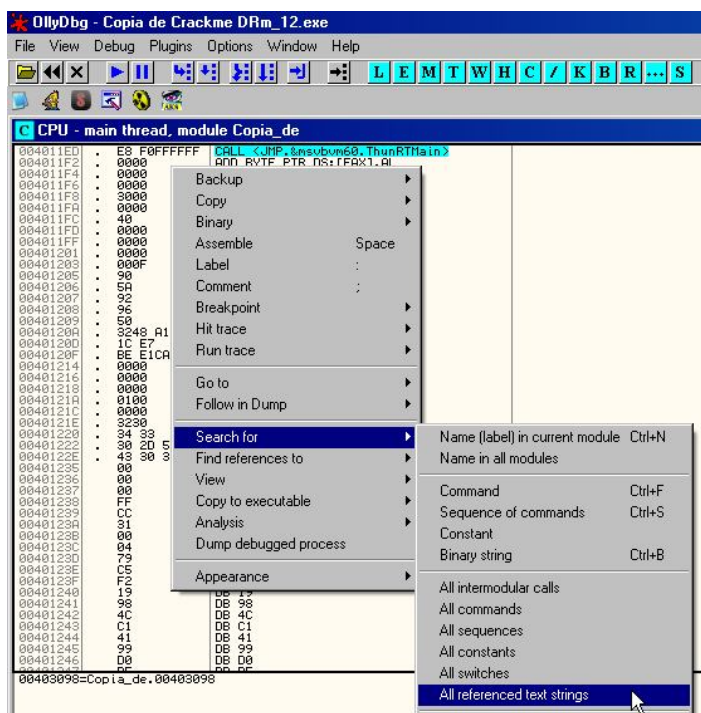
Y continuamos....

TERCERA MISIÓN: debemos registrarnos.

Abrimos nuestro Crackme ahora llamado **"Copia de Crackme DRm_12.exe"**, le ponemos un password cualquiera para registrarnos, le damos a "Comprobar" y no cuela. Pero tenemos un dato interesante, y es que nos muestra un espléndido mensaje que nos dice **"Perdido??"**. Si no recuerdo mal, este mensaje de texto también nos lo mostraba Olly al buscar Referencias de texto. Vamos a comprobarlo.



Aceptamos, salimos del Crackme, lo abrimos con Olly, buscamos todas las referencias



Hacemos "scroll" hacia abajo y efectivamente, aquí la tenemos. Se encuentra en la address **"00403E10"**

