

**[Tuto006 - PearlMountain Picture Collage Maker Pro v4.1.4  
(MVC++9.0)(KeyGen.FUERZA.BRUTA)]**



<b>Software</b>	<b>PearlMountain Picture Collage Maker Pro v4.1.4</b>
<b>DESCARGA</b>	<a href="https://www.pearlmountainsoft.com/picture-collage-maker/">https://www.pearlmountainsoft.com/picture-collage-maker/</a>
<b>Protección</b>	<b>Serial.</b>
<b>Herramientas</b>	Windows 7 SP1 x64 Bits (WMWare Workstation 12 Pro) OllyDBG OllyICE v1.10 (for.Win_v8.1_x64 nad_x32) RDG Packer Detector v0.7.6.2017 ZSoft Uninstaller v2.5 Visual Studio 2015  <a href="#">DESCARGAR HERRAMIENTAS</a>  <a href="#">DESCARGAR TUTO+ARCHIVOS</a>
<b>SOLUCIÓN</b>	Hallar SERIAL. KEYGEN (FUERZA BRUTA)
<b>AUTOR</b>	LUISFECAB
<b>RELEASE</b>	Agosto 06 2018 [TUTORIAL 006]

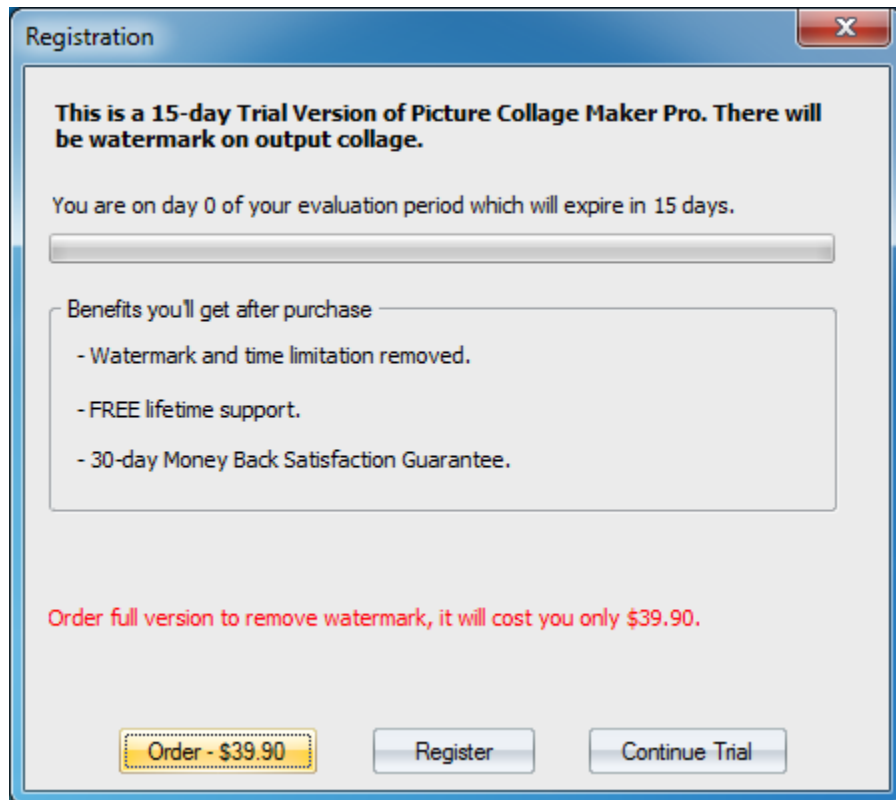
## INTRODUCCIÓN

Empiezo un nuevo tuto de una aplicación que descargué hace muchos años atrás y que registré usando el KeyGen que traía, que yo en ese entonces solo vivía fascinado, bueno lo sigo estando, de ver cómo era posible que hubiera gente que fuera capaz de hacer esas cosas como los KeyGen, permitiendo que puedas ejecutar estas aplicaciones sin restricciones. Ya pasaron mucho, y tuve la necesidad de utilizar este programa y utilicé esa misma versión de hace años, pero descubrí que ya tenían nueva versión y fui a utilizar los seriales de ese KeyGen y no sirvieron. Luego pensé en buscar esa versión en Internet que estuviera full, pero me dije, !PERA' UN MOMENTO LUIS, CÓMO ASÍ QUE BUSCARLA, NO SE SUPONE QUE ESTÁS APRENDIENDO REVERSING;, así que con lo aprendido de Reversing y mis pequeños logros me animé a descargar esta última versión y al revisarla que no estaba empacada, pues me metí al ruedo con este toro y para mi sorpresa que logré hacerlo. Hasta el momento, mientras escribo estas palabras ya hice un KeyGen que saca el serial a <FUERZA BRUTA> de la "Licencia Personal".

Lo anterior porque me gusta siempre contar un poco de historia. Terminando estas palabras, como siempre saludos para todos, para Ricardo al que debemos agradecerle por todo, también para solid que me aconsejo aprender algo ASM para hacer los KeyGen, que lo tengo como tarea pendiente; y también para QwErTy que siempre deja sus comentarios.

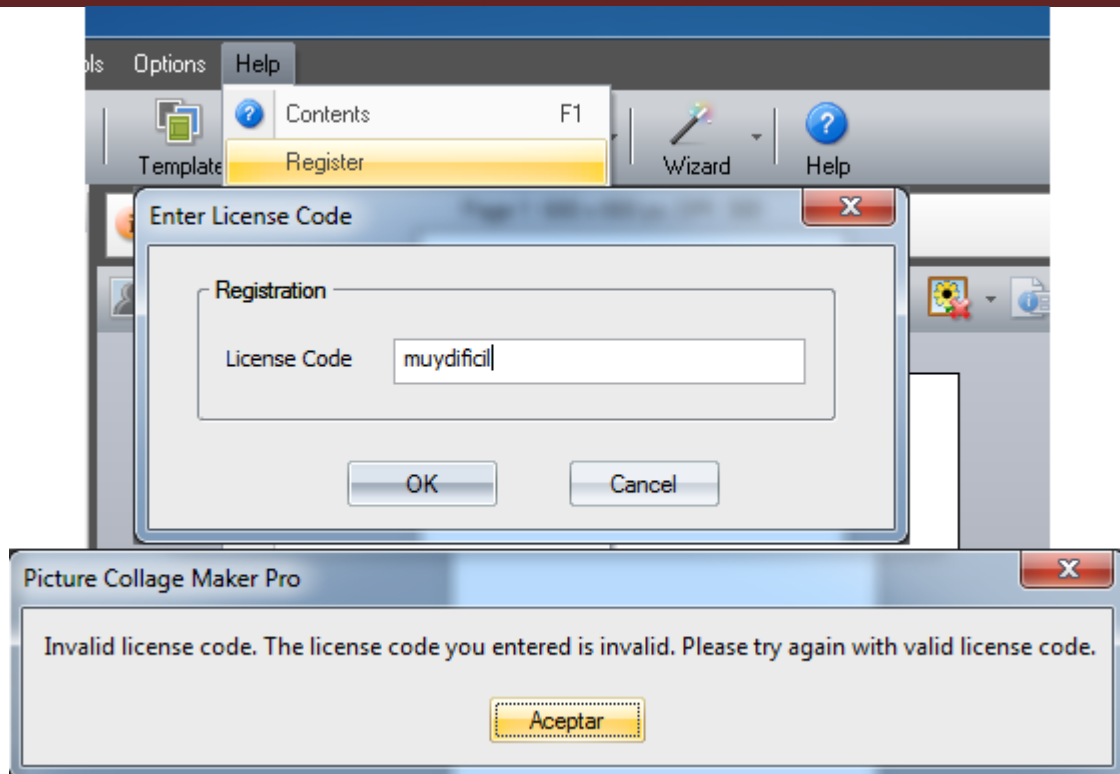
## ANALISIS INICAL

Una vez instalado, lo iniciamos para ver qué nos sale. Primero muestra unas ventanas de opciones para ver qué tipo de collage quieres trabajar, una especie de asistente, pasamos eso y ahí si tenemos nuestra <NAG> de versión de prueba.



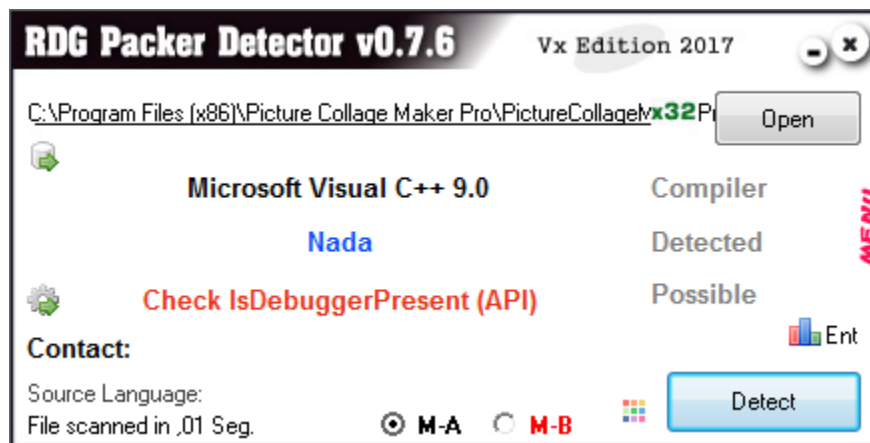
Muy bonita la pilluela, tenemos 15 días de prueba y te dice que si quiere la versión completa te costara US\$39.90 "dolorosos". Desde aquí podemos llegar a la solución, y efectivamente yo terminé entrando por este lado para terminar el trabajo, porque se me pasaron los 15 días de prueba; ya lo dicho varias veces que a mí no me rinde mucho el cracking, pero ahí voy, "lento como el elefante, pero aplastante". Bueno, en esos 15 días de prueba me dediqué a entrar por <Help->Register> que termina siendo lo mismo. Ya entrando al "Register", voy y le meto mi serial de guerra, "muydifícil".

## [Tuto006 - PearlMountain Picture Collage Maker Pro v4.1.4 (MVC++9.0)(KeyGen.FUERZA.BRUTA)]



Ahí hicimos lo de siempre y como debería ser lo esperado sale nuestro <CHICO MALO>.

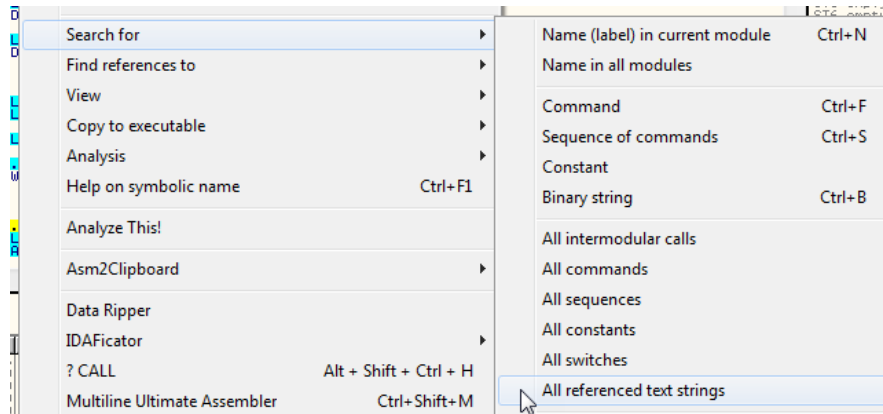
Antes de ir al ataque revisemos con el <RDG Packer Detector v0.7.6.2017> que más tiene este programa escondido bajo la manga.



