

[Tuto007 - Nitro Pro v12.1.0.195 (Crack.Patch)(OllyDBG v1.10)]



Software	Nitro Pro v12.1.0.195
DESCARGA	https://www.gonitro.com/es/download
Protección	Serial. Activación por Internet.
Herramientas	Windows 7 Home Premium SP1 x32 Bits (S.O donde trabajamos.) OllyDBG v1.10 (Este Olly es el que vamos armando en el Curso OLLY DESDE CERO) Detect It Easy v1.01 RDG Packer Detector v0.7.6.2017 dUP2 Diablo's Universal Patcher v2.26 DESCARGAR HERRAMIENTAS DESCARGAR TUTO+ARCHIVOS
SOLUCIÓN	Crear un archivo Crackeado y su Patch.
AUTOR	LUISFECAB
RELEASE	Agosto 26 2018 [TUTORIAL 007]

INTRODUCCIÓN

Hola mi gente querida de la lista CracksLatinoS y más allá, como mis amigos de PerúCrackerS. Y por supuesto saludes a Ricardo Narvaja que gracias a él ya voy por mi tutorial 7; bueno gracias a todos y a cada uno que haya aportado su granito de arena para que los nuevos podamos salir adelante con nuestro Cracking.

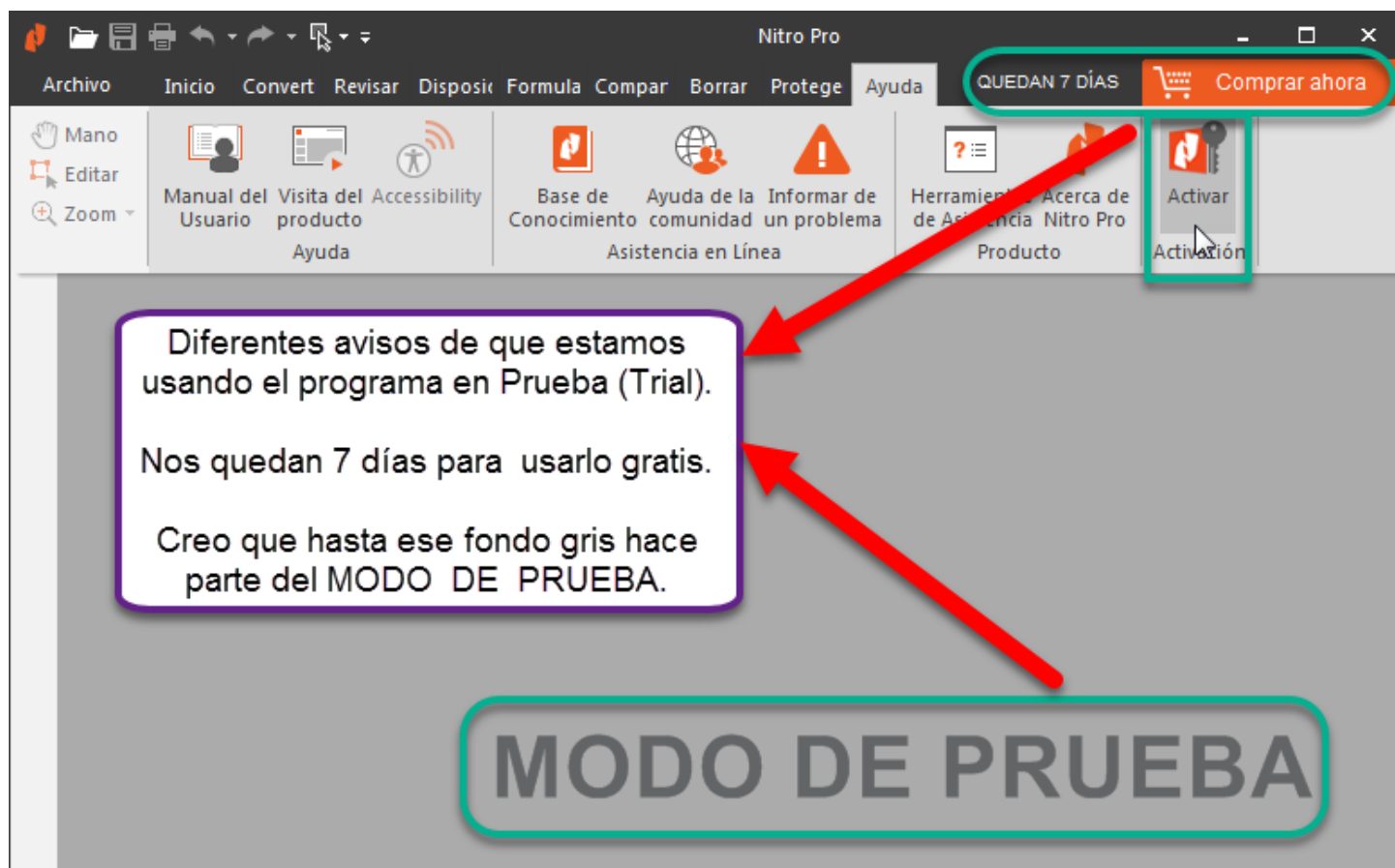
En este tuto traté de plasmar lo que hice para llegar a la solución que fue picando aquí, picando allá, me llené de **BREAKPOINTS (BP)**; de tracear y repetir muchas veces hasta ir reduciendo las zonas relevantes para luego encontrar el lugar correcto donde ocurriría la magia.

Este tuto es especial como el tuto anterior que está en las teorías numeradas [1658](#) porque es un programa que siempre uso, ya que es mi visor de archivos PDF y que lo utilizo para mucho más.

El programa viene con versiones para 32 y 64 bits. Todo el cracking lo hice para la versión de 32 Bits porque sigo fiel a mi Olly pero a partir de crackear la versión de 32 Bits paso a medirmele a la versión de 64 Bits con el **x64DBG**.