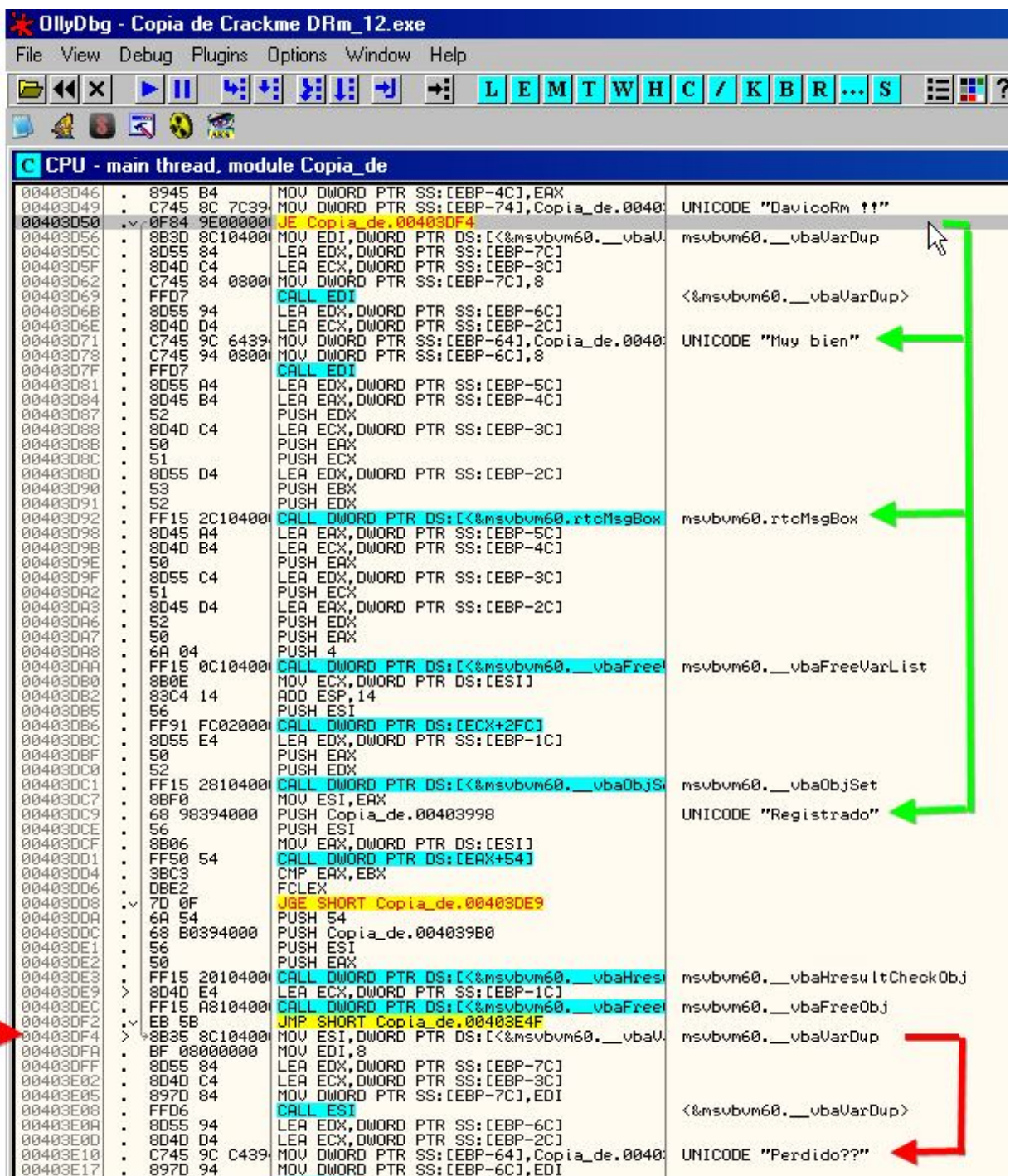
R Text strings referenced in Copia_de:.text		
Address	Disassembly	Text string
00403B24	ASCII " _vbaFreeStr",0	
00403B34	ASCII " _vbaFreeVarList"	
00403B44	ASCII 0	
00403B48	ASCII " _vbaVarDup",0	
00403B54	ASCII " _vbaVarTstEq",0	
00403B64	ASCII " _vbaFreeObj",0	
00403B74	ASCII " _vbaHresultChec"	
00403B84	ASCII "kObj",0	
00403B8C	ASCII " _vbaStrMove",0	
00403D49	MOV DWORD PTR SS:[EBP-74],Copia_de.00403D50	Unicode "DavicoRm !?"
00403D71	MOV DWORD PTR SS:[EBP-64],Copia_de.00403D72	Unicode "Muy bien"
00403DC9	PUSH Copia_de.00403998	Unicode "Registrado"
00403E10	MOV DWORD PTR SS:[EBP-64],Copia_de.00403E11	Unicode "Perdido??"
00403F46	PUSH Copia_de.004039E0	Unicode "Sin Registrar"
00403F98	MOV DWORD PTR SS:[EBP-74],Copia_de.00403F99	Unicode "DavicoRm"
00403FAA	MOV DWORD PTR SS:[EBP-64],Copia_de.00403FAB	Unicode "Vamos NewBie!! Tu Puedes Resolverme"
00404003	MOV DWORD PTR SS:[EBP-64],Copia_de.00404004	Unicode "PeruHa"
00404020	MOV DWORD PTR SS:[EBP-64],Copia_de.00404021	Unicode "ckCLS"
0040406F	MOV DWORD PTR SS:[EBP-74],Copia_de.00404070	Unicode "Crackme!?"
00404081	MOV DWORD PTR SS:[EBP-64],Copia_de.00404082	Unicode "Cracker??"
00404168	ASCII "FB",0	
0040416C	ASCII "?R".0	

Nos posicionamos sobre ella, damos dos clics de ratón y aparecemos en la ventana principal, ahora si subimos haciendo scroll observamos que estamos en la zona caliente, y vemos el entramado de cómo funciona todo, je.je.je.