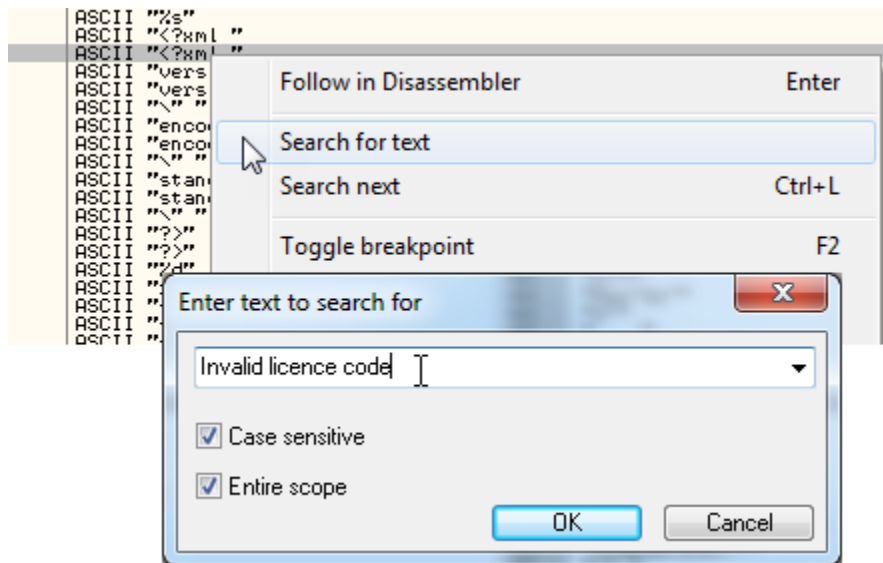
Con la vista <M-A> nos avisa que tiene la <API\_IsDebuggerPresent>. Para no colocar más capturas, la vista <M-B> arroja lo mismo. Con el OllyICE que utilizo aquí no sufrimos por eso.

## AL ATAQUE

Bueno, ahora a repetir esto, pero ya debugueando el programa con el OllyICE, llegando al <ENTRY POINT> (EP). Lo primero que hice fue buscar en las Strings.

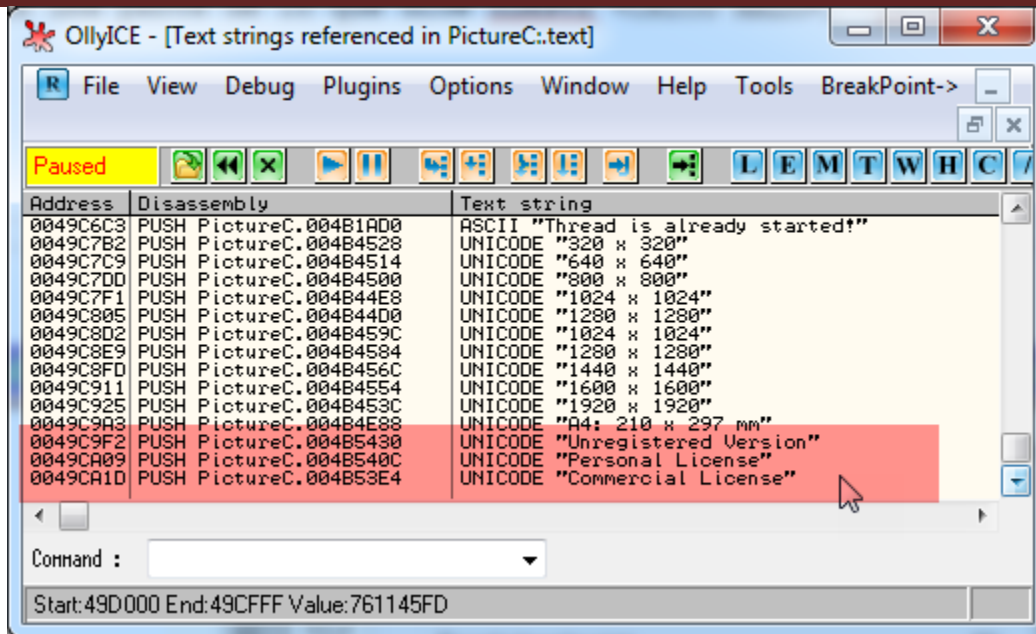


Y busqué una parte de lo que dice nuestro <CHICO MALO>, "Invalid license code". Si ves la imagen de abajo puedes notar que escribí mal "licence".

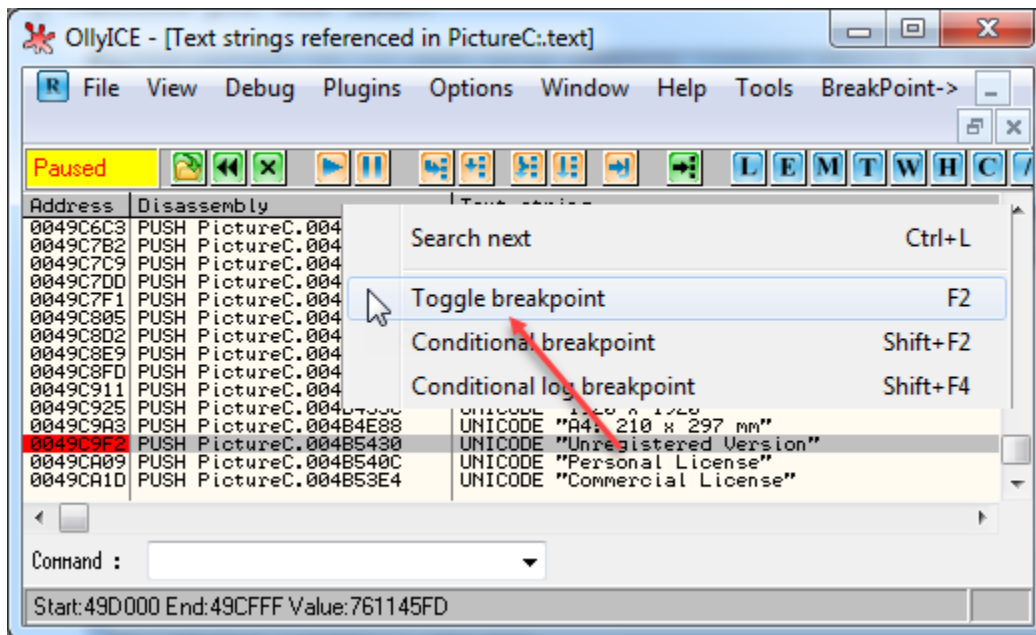


No encontré nada, no por mi error ortográfico que en la búsqueda lo corregí. Pero mira cómo es la vida, me acabo de encontrar unas Strings escribiendo el tuto que antes no había visto, así que desviémonos un poco de lo trazado y vámonos por ese lado.

## [Tuto006 - PearlMountain Picture Collage Maker Pro v4.1.4 (MVC++9.0)(KeyGen.FUERZA.BRUTA)]



Entonces, ya voy a escribir con la experiencia vivida de mis otros par de tutos y que ya uno le va cogiendo lógica a la cosa, entonces ahí vemos que tenemos dos tipos de licencia como les comentaba en la **INTRODUCCIÓN** pero lo más relevante es que fijo, pero fijo pasaremos por **"Unregistered Version"**.




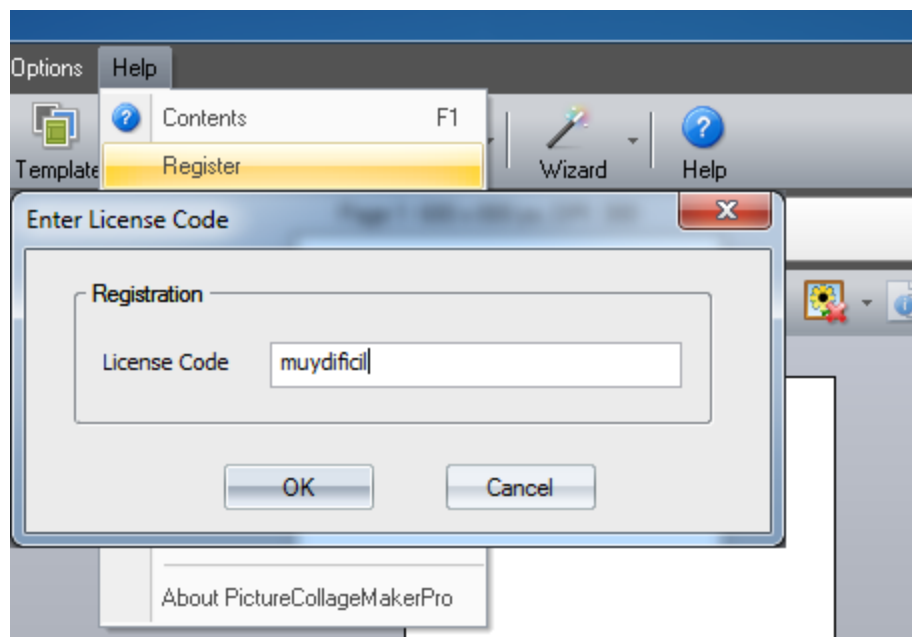
Le colocamos su **<BREAKPOINT>** (BP) y corremos el programa con **<F9>**.

# [Tuto006 - PearlMountain Picture Collage Maker Pro v4.1.4 (MVC++9.0)(KeyGen.FUERZA.BRUTA)]



Address	Hex dump	Disassembly	Comment
0049C9CF	CC	INT3	
0049C9D0	55	PUSH EBP	
0049C9D1	8BEC	MOV EBP,ESP	
0049C9D3	6A FF	PUSH -1	
0049C9D5	68 C6BE4900	PUSH PictureC.0049BEC6	
0049C9DA	64:A1 00000000	MOV EAX,DWORD PTR FS:[0]	
0049C9E0	50	PUSH EAX	
0049C9E1	A1 6C564E00	MOV EAX,DWORD PTR DS:[4E566C]	
0049C9E6	33C5	XOR EAX,EBP	
0049C9E8	50	PUSH EAX	
0049C9E9	8D45 F4	LEA EAX,[LOCAL.3]	
0049C9EC	64:A3 00000000	MOV DWORD PTR FS:[0],EAX	
0049C9F2	68 30544B00	PUSH PictureC.004B5430	UNICODE "Unregistered Version"
0049C9F7	B9 B45C4E00	MOV ECX,PictureC.004E5CB4	
0049C9FC	FF15 1CF14900	CALL DWORD PTR DS:[&mfc90u.#286]	mfc90u.#286
0049CA02	C745 FC 00000000	MOV [LOCAL.1],0	
0049CA09	68 0C544B00	PUSH PictureC.004B540C	UNICODE "Personal License"
0049CA0E	B9 B85C4E00	MOV ECX,PictureC.004E5CB8	
0049CA13	FF15 1CF14900	CALL DWORD PTR DS:[&mfc90u.#286]	mfc90u.#286
0049CA19	C645 FC 01	MOV BYTE PTR SS:[EBP-4],1	
0049CA1D	68 E4534B00	PUSH PictureC.004B53E4	UNICODE "Commercial License"
0049CA22	B9 BC5C4E00	MOV ECX,PictureC.004E5CB0	
0049CA27	FF15 1CF14900	CALL DWORD PTR DS:[&mfc90u.#286]	mfc90u.#286
0049CA2D	C745 FC FFFFFFFF	MOV [LOCAL.1],-1	
0049CA34	68 E0CC4900	PUSH PictureC.0049CCE0	
0049CA39	E8 7A13FFFF	CALL PictureC.0048D0B8	
0049CA3E	83C4 04	ADD ESP,4	
0049CA41	8B4D F4	MOV ECX,[LOCAL.3]	
0049CA44	64:890D 00000000	MOV DWORD PTR FS:[0],ECX	PictureC.0049CCD0 PictureC.0049CCD0
0049CA4B	59	POP ECX	
0049CA4C	8BE5	MOV ESP,EBP	
0049CA4E	5D	POP EBP	
0049CA4F	C3	RETN	

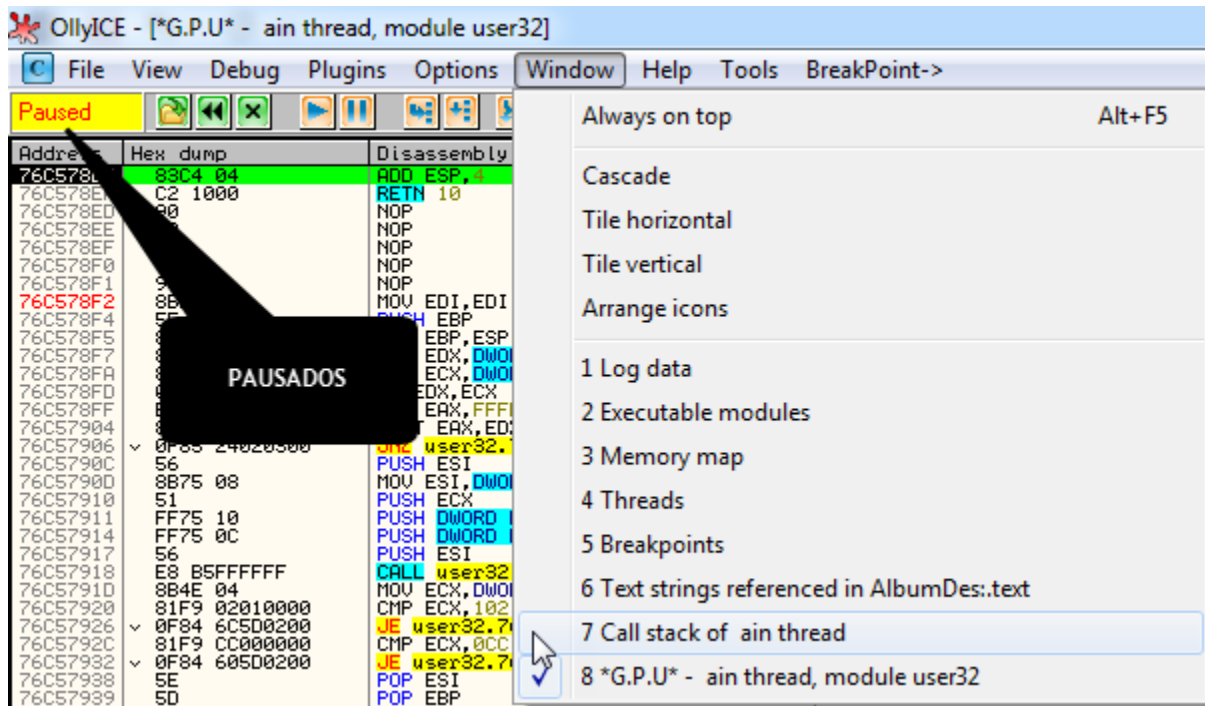
Repito que esta ruta no la había seguido, así que ni idea por dónde vamos. Bueno ahí estamos parados en **0049C9F2** y si observamos no hay ningún salto así que pasaremos por las licencias también, tracié un poco y no encontré nada importante, pero algo que aprendí en este tutorial es que los programas también utilizan otros archivos como las .DLL para comprobar los datos de registro, un poco de adelanto para recordarlo luego. Sigo con <F9> y no pasó nada más, así que termino en el inicio.

Ahora, a utilizar el famoso <CALL STACK> que no lo entiendo o manejo muy bien (pero con este tuto como que ya la estoy cogiendo), pero que lo vamos a utilizar para pillar el lugar que nos aproxime a lo que queremos buscar y es qué pasa con el serial. Para poder utilizarlo debemos primero pausar el programa cuando estemos en nuestro lugar de interés, con <F12> o .



## [Tuto006 - PearlMountain Picture Collage Maker Pro v4.1.4 (MVC++9.0)(KeyGen.FUERZA.BRUTA)]

La imagen anterior va a ser mi lugar de interés, así que pausamos el programa con <F12> o , y vamos a la ventana <CALL STACK>. Podemos utilizar los siguientes caminos: <ALT+K>,  o <Window->Call stack of ain thread>.



Con eso llegamos al <CALL STACK>. Yo voy a tratar de explicar lo que pienso es de utilidad para mí en estos casos. La forma en que entiendo esto es que a partir de un <CALL> se originan más llamados a nuevos procedimientos, nuevos <CALLS> que están unidos al <CALL> que inició todo, como un efecto domino. Entonces con esa lógica debo buscar e ir a ese <CALL> principal. Bueno, recordar que estamos parados cuando hemos ingresado nuestro serial, pero mucho ojo, no hemos dado "OK", si no tendríamos al <CHICO MALO>.



# [Tuto006 - PearlMountain Picture Collage Maker Pro v4.1.4 (MVC++9.0)(KeyGen.FUERZA.BRUTA)]

OlllyICE - [Call stack of ain thread]

File View Debug Plugins Options Window Help Tools BreakPoint->

Paused

Este es el de interés. Sale del módulo principal del programa.

Address	Stack	Procedure / arguments	Called from	Frame
000CF8D8	76C57910	user32.76C578D2	user32.76C57918	000CF8F0
000CF8F4	5E7EE39A	user32.GetMessageW	mfc90u.5E7EE394	000CF8F0
000CF8F8	00A1CA78	pMsg = 00A1CA78		
000CF8FC	00000000	hwnd = NULL		
000CF900	00000000	MsgFilterMin = 0		
000CF904	00000000	MsgFilterMax = 0		
000CF910	5E7B280E	mfc90u.#1218	mfc90u.5E7B2809	000CF938
000CF93C	5E7CB89B	mfc90u.#5022	mfc90u.5E7CB896	000CF938
000CF988	1007AF53	? <JMP.&mfc90u.#2208>	BCGCBPRO.1007AF4E	000CF984
000CF9B0	0044257A	BCGCBPRO.CBCGPDIALOG::DoModal	PictureC.00442574	000CF9AC
000CF9E8	5E7DD227	PictureC.00442520	mfc90u.5E7DD225	000CF9E4
000CFAF4	5E7DD409	mfc90u.5E7DD1EC	mfc90u.5E7DD404	000CFAF0
000CFB10	5E7B536E	Includes mfc90u.5E7DD409	mfc90u.5E7B5368	000CFB0C
000CFB50	100EA8BF	? <JMP.&mfc90u.#4685>	BCGCBPRO.100EA8BA	000CFB4C
000CFB6C	5E7AFF73	Includes BCGCBPRO.100EA8BF	mfc90u.5E7AFF70	000CFB68
000CFB8C	5E7B4837	mfc90u.#4702	mfc90u.5E7B4832	000CFB88
000CFBD8	100E9610	<JMP.&mfc90u.#4697>	BCGCBPRO.100E960B	000CFBD4
000CFBF0	5E7AF75B	Includes BCGCBPRO.100E9610	mfc90u.5E7AF755	000CFBEC
000CFC9C	5E7AF6CE	Includes mfc90u.5E7AF75B	mfc90u.5E7AF6C8	000CFC98
000CFCBC	5E7AE2F4	Includes mfc90u.5E7AF6CE	mfc90u.5E7AE2EE	000CFCB8
000CFD24	5E7AE580	? mfc90u.#1067	mfc90u.5E7AE57B	000CFD20
000CFD48	5E7AC247	? mfc90u.#1274	mfc90u.5E7AC242	000CFD44
000CFD90	76C562FA	Includes mfc90u.5E7AC247	user32.76C562F7	000CFD8C
000CFDBC	76C56D3A	? user32.76C562D7	user32.76C56D35	000CFD88
000CFE34	76C577D3	? user32.76C56C83	user32.76C577CE	000CFE30
000CFE98	76C5789A	? user32.76C576DF	user32.76C57895	000CFE94
000CFE98	5E7EE3C0	? user32.DispatchMessageW	mfc90u.5E7EE3BA	000CFE94
000CFEAC	00A1CA78	pMsg = WM_TIMER hw = 10410 (class="BCGPToolBar:400000:8:10003:10") ID		

Me voy extender con un poco de análisis, esperando que lo que explique no sea erróneo, todo este sale de mi interpretación. Ahí podemos ver que el procedimiento **PictureC.00442520** es el único que hace referencia al programa y si vamos a ese procedimiento.

000CF93C	5EA3B89B	mfc90u.#6022	
000CF988	1007AF53	? <JMP.&mfc90u.#2208>	
000CF9B0	0044257A	BCGCBPRO.CBCGPDIALOG::DoModal	
000CFAE8	5EA4D227	PictureC.00442520	
000CFAF4	5EA4D409	mfc90u.5EA4D1EC	
000CFB10	5EA2536E	Includes mfc90u.5EA4D409	
000CFB50	100EA8BF	? <JMP.&mfc90u.#4685>	
000CFB6C	5EA1FF73	Includes BCGCBPRO.100EA8BF	
000CFB8C	5EA24837	mfc90u.#4702	
000CFBD8	100E9610	<JMP.&mfc90u.#4697>	
000CFBF0	5EA1F75B	Includes BCGCBPRO.100E9610	
000CFC9C	5EA1F6CE	Includes mfc90u.5EA1F75B	
000CFCBC	5EA1E2F4	Includes mfc90u.5EA1F6CE	
000CFD24	5EA1E580	? mfc90u.#1067	
000CFD48	5EA1C247	? mfc90u.#1274	
000CFD90	76D962FA	Includes mfc90u.5EA1C247	
000CFDBC	76D96D3A	? user32.76D962D7	
000CFE34	76D977D3	? user32.76D96C83	
000CFE98	76D9789A	? user32.76D976DF	
000CFE98	5EA5E3C0	? user32.DispatchMessageW	
000CFEAC	00A9CA78	pMsg = WM_TIMER hw = 10410 (class="BCGPToolBar:400000:8:10003:10") ID	