En la address **"00403D50"** esta el saldo condicional "JE" que decide, que si el password introducido es el correcto, seguirá el código y nos mostrará el mensaje de chico bueno "Muy Bien" y "Registrado".

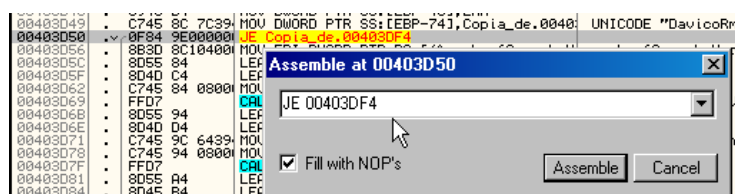
Si el password introducido no es el correcto, saltará a la address **"00403DF4"** y mostrará el mensaje de chico malo "Perdido??"

La ruta de Chico malo la he pintado en rojo,
Y la de Chico bueno en verde

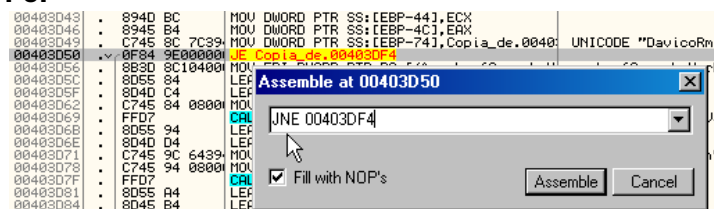


Pues a invertir el saldo condicional **JE** por un **JNE** consiguiendo que cualquier password que introduzcamos (menos el correcto), nos registrará y problema solucionado.

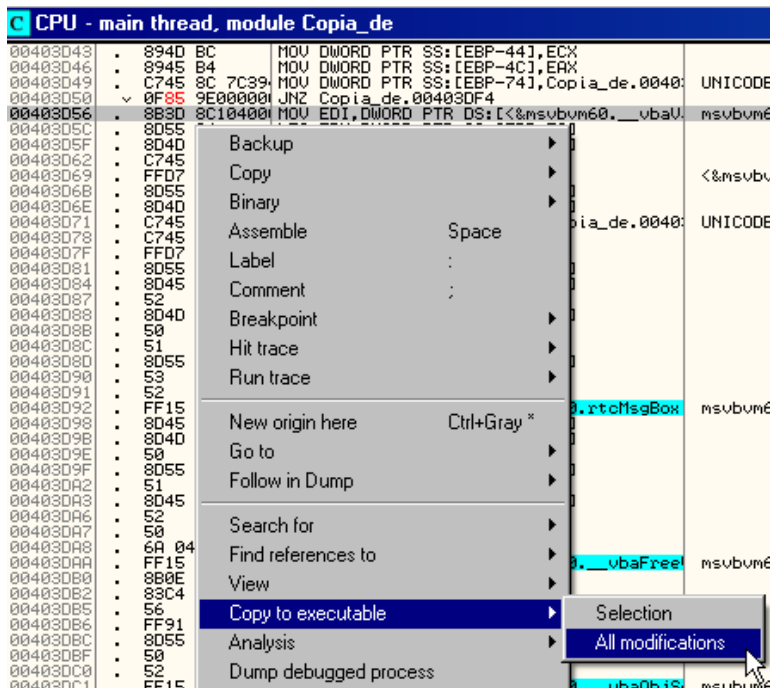
Nos posicionamos sobre el saldo condicional **JE**, damos dos clics izquierdo de ratón y cambiamos



Por



Le damos a "Assemble", Guardamos cambios "Clic izquierdo "Copy to executable - All modifications - Copy all" y lo guardamos con otro nombre "**Copia de Crackme DRm_123**"



Salimos de Olly, abrimos el nuevo y final crackme modificado por tercera vez al que hemos llamado "**Copia de Crackme DRm_123**", lo ejecutamos, rellenamos la caja de texto con un password cualquiera, le damos a "Comprobar" y por fin estamos Registrados, con lo cual hemos conseguido nuestra tercera misión.