Actualizar

Hide arguments Space

Thread ▶

Follow address in stack

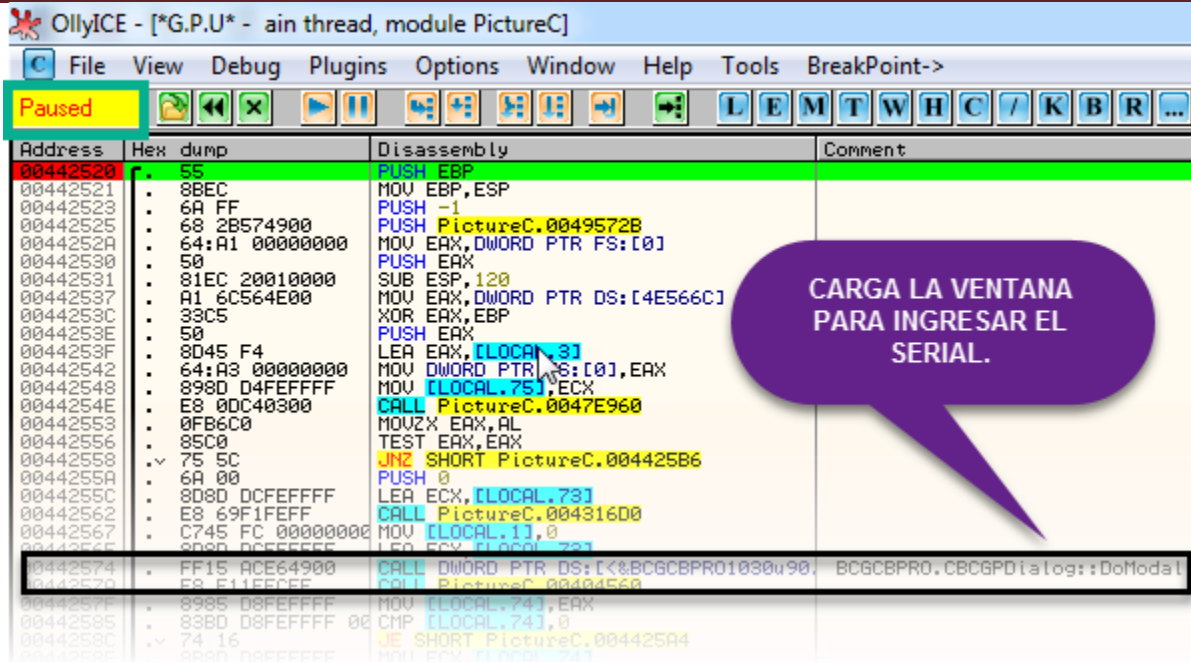
Show procedure Enter

Show call

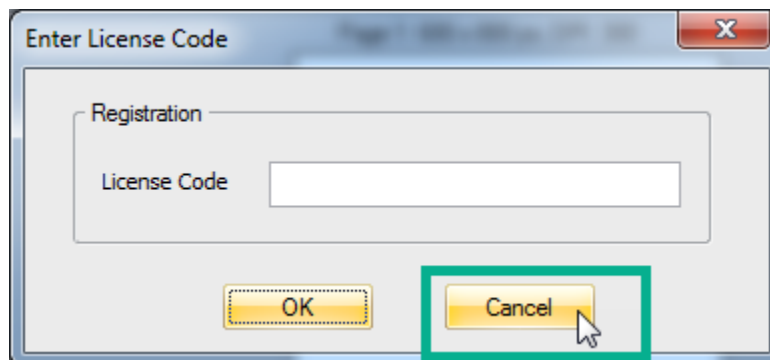
Execute to return F4


Abajo tenemos el procedimiento, así que le colocaremos un **<BREAKPOINT>**, **(BP)**. Notemos que todavía estamos **"PAUSADOS"**.

## [Tuto006 - PearlMountain Picture Collage Maker Pro v4.1.4 (MVC++9.0)(KeyGen.FUERZA.BRUTA)]



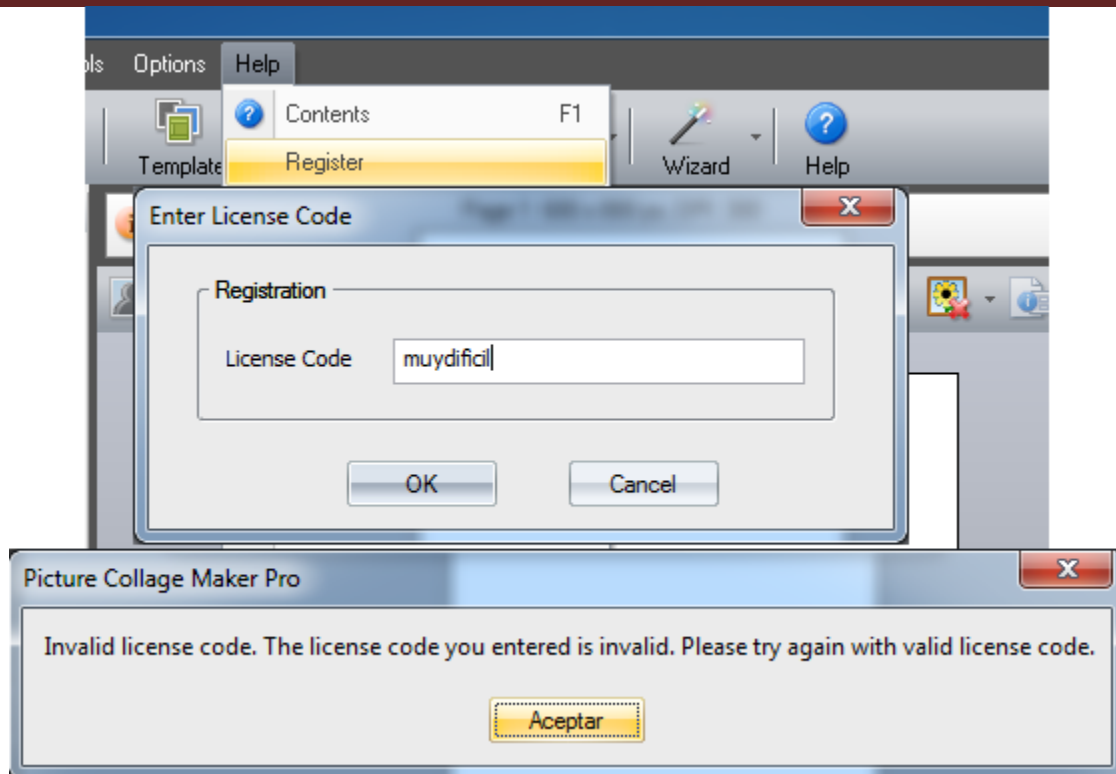
Desde este procedimiento se mostrará la ventana para ingresar el serial. Pueden hacer ese seguimiento; primero cierran la ventana del serial.



Y ahora corremos el programa con <F9> o . Recordemos que lo teníamos pausado. Cuando hagamos eso verán que paramos en ese procedimiento **PictureC.00442520** y al tracear con <F8> llega a otro <CALL> donde carga la ventana para ingresar el serial.

Bueno, todo lo anterior para tratar de explicar el uso del <CALL STACK> para nuestro beneficio, y entenderlo un poquito más. Entonces la idea es hacer lo mismo, pero cuando estemos en el <CHICO MALO> y saber cuál procedimiento escoger.

## [Tuto006 - PearlMountain Picture Collage Maker Pro v4.1.4 (MVC++9.0)(KeyGen.FUERZA.BRUTA)]



Ya estando con nuestro <CHICO MALO>, pausamos el OllyICE y vamos al <CALL STACK>.

Address	Stack	Procedure / arguments
000CE9B4	76C57910	user32.76C578D2
000CE9D0	5E7EE39A	user32.GetMessageW
000CE9D4	00A1CA78	pMsg = 00A1CA78
000CE9D8	00000000	hWnd = NULL
000CE9DC	00000000	MsgFilterMin = 0
000CE9E0	00000000	MsgFilterMax = 0
000CE9EC	5E7B280E	mfc90u.#1218
000CEA18	5E7CB89B	mfc90u.#6022
000CEA64	1007AF53	? <JMP.&mfc90u.#2208>
000CEA8C	10166329	BCGCBPRO.CBCGDialog::DoModal
000CF420	101662D1	BCGCBPRO.BCGPMessageBoxIndirect
000CF454	10166229	BCGCBPRO.BCGPMessageBoxEx
000CF474	004319B5	BCGCBPRO.BCGPMessageBox
000CF4E8	5E7DD227	PictureC.004317E0
000CF4F4	5E7DD409	mfc90u.5E7DD1EC
000CF510	5E7CDE60	mfc90u.#4681
000CF534	5E7AFF73	Includes mfc90u.5E7CDE60
000CF584	1007AEC8	<JMP.&mfc90u.#4702>
000CF598	5E7AF75B	Includes BCGCBPRO.1007AEC8
000CF644	5E7AF6CE	Includes mfc90u.5E7AF75B
000CF664	5E7AE2F4	Includes mfc90u.5E7AF6CE
000CF6CC	5E7AE580	? mfc90u.#1067
000CF6F0	5E79C247	? mfc90u.#1274


Es un <CALL STACK> más largo que el anterior, pero aplicando lo tratado anteriormente, aquí el procedimiento de interés es **PictureC.004317E0**. Fijo desde este podemos tracear y ver donde carga al insoportable <CHICO MALO> y de paso por ahí pillamos lo que sucede con nuestro serial. Se preguntarán: "¿por qué este y no otro?, y dirán: "¿claro que gracia, como ya lo hizo, pues de memoria lo hace?, muy cierto y el detalle está en analizar e ir entendiendo todo lo que sale ahí, y si miramos son <CALLS> de APIS cargadas desde .DLL de Windows. Recordemos, que por lo que entiendo, las validaciones las hace el programa, ya sea atreves de su .EXE o con sus propios módulos (.DLL) como es este caso. Aquí yo imaginándome casos futuros, debemos agudizar el ojo y reconocer eso, pienso que veremos <CALLS> que hace el programa y sus módulos, y debemos escoger cuál escoger o bueno revisarlos

## [Tuto006 - PearlMountain Picture Collage Maker Pro v4.1.4 (MVC++9.0)(KeyGen.FUERZA.BRUTA)]

todos y ver cuál nos dirige por el camino correcto. Al fin y al cavo estamos es en cracking, ensayo y error.

Bueno, después de esa carreta que pienso yo, es lo principal de un tuto; seguimos con el crackeo. Vallamos a ese procedimiento desde ese **<CALL>** **PictureC.004317E0** teniéndolo seleccionado y **<ENTER>**.

Address	Hex dump	Disassembly
004317E0	55	PUSH EBP
004317E1	8BEC	MOV EBP,ESP
004317E3	6A FF	PUSH -1
004317E5	68 41414900	PUSH PictureC.00494141
004317EA	64:A1 00000000	MOV EAX,DWORD PTR FS:[0]
004317F0	50	PUSH EAX
004317F1	83EC 38	SUB ESP,38
004317F4	A1 6C564E00	MOV EAX,DWORD PTR DS:[4E566C]
004317F9	33C5	XOR EAX,EBP
004317FB	8945 E0	MOV [LOCAL.8],EAX
004317FE	50	PUSH EAX
004317FF	8D45 F4	LEA EAX,[LOCAL.3]
00431802	64:A3 00000000	MOV DWORD PTR FS:[0],EAX
00431808	894D D4	MOV [LOCAL.11],ECX
0043180B	68 048C4A00	PUSH PictureC.004A8C04
00431810	8D4D EC	LEA ECX,[LOCAL.5]
00431813	FF15 1CF14900	CALL DWORD PTR DS:[&f90u.#286]
00431819	C745 FC 00000000	MOV [LOCAL.13],0
00431820	8D45 EC	LEA EAX,[LOCAL.5]

Estando parados en el inicio de ese procedimiento **004317E0** le ponemos su **<BREAK POINT>** y seguimos como con el **<CALL STAK>** anterior de ingresar el serial, entonces seguimos la ejecución del OllyICE con **<F9>** () , aceptamos al **<CHICO MALO>** y volvemos a meter nuestro serial para parar en ese procedimiento (**004317E0**). Si el programa se les cierra mostrándoles el mensaje de error de Windows, no hay lio, reiniciarlo en el OllyDBG y meten el serial, y con eso terminamos parados en el **<BREAKPOINT 004317E0>**. De aquí en adelante estamos en la **<ZONA CALIENTE>**. Ahora todo se reduce a tracear y tracear e ir reconociendo las zonas de interés que son más calientes aún.