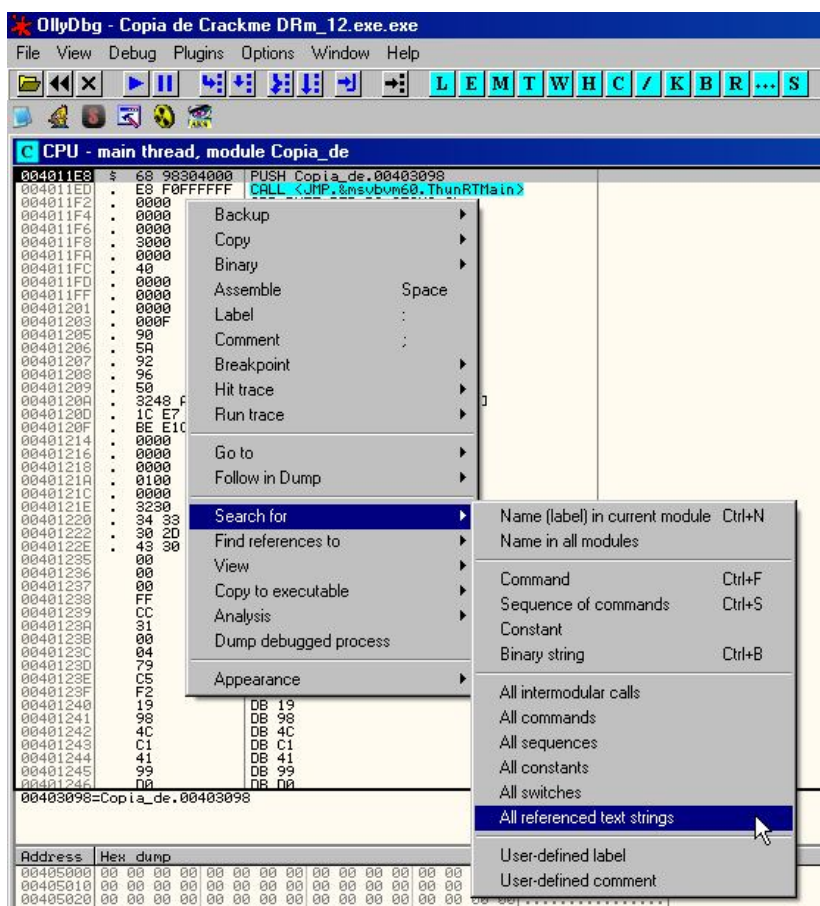


Pero eso, no es todo, ahora vamos a buscar el Password correcto. (y este se lo dedico a **Ismael Pérez**)

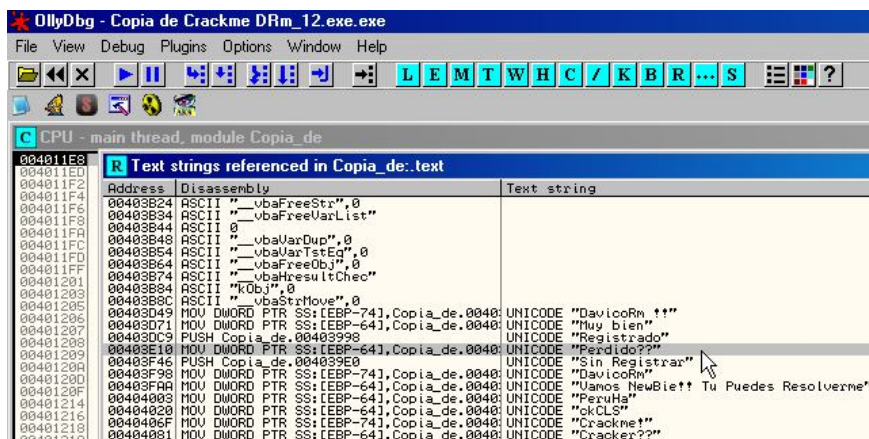
Para ello debemos retroceder y cargar en Olly el "**Copia de Crackme DRm_12.exe**" (Ojo, es el crackme que hemos modificado por segunda vez, al que únicamente hemos burlado la protección antidebbugger y activado el button "Comprobar"), que cuando ponemos un password cualquiera nos sale el mensaje de chico malo "**Perdido??**"