Address	Hex dump	Disassembly	Registers (FPU)
00431838	8D4D EC	LEA ECX,[LOCAL.5]	EAX 0474A310 UNICODE "myydificil"
0043183B	FF15 DCF04900	CALL DWORD PTR DS:[&f90u.#909]	ECX 000CF008
00431841	50	PUSH EAX	EDX 00000000
00431842	FF15 4CD44900	CALL DWORD PTR DS:[&AlbumDesignCore.winGetSizeANSI]	EBX 00000001
00431848	83C4 04	ADD ESP,4	ECX 00000000
0043184B	85C0	TEST EAX,EAX	EBX 00000001
0043184D	75 09	JNZ SHORT PictureC.00431858	ECX 00000000
0043184F	C745 D0 00000000	MOV [LOCAL.12],0	EBX 00000001

Cuando lleguemos a **00431842** podemos ver un **<CALL>** que va a otro módulo que hace parte del programa y retorna la longitud+1 de nuestro serial siendo **EAX=0x0B**.

Address	Hex dump	Disassembly	Registers (FPU)
00431838	8D4D EC	LEA ECX,[LOCAL.5]	EAX 0000000B
0043183B	FF15 DCF04900	CALL DWORD PTR DS:[&f90u.#909]	ECX 00000000
00431841	50	PUSH EAX	EDX 00000000
00431842	FF15 4CD44900	CALL DWORD PTR DS:[&AlbumDesignCore.winGetSizeANSI]	EBX 00000001
00431848	83C4 04	ADD ESP,4	ECX 00000000
0043184B	85C0	TEST EAX,EAX	EBX 00000001
0043184D	75 09	JNZ SHORT PictureC.00431858	ECX 00000000

Después de tracear varias veces podemos ver que para nuestro objetivo no es relevante, pero lo que quería hacer notar es que utiliza funciones propias de otro lugar y es lo nuevo que he podido aprender en este tuto. Traciamos con **<F8>** hasta llegar a lo que realmente nos interesa y es la dirección **004318C2**.

# [Tuto006 - PearlMountain Picture Collage Maker Pro v4.1.4 (MVC++9.0)(KeyGen.FUERZA.BRUTA)]

004318B8	. 8D45 F8	LEA EAX, [LOCAL.4]
004318BB	. 50	PUSH EAX
004318BC	. 8B0D B05C4E00	MOV ECX, DWORD PTR DS:[4E5C4E00]
004318C2	. E8 693A0500	CALL PictureC.00485330
004318C7	. 8945 E8	MOV [LOCAL.6], EAX
004318CA	. 837D E8 00	CMP [LOCAL.6], 0
004318CE	. 0F84 B1000000	JE PictureC.00431985
004318D4	. 8D4D EC	LEA ECX, [LOCAL.5]
004318D7	. 51	PUSH ECX
004318D8	. FF15 54D44900	CALL DWORD PTR DS:[&AlbumDesignCore.RegConfig::GetIn
004318DE	. 83C0 08	ADD EAX, 8
004318E1	. 8BC8	MOV ECX, EAX
004318E3	. FF15 D8F04900	CALL DWORD PTR DS:[&mfc90u.#811]
004318E9	. FF15 54D44900	CALL DWORD PTR DS:[&AlbumDesignCore.RegConfig::GetIn
004318EF	. 8BC8	MOV ECX, EAX
004318F1	. FF15 58D44900	CALL DWORD PTR DS:[&AlbumDesignCore.RegConfig::SaveD
004318F7	. 8D4D E4	LEA ECX, [LOCAL.7]
004318FA	. FF15 E0F04900	CALL DWORD PTR DS:[&mfc90u.#296]
00431900	. C645 FC 02	MOV BYTE PTR SS:[EBP-4], 2
00431904	. 8B55 E8	MOV EDX, [LOCAL.6]

→ ZONA CALIENTE

→ SALTO DECISIVO

Como podemos ver en la imagen de arriba en ese <004318C2 CALL> se harán las trapisondas del serial y de acuerdo a lo que retorne non enviará al paraíso o al infierno. Yo no sé ustedes, pero lo que es de mí, prefiero el paraíso y vamos a resolver esto para ganarnos el boleto directo. Entremos a <004318C2 CALL> con <F7>.

Address	Hex dump	Disassembly
00485330	. 55	PUSH EBP
00485331	. 8BEC	MOV EBP, ESP
00485333	. 83EC 08	SUB ESP, 8
00485336	. 894D F8	MOV [LOCAL.2], ECX
00485339	. C745 FC 00000000	MOV [LOCAL.1], 0
00485340	. 8B45 F8	MOV EAX, [LOCAL.2]
00485343	. 05 8C000000	ADD EAX, 8C
00485348	. 50	PUSH EAX
00485349	. 8B4D F8	MOV ECX, [LOCAL.2]
0048534C	. 8B91 88000000	MOV EDX, DWORD PTR DS:[ECX+88]
00485352	. 52	PUSH EDX
00485353	. 8B45 08	MOV EAX, [ARG.1]
00485356	. 50	PUSH EAX
00485357	. FF15 1CD74900	CALL DWORD PTR DS:[&AlbumDesignCore.CheckRandomSerialCode]
0048535D	. 83C4 0C	ADD ESP, 0C
00485360	. 0FB6C8	MOVZX ECX, AL
00485363	. 85C9	TEST ECX, ECX
00485365	. 74 09	JE SHORT PictureC.00485370
00485367	. C745 FC 01000000	MOV [LOCAL.1], 1

EBP=000CF01C

Local calls from 0041A742, 004318C2, 00485485, 00485595

Podemos ver en las observaciones del OllyDBG que este procedimiento se llama desde 4 <CALLS>, entonces comprueba nuestro serial desde varios caminos. A seguir traceado con <F7>. Lleguemos a 00485348.

Address	Hex dump	Disassembly	Registers (FPU)
00485339	. C745 FC 00000000	MOV [LOCAL.1], 0	EAX 0232AA20 ASCII "CM4PP"
00485340	. 8B45 F8	MOV EAX, [LOCAL.2]	ECX 0232AA20
00485343	. 05 8C000000	ADD EAX, 8C	EDX 02416B20 ASCII "muydificil1"
00485348	. 50	PUSH EAX	EBX 00000001
00485349	. 8B4D F8	MOV ECX, [LOCAL.2]	ESP 000CEFA4
0048534C	. 8B91 88000000	MOV EDX, DWORD PTR DS:[ECX+88]	EBP 000CEFB4
00485352	. 52	PUSH EDX	ESI 00000000
00485353	. 8B45 08	MOV EAX, [ARG.1]	EDI 00000111
00485356	. 50	PUSH EAX	
00485357	. FF15 1CD74900	CALL DWORD PTR DS:[&AlbumDesignCore.CheckRandomSerialCode]	
0048535D	. 83C4 0C	ADD ESP, 0C	

De aquí en adelante las cosas las explicaré de forma ya final, que después de tracear y tracear les hallé su sentido. En EAX tenemos EAX=0232AA20 ASCII "CM4PP". Este es la primera parte del verdadero serial y nos da el tipo de licencia que en este caso es "Personal License". Avancemos un poquitín más hasta llegar a 0048534C.

## [Tuto006 - PearlMountain Picture Collage Maker Pro v4.1.4 (MVC++9.0)(KeyGen.FUERZA.BRUTA)]

```

00485348 . 50          PUSH EAX
00485349 . 8B4D F8     MOV ECX, [LOCAL.2]
0048534C . 8B91 80000000 MOV EDX, DWORD PTR DS:[ECX+80]
00485352 . 52          PUSH EDX
00485353 . 8B45 08     MOV EAX, [ARG.1]
00485356 . 50          PUSH EAX
00485357 . FF15 1CD74900 CALL DWORD PTR DS:[<&AlbumDesignCore.CheckRandomSerialCode>] AL
DS:[0232AAA8]=00017219
EDX=02416B20, (ASCII "muydificil1")

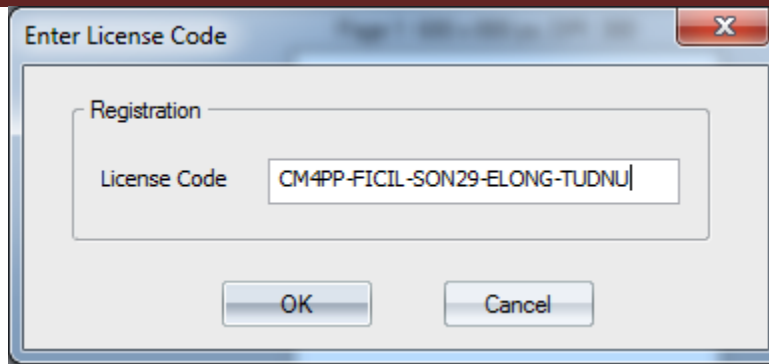
```

Pues he ahí un valor constante de **0x00017219**, ese no es ni nada menos, ni nada más que el valor a ser comparado con el resultado de las operaciones de nuestro serial; entonces intuimos que una de esas operaciones debe dar **0x17219**. Seguimos traceado hasta llegar a **<00485357 CALL>** y que es muy evidente no, **CheckRandomSerialCode**. Entremos ahí para ver qué sucede con nuestro serial. Ya estando adentro traceamos con **<F8>** para evitar en todos esos **<CALLS>**.

Address	Hex dump	Disassembly	Registers (FPU)
001D5B26	8D4D EC	LEA ECX, DWORD PTR SS:[EBP-14]	EAX 000CEFA4
001D5B29	FF15 60472100	CALL DWORD PTR DS:[&mfc90u.#5939]	ECX 0474B300 UNICODE "MUYDIFICIL1"
001D5B2F	8D4D EC	LEA ECX, DWORD PTR SS:[EBP-14]	EDX 00000000
001D5B32	FF15 64472100	CALL DWORD PTR DS:[&mfc90u.#4494]	EBX 00000001
001D5B38	8D4D EC	LEA ECX, DWORD PTR SS:[EBP-14]	ESP 000CEFA4
001D5B3B	FF15 38472100	CALL DWORD PTR DS:[&mfc90u.#3185]	EBP 000CEFA8
001D5B41	83F8 1D	CMP EAX, 1D	ESI 00000000
001D5B44	74 1C	JE SHORT AlbumDes.001D5B62	EDI 00000111
001D5B46	C645 D3 00	MOV BYTE PTR SS:[EBP-2D], 0	
001D5B4A	C745 FC FFFFFFFF	MOV DWORD PTR SS:[EBP-4], -1	
001D5B51	8D4D EC	LEA ECX, DWORD PTR SS:[EBP-14]	
001D5B54	FF15 F0462100	CALL DWORD PTR DS:[&mfc90u.#600]	
001D5B5A	8A45 D3	MOV AL, BYTE PTR SS:[EBP-2D]	
001D5B5D	E9 CC010000	JMP AlbumDes.001D5D2E	
001D5B62	6A 05	PUSH 5	
001D5B64	8D4D EC	LEA ECX, DWORD PTR SS:[EBP-14]	
001D5B67	FF15 60472100	CALL DWORD PTR DS:[&mfc90u.#2676]	
001D5B6D	0FB7C0	MOVBX EAX, AX	
001D5B70	83F8 2D	CMP EAX, 2D	
001D5B73	75 39	JNZ SHORT AlbumDes.001D5BAE	
001D5B75	6A 0B	PUSH 0B	
001D5B77	8D4D EC	LEA ECX, DWORD PTR SS:[EBP-14]	
001D5B7A	FF15 60472100	CALL DWORD PTR DS:[&mfc90u.#2676]	
001D5B80	0FB7C8	MOVBX ECX, AX	
001D5B83	83F9 2D	CMP ECX, 2D	
001D5B86	75 26	JNZ SHORT AlbumDes.001D5BAE	
001D5B88	6A 11	PUSH 11	
001D5B8A	8D4D EC	LEA ECX, DWORD PTR SS:[EBP-14]	
001D5B8D	FF15 60472100	CALL DWORD PTR DS:[&mfc90u.#2676]	
001D5B93	0FB7D0	MOVBX EDX, AX	
001D5B96	83FA 2D	CMP EDX, 2D	
001D5B99	75 13	JNZ SHORT AlbumDes.001D5BAE	
001D5B9B	6A 17	PUSH 17	
001D5B9D	8D4D EC	LEA ECX, DWORD PTR SS:[EBP-14]	
001D5BA0	FF15 60472100	CALL DWORD PTR DS:[&mfc90u.#2676]	
001D5BA6	0FB7C0	MOVBX EAX, AX	
001D5BA9	83F8 2D	CMP EAX, 2D	
001D5BAC	74 1C	JE SHORT AlbumDes.001D5BCA	
001D5BAE	C645 D2 00	MOV BYTE PTR SS:[EBP-2E], 0	
001D5BB2	C745 FC FFFFFFFF	MOV DWORD PTR SS:[EBP-4], -1	
001D5BB9	8D4D EC	LEA ECX, DWORD PTR SS:[EBP-14]	
001D5BBC	FF15 F0462100	CALL DWORD PTR DS:[&mfc90u.#600]	
001D5BC2	8A45 D2	MOV AL, BYTE PTR SS:[EBP-2E]	

Aquí hallamos cómo debe ser el formato verdadero de nuestro serial. En **001D5B32** se pasa nuestro serial a mayúsculas. Lo resaltado en **VERDE** compara la longitud de nuestro serial con **0x1D (29)**, entonces nuestro serial debe tener 29 caracteres; y lo resaltado en **AZUL** compara que después de cada cinco caracteres tengamos un **"-"** = **0x2D**. Bueno, armemos nuestro serial con lo que hemos averiguado, **"CM4PP-FICIL-SON29-ELONG-TUDNU"**. Coloquemos un **<BREAKPOINT>** en **001D5B73** para llegar más rápido la próxima vez. **<F9>** para volver a ingresar nuestro nuevo serial.

## [Tuto006 - PearlMountain Picture Collage Maker Pro v4.1.4 (MVC++9.0)(KeyGen.FUERZA.BRUTA)]



Cuando lleguemos a nuestro último <BREAKPOINT>, sigamos traceado y viendo cómo pasamos por los lugares correctos. Leguemos hasta **001D5C10**.

001D5C0C	8D4D F0	LEA ECX,DWORD PTR SS:[EBP-10]
001D5C0F	51	PUSH ECX
001D5C10	E8 3B040000	CALL AlbumDes.001D6050
001D5C15	83C4 08	ADD ESP,8
001D5C18	0FB6D0	MOVZX EDX,AL
001D5C1B	85D2	TEST EDX,EDX
001D5C1D	74 29	JE SHORT AlbumDes.001D5C48
001D5C1F	C645 CB 00	MOV BYTE PTR SS:[EBP-35],0

Ese <CALL> compara nuestra primera parte del serial "CM4PP" y que ya sabemos es correcta. A seguir traceando.

001D5C48	C745 D8 00	MOV DWORD PTR SS:[EBP-28],0
001D5C4F	EB 09	JMP SHORT AlbumDes.001D5C5A
001D5C51	8B45 D8	MOV EAX,DWORD PTR SS:[EBP-28]
001D5C54	83C0 01	ADD EAX,1
001D5C57	8945 D8	MOV DWORD PTR SS:[EBP-28],EAX
001D5C5A	837D D8 04	CMP DWORD PTR SS:[EBP-28],4
001D5C5E	7D 6E	JE SHORT AlbumDes.001D5CCE
001D5C60	6A 05	PUSH 5
001D5C62	8B4D D8	MOV ECX,DWORD PTR SS:[EBP-28]
001D5C65	6BC9 06	IMUL ECX,ECX,6
001D5C68	83C1 06	ADD ECX,6
001D5C6B	51	PUSH ECX
001D5C6C	8D55 C4	LEA EDX,DWORD PTR SS:[EBP-3C]
001D5C6F	52	PUSH EDX
001D5C70	8D4D EC	LEA ECX,DWORD PTR SS:[EBP-14]
001D5C73	FF15 5C472	CALL DWORD PTR DS:[<&mfc90u.#4519>]
001D5C79	8945 B4	MOV DWORD PTR SS:[EBP-4C],EAX
001D5C7C	8B45 B4	MOV EAX,DWORD PTR SS:[EBP-4C]
001D5C7F	8945 B0	MOV DWORD PTR SS:[EBP-50],EAX
001D5C82	C645 FC 04	MOV BYTE PTR SS:[EBP-4],4
001D5C86	8B4D B0	MOV ECX,DWORD PTR SS:[EBP-50]
001D5C89	FF15 FC462	CALL DWORD PTR DS:[<&mfc90u.#909>]
001D5C8F	50	PUSH EAX
001D5C90	8D4D D4	LEA ECX,DWORD PTR SS:[EBP-2C]
001D5C93	FF15 58472	CALL DWORD PTR DS:[<&mfc90u.#306>]
001D5C99	C645 FC 06	MOV BYTE PTR SS:[EBP-4],6
001D5CA0	8D4D C4	LEA ECX,DWORD PTR SS:[EBP-3C]
001D5CA6	FF15 F0462	CALL DWORD PTR DS:[<&mfc90u.#600>]
001D5CA9	8D4D D4	LEA ECX,DWORD PTR SS:[EBP-2C]
001D5CAF	FF15 B8462	CALL DWORD PTR DS:[<&mfc90u.#2697>]
001D5CB0	50	PUSH EAX
001D5CB0	E8 8BFAFFF	CALL AlbumDes.HexToInt
001D5CB5	83C4 04	ADD ESP,4
001D5CB8	8B4D D8	MOV ECX,DWORD PTR SS:[EBP-28]

Vamos a decir qué hace cada zona resaltada, La zona **AZUL** es un contador que se repite cuatro veces y es para trabajar con las cuatro últimas partes de nuestro serial, "CM4PP-FICIL-SON29-ELONG-TUDNU". La zona **VERDE** lo único que hace es tomar la parte del serial según vaya el contador (zona **AZUL**) para luego entra al <001D5CB0 CALL AlbumDes.HexToInt>. Entremos a esa función. Vamos a ir por partes, con esta función. Empieza desde el carácter final hasta el primero para cada parte del serial que son de cinco caracteres.



# [Tuto006 - PearlMountain Picture Collage Maker Pro v4.1.4 (MVC++9.0)(KeyGen.FUERZA.BRUTA)]

Si es número hace esto

Si es letra MAYUSCULA hace esto

La imagen de arriba explica bien la cosa. Compara si es número o letra. Como no tenemos letras minúsculas **a-z** (0x61-0x7A), pues ahí nunca entraremos. Este se repite cuatro veces, que serían las cuatro partes del serial, "CM4PP-FICIL-SON29-ELONG-TUDNU". Al terminar una parte del serial, el valor de esa operación se guarda, para con eso tendremos cuatro valores.

Si es letra MAYUSCULA hace esto

Address	Hex dump
000CE64	35 2D 10 00 29 97 1D 00 80 69 0F 00 8E EE 1E 00
000CE67	50 8B 4B 02 50 D2 C5 00 10 F0 0C 00 AD 06 99 00
000CE6A	06 00 00 00 A4 EF 0C 00 5D 53 48 00 0C F0 0C 00
000CE6D	19 72 01 00 CC AC 42 02 40 AC 42 02 00 00 00 00
000CE6F	1C F0 0C 00 C7 18 43 00 0C F0 0C 00 43 4D 34 5A

Después de salir del <00955CB0 CALL>, recordemos de nuevo, que ahí se calcula un valor con la sección de nuestro serial, el cual será guardado 00955CBB. En el DUMP podemos ver esos cuatro valores. Esas direcciones pueden ser diferentes para cada uno.



## [Tuto006 - PearlMountain Picture Collage Maker Pro v4.1.4 (MVC++9.0)(KeyGen.FUERZA.BRUTA)]

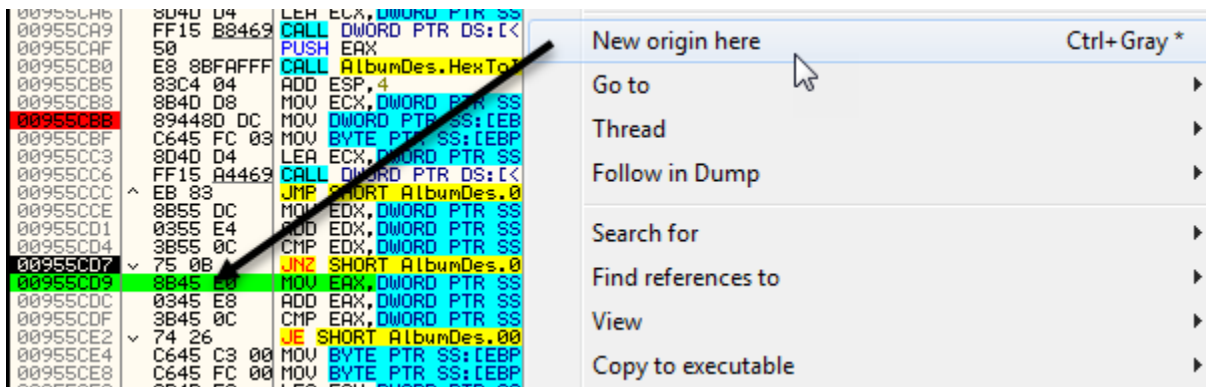
CM4PP-352D10 (FICIL) -29971D (SON29) -10000F00 (ELONG) -8EE1E00 (TUDNU)

Miremos lo resaltado en **VIOLETA**. La primera parte resaltada en **VIOLETA** suma lo hallado de la parte 2 más parte 4 del serial y lo compara con nuestra famosa constante, **0x17239**.

$$0x102D35 + 0x0F6980 = 0x17239$$

$$0x1F96B5 = 0x17239$$

Como no hay igualdad, para fuera nos manda. Supongamos que hubo igualdad, para eso cambiemos **FLAG-Z** a 1 o con <clíc derecho-New origin here>



Con eso llegamos a la segunda parte resaltada en **VIOLETA** que suma lo hallado de la parte 3 más parte 5 del serial y lo compara con nuestra famosa constante, **0x17239**.

$$0x1D9729 + 0x1EEE8E = 0x17239$$

$$0x3C85B7 = 0x17239$$

Listo, tenemos todo para sacar nuestro serial para la "**Licencia personal**". Supongo que se podría planear una solución mediante eliminación o métodos de aproximación de matrices y condicionar el valor de las variables entre **0x30** a **0x39** y **0x41** a **0x5A**, pero eso para mí está complicado, entonces mejor me voy a hallarlo a <**FUERZA BRUTA**>.