parados en el punto de entrada, buscamos todas las referencias

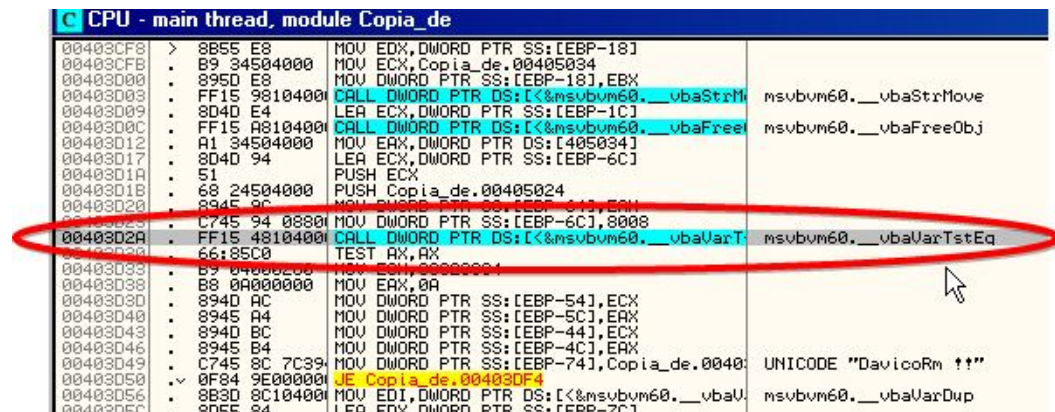


Hacemos "scroll" hacia abajo y aquí la tenemos. Se encuentra en la address "00403E10"

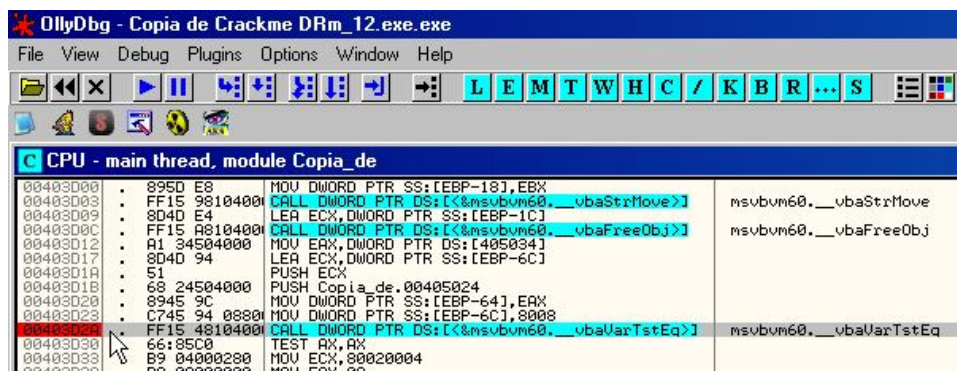


Nos posicionamos sobre ella, damos dos clics de ratón y aparecemos en la ventana principal, ahora si subimos haciendo scroll nos paramos en la address "00403D2A" que es la primera "CALL" que hay antes de llegar a la address "00403D50" donde se encuentra el saldo condicional "JE" que decide, si el password introducido es el correcto, seguirá el código y nos mostrará el mensaje de chico bueno "Muy Bien" y "Registrado" y si el password introducido no es el correcto, saltará a la address "00403DF4" y mostrará el mensaje de chico malo "Perdido??"