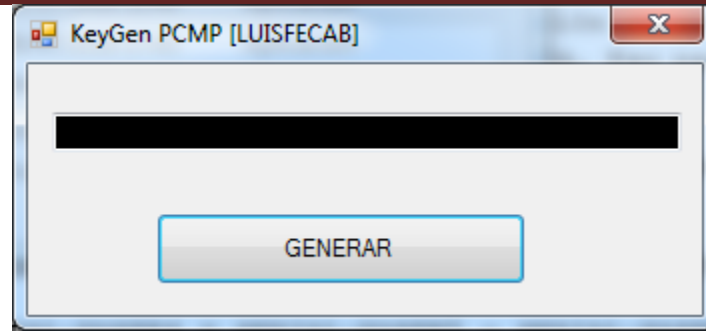
Lo bueno del serial es que con solamente una igualdad que sume **0x17239** sirve, ya que la otra mitad es igual, pues utilizamos esos mismos valores. Explicándome:

SERIAL\_PARTE2 = SERIAL\_PARTE3 o SERIAL\_PARTE5

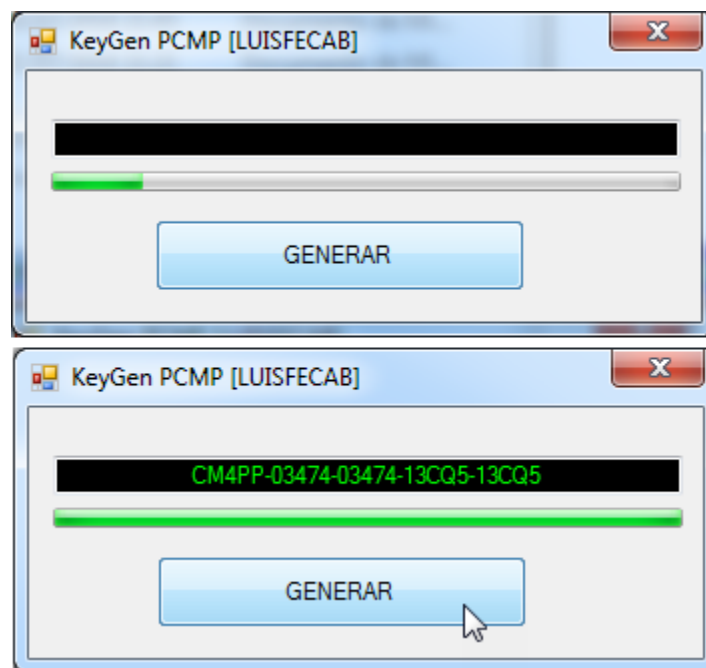
SERIAL\_PARTE4 = SERIAL\_PARTE5 o SERIAL\_PARTE3

No importa la posición porque es una suma. Con el tuto viene adjunto el KeyGen y su SRC.

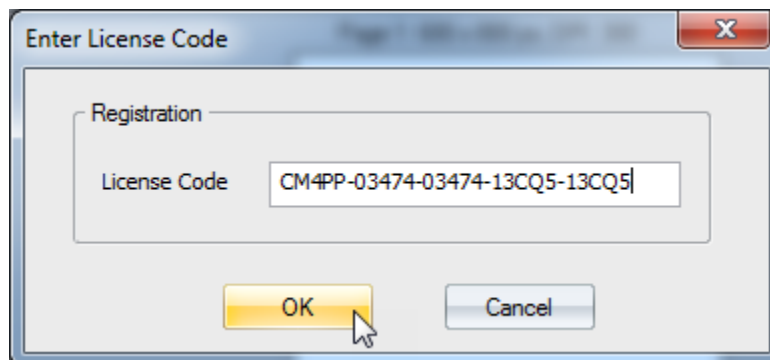
## [Tuto006 - PearlMountain Picture Collage Maker Pro v4.1.4 (MVC++9.0)(KeyGen.FUERZA.BRUTA)]



Ahí tenemos el KeyGen en desarrollo todavía. Toma su tiempo, pero vale la pena si lo comparas con que obtendremos nuestro serial.



Ahí tenemos el serial, "**CM4PP-03474-03474-13CQ5-13CQ5**". Como ven repetimos las partes en el serial, ahora solo nos queda probarlo.



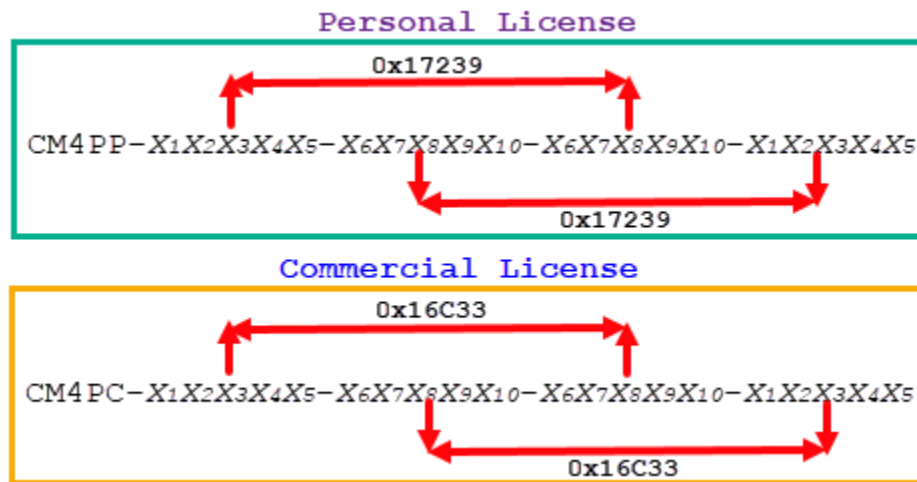
Es una maravilla, funciona bien, y ahora debemos hacerlo para la "**Licencia comercial**", que prácticamente es lo mismo.

## [Tuto006 - PearlMountain Picture Collage Maker Pro v4.1.4 (MVC++9.0)(KeyGen.FUERZA.BRUTA)]

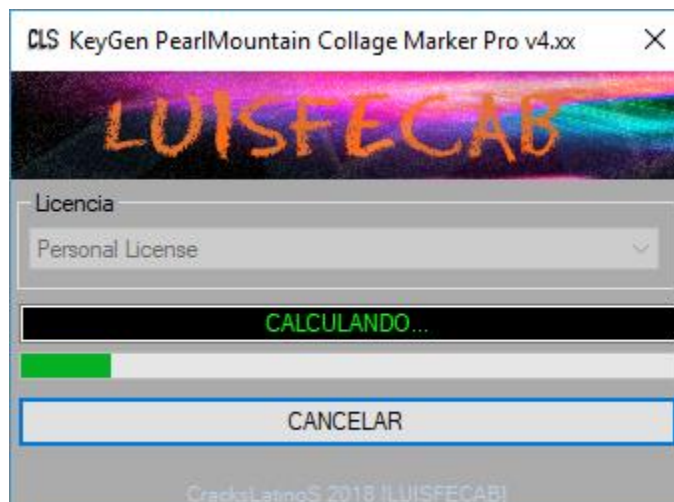
Address	Hex dump	Disassembly	Registers (FPU)
00485370	8B55 F8	MOV EDX, DWORD PTR SS:[EBP-8]	EAX 023FC300
00485373	81C2 10010	ADD EDX, 110	ECX 00000000
00485379	52	PUSH EDX	EDX 00000000 ASCII "CM4PC"
0048537D	8B55 F8	MOV EDX, DWORD PTR SS:[EBP-8]	EBX 00000001
00485380	8B55 08	MOV EDX, DWORD PTR SS:[EBP+8]	ESP 000CE998
00485387	52	PUSH EDX	EBP 000CEFA4
00485388	FF15 1CD74	CALL DWORD PTR DS:[&AlbumDesignCore.CheckRandomSerialCode]	ESI 00000000
0048538E	83C4 0C	ADD ESP, 0C	EDI 00000111
00485391	0FB6C0	MOVZX EAX, AL	EIP 0048537D PictureC.00485
00485394	85C0	TEST EAX, EAX	C 0 ES 002B 32bit 0(FFFFFF)
00485396	74 07	JE SHORT PictureC.0048539F	P 0 CS 0023 32bit 0(FFFFFF)
00485398	C745 FC 02	MOV DWORD PTR DS:[EBP+4], 02FC45C7	A 0 SS 002B 32bit 0(FFFFFF)
DS:[023FC40C]=00016C33			Z 0 NS 002B 32bit 0(FFFFFF)
ECX=00000000			

CONSTANTE

La "Licencia comercial" viene dada por "CM4PC" y solo nos falta hallar la constante que viene siendo 0x16C33. El resto es completamente igual, así que no lo voy a repetir. Solo falta terminar el KeyGen que no es mucho tampoco, solo colocar esta constante y listo. Una explicación un poco más gráfica.



El resultado final del KeyGen me gustó. Como lo dije antes, se demora su tiempo, pero hace su trabajo y me saca los seriales para juntas licencias y he sacado varias y ninguno me ha fallado.



Les dejo otro par de seriales más.

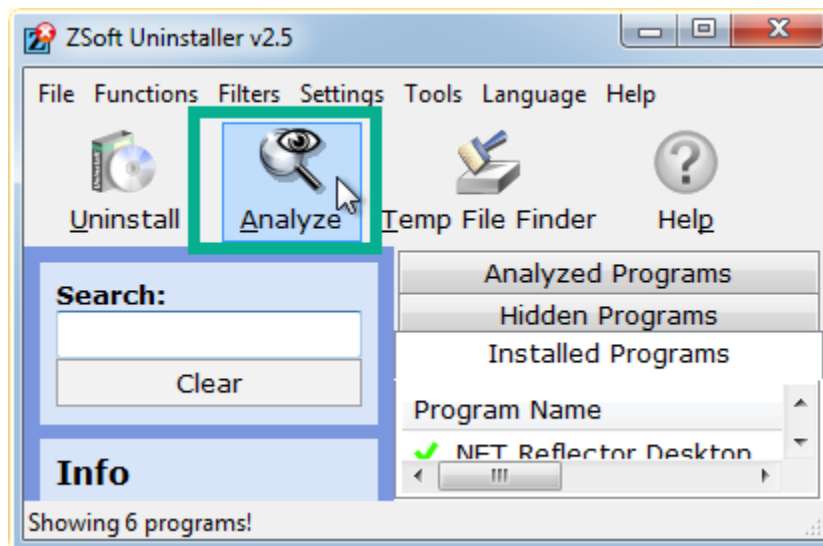
## [Tuto006 - PearlMountain Picture Collage Maker Pro v4.1.4 (MVC++9.0)(KeyGen.FUERZA.BRUTA)]

CM4PP-05BL7-115BI-115BI-05BL7  
CM4PP-13XY4-01DV5-01DV5-13XY4  
CM4PP-02YI3-11UF6-11UF6-02YI3  
CM4PP-11AS0-0564P-0564P-11AS0

CM4PC-0L3V7-00N3C-00N3C-0L3V7  
CM4PC-06532-0G701-0G701-06532  
CM4PC-144N1-01MBI-01MBI-144N1

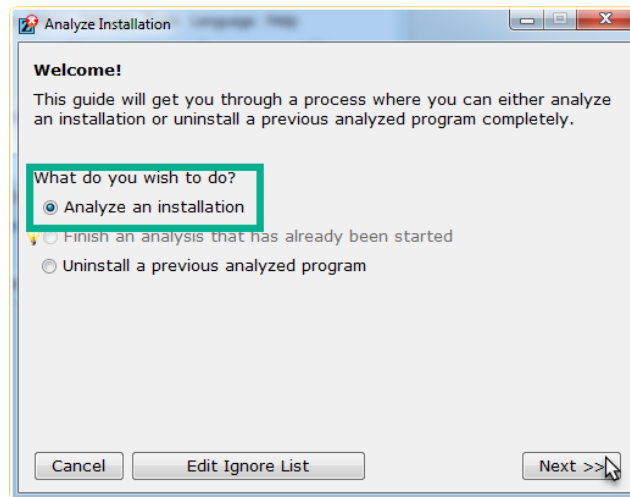
Ahora a buscar dónde se guarda el serial cuando lo registramos, y para eso utilizaremos el programa "**ZSoft Uninstaller v2.5**", que trae la opción para comparar todo lo que instala y los cambios que hace antes y después de instalado o registrado un programa. Nosotros solo debemos escoger la captura inicial y final, luego el programa nos muestra las diferencias.

Yo hice dos comparaciones, una antes de instalar el programa y después de instalado; y la revisé para ver qué nos reportaba el programa y para ir entendiendo cómo analizarlo, lo que pude observar es que se instala mucha cosa y hace cambios en el registro de Windows, pero nada relevante. La segunda captura la hice una vez instalado y después de registrarlo con uno de mis serials, y ahí si encontré lo que quería. Entonces como una guía para quien no ha utilizado el programa.

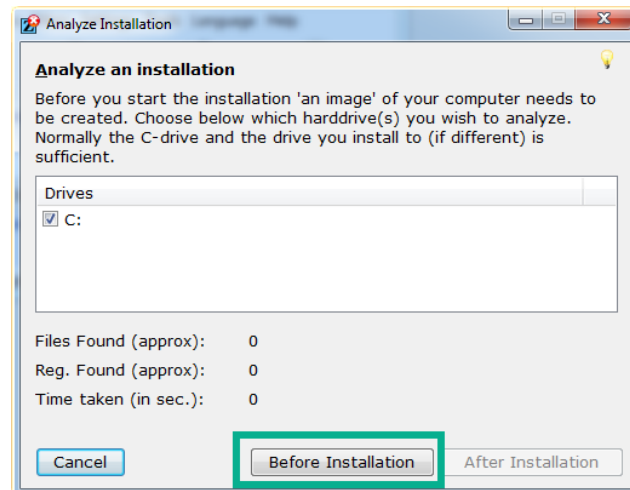


## [Tuto006 - PearlMountain Picture Collage Maker Pro v4.1.4 (MVC++9.0)(KeyGen.FUERZA.BRUTA)]

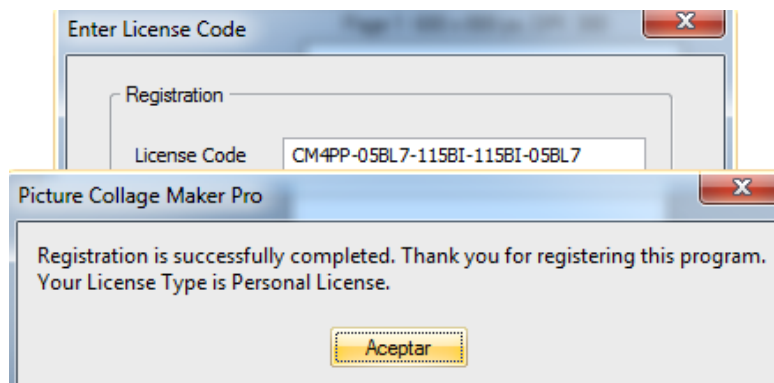
Iniciamos el programa y escogemos **"Analyze"** y ya con eso tenemos nuestro asistente.



Escogemos **"Analyze an installation"** y **"Next>>"**.

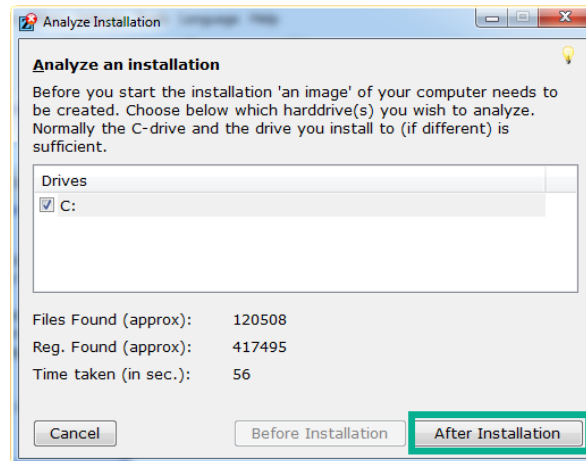


Ahí tomamos la captura inicial con **"Before Installation"**, que en nuestro caso es una vez instalado el programa. Esperamos a que la cree, tomará su tiempo en función de lo lleno que tengas tu disco. Una vez tomada la captura, registramos nuestro programa.

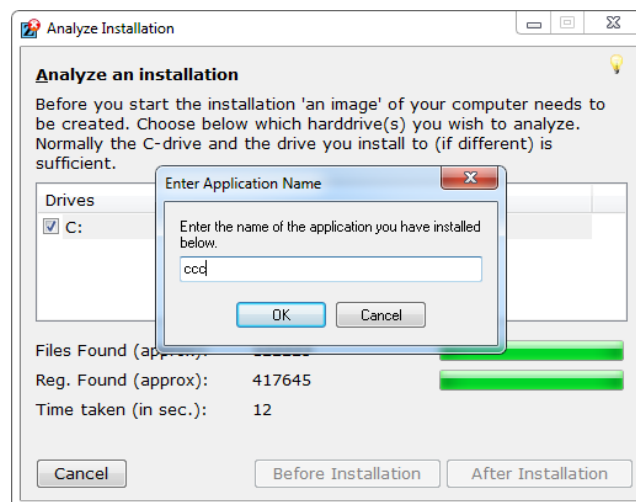


## [Tuto006 - PearlMountain Picture Collage Maker Pro v4.1.4 (MVC++9.0)(KeyGen.FUERZA.BRUTA)]

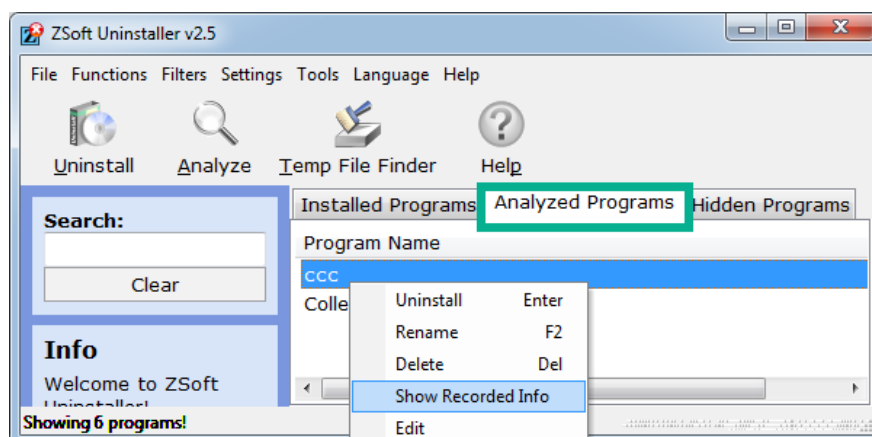
Ahí tenemos a nuestro <CHICO BUENO>. Le damos a aceptar y ahora tomamos nuestra segunda captura con "After Installation".



Después de tomar la captura, automáticamente hace la comparación y luego tu le colocas el nombre con que quieras guardarlo.



Ahora, solo debes abrirla y ver lo que hay de diferencia entre las dos capturas.



## [Tuto006 - PearlMountain Picture Collage Maker Pro v4.1.4 (MVC++9.0)(KeyGen.FUERZA.BRUTA)]

Listo, solo queda abrir bien el ojo y encontrar lo que te interesa.

```
REG ADDED! HKU S-1-5-21-2011889505-1001465476-2397396996-1000\Software\PearlMountain\PictureCollageMakerPro FirstUse bin:AAAAAA==
REG ADDED! HKU S-1-5-21-2011889505-1001465476-2397396996-1000\Software\PearlMountain\PictureCollageMakerPro InstallDay bin:MD4LAA==
REG ADDED! HKU S-1-5-21-2011889505-1001465476-2397396996-1000\Software\PearlMountain\PictureCollageMakerPro NoSaveLayoutState bin:AAAAAA==
REG ADDED! HKU S-1-5-21-2011889505-1001465476-2397396996-1000\Software\PearlMountain\PictureCollageMakerPro RegCode "CM4PP-05BL7-115BI-115BI-05BL7"
REG ADDED! HKU S-1-5-21-2011889505-1001465476-2397396996-1000\Software\PearlMountain\PictureCollageMakerPro UserName ""
```

Una maravilla, ahí está nuestro serial. Yo eliminé esa clave pensando que con eso era suficiente para volverlo trial, pero no, seguía registrado, pero eliminé la carpeta completa y ahí si quedó trial.

REG ADDED! HKU S-1-5-21-2011889505-1001465476-2397396996-1000\Software\PearlMountain\PictureCollageMakerPro

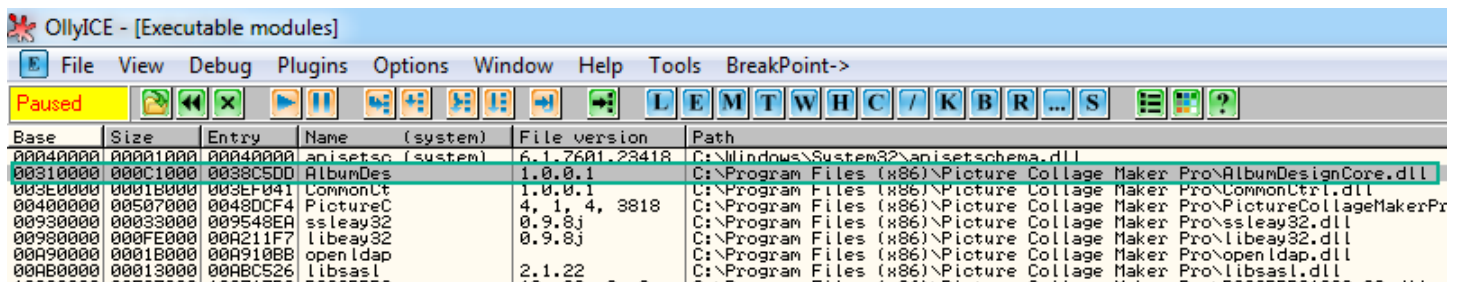
Suponiendo un poco pero no comprobándolo, podemos suponer que utiliza el serial para registrarse y que una o más de esas claves del registro creadas son utilizadas como comprobación de si ya está registrado.

Bueno, "hasta aquí me trajo el río". Misión cumplida.

## PARA TERMINAR

Contento y muy satisfecho con estos, mis pequeños logros; falta mucho pero mucho por aprender, pero ahí lo vamos llevando.

Podemos resaltar después de todo lo hecho que los programas pueden usar módulos externos para validar sus registros.



No sé si la solución del KeyGen a <FUERZA BRUTA> sea la más idónea, pero fue la que pude hallar, y la misión al fin y al cabo es vencer la protección, lograr el objetivo.

Tratar de sacar los seriales y hacer un KeyGen son de mucha ayuda para entender el ASSAMBLE, eso ayuda a ir agudizando la vista y el análisis.

Ya terminando, saludos para todos y espero que este tuto sea de ayuda para el que lo requiere y agradable para al lector.