Además en la información de esta "CALL", Olly tambien nos muestra la APIs "__vbaVarTstEq" (que sirve para comparar cadenas), y esto es interesantísimo para nosotros, je,je,je



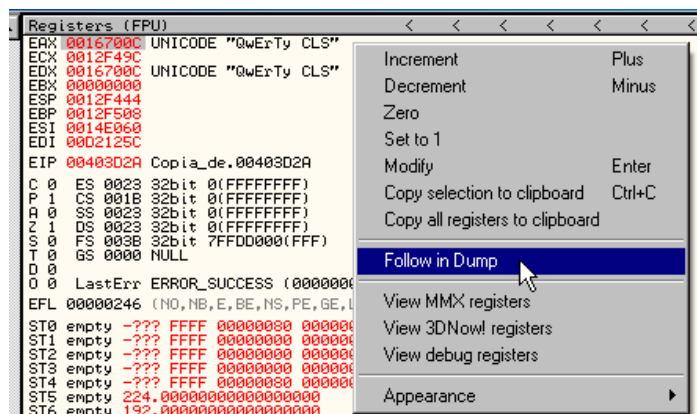
Bien, pues vamos a poner un "Breackpoint Toogle" a esta CALL para que Olly pare aquí cuando corramos el Crackme. Nos posicionamos sobre la "Call", le damos a la tecla "F2", y nos quedará así:



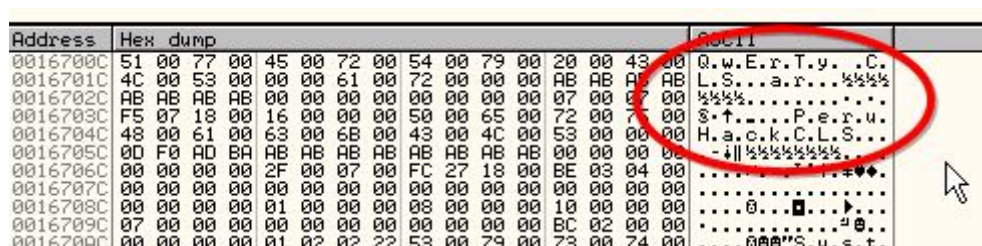
Ahora corremos el crackme, tipeamos en la caja de texto un serial cualquiera



Le damos a "Comprobar" y Olly para en el "Breackpoint", ahora nos vamos a la ventana "Registers" y vemos que en los registros "EAX" y "EDX" aparece nuestro password falso que hemos introducido, pues nos posionamos sobre cualquiera de estos dos registros, (yo lo hago sobre EAX", y con clic izquierdo del ratón le damos a "Follow in Dump" para que nos enseñe en el Dump lo que lleva dentro



Por último vamos a la ventana "Dump", y Olly nos muestra la comparación que buscamos Osea nuestro password falso introducido y el bueno.



Por lo que el password correcto es un magnífico hardcoded "PeruHackCLS"

Salimos de Olly, abrimos nuestro "Copia de Crackme DRm_12.exe", insertamos el password obtenido, le damos a Comprobar y.....



Estamos Registrados.....Yuuuupiiiiiii....

Me parece que me he enrollado demasiado y con una cuarta parte de la explicación y capturas de pantalla hubiera sido más que suficiente, pero es que hoy no tenía F1ACA y si muchas ganas de escribir.

A todo el que haya leído este tute quiero que sepa que si me he animado a escribirlo ha sido por la gran persona que hay dentro de "SoftDat Newzombie" por sus amenos e inteligentes videotutoriales que asiduamente comparte con todo el grupo de CLS y en la red compartiendo enseñanza y conocimientos, a "DavicoRm" creador del crackme, como no, al "MAESTRO RICNAR" culpable de que yo me metiera en esto je...je...je..., y a todo el gran grupo de "CLS".GRACIAS A TODOS.....AMIGOS

Otro para la colección.....

.....**MISIÓN CUMPLIDA**.....



Mis agradecimientos infinitos a

RICARDO NARVAJA, Ratón, Karpoff, _/_-=InDuLgEo=-_/_, Makkakko, Raziell, Guan de Dio, RDGMax, SoftDat Newzombie a todo el grupo de Cracks LatinoS y a todos los crackers del mundo

Que la serenidad nos haga fuertes y tenaces, y no se convierta en somnolencia, plácida aceptación de realidades insatisfactorias pero cómodas, que la paciencia no nos haga cobardes, que el progreso no nos haga autocomplacientes, y que lo logrado hasta ahora no nos haga olvidar que estamos a mitad de camino.

"QwErTy"

11 de septiembre de 2016