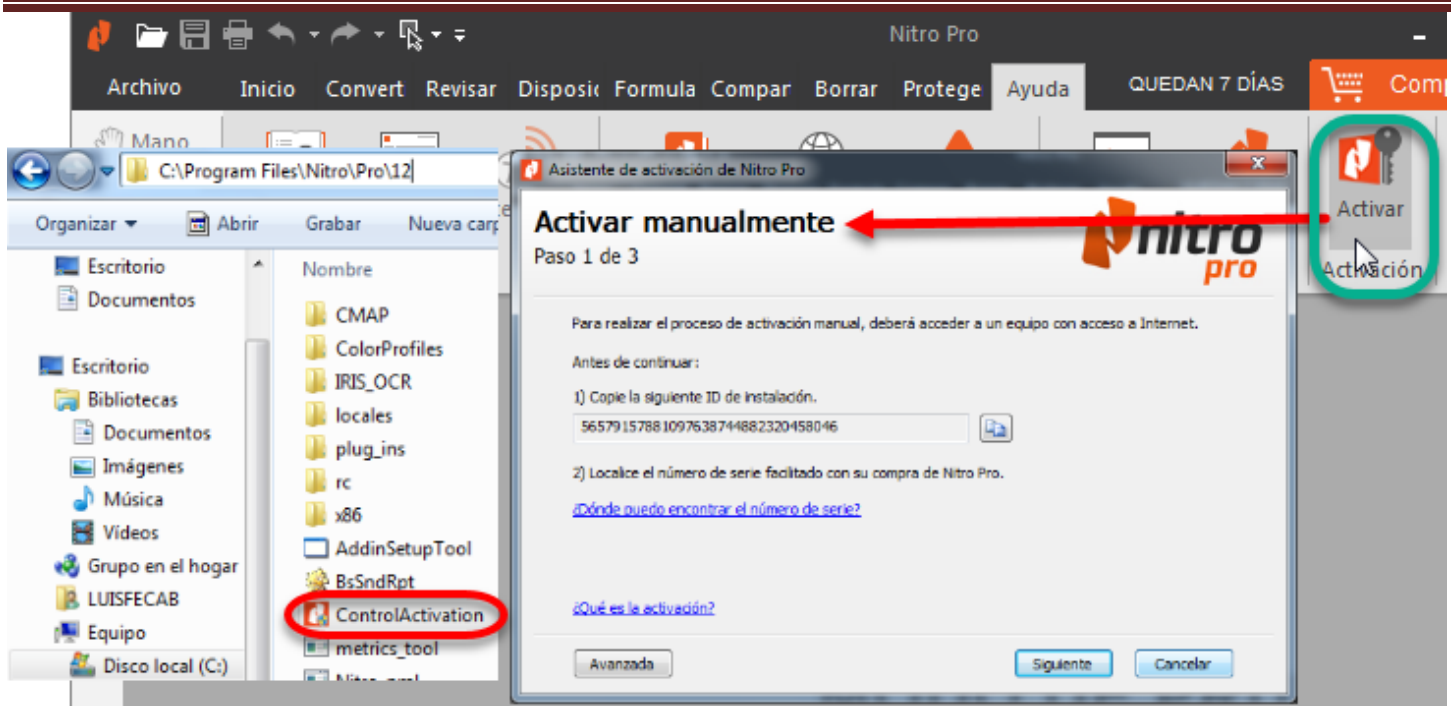
ANALISIS INICAL

Aquí hacemos lo de siempre. Instalamos el programa y lo iniciamos. Como el tuto lo hago ya después de unos días, no recuerdo exactamente lo que sale cuando lo ejecutamos por primera vez, pero lo relevante es que no nos muestra una **NAG** de trial cada vez que lo ejecutamos, si no que el programa se nos ejecuta mostrándonos varias marcas donde nos muestra que estamos utilizando la versión de prueba.

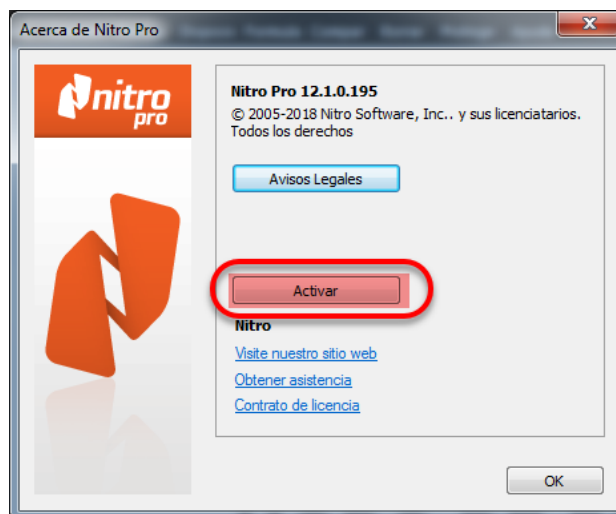


Tenemos muchos avisos de que estamos en **MODO PRUEBA (TRIAL)**. Ahí vemos que nos muestra que nos quedan 7 días; este programa te ofrece probarlo por 14 días, así que ya pasaron 7 días hasta que pude crackearlo y empecé a hacer este tuto para todos ustedes. Si leyeron mi tuto anterior recordarán que con ese programa se me pasaron los 15 días y no había terminado de crackearlo; eso muestra que ya me rinde más y que este "elefante" se mueve más rápido. Claro, debemos aclarar que aquí hago es un Parcheo y no un KeyGen. Sigamos analizando por encimita un poco más.

[Tuto007 - Nitro Pro v12.1.0.195 (Crack.Patch)(OllyDBG v1.10)]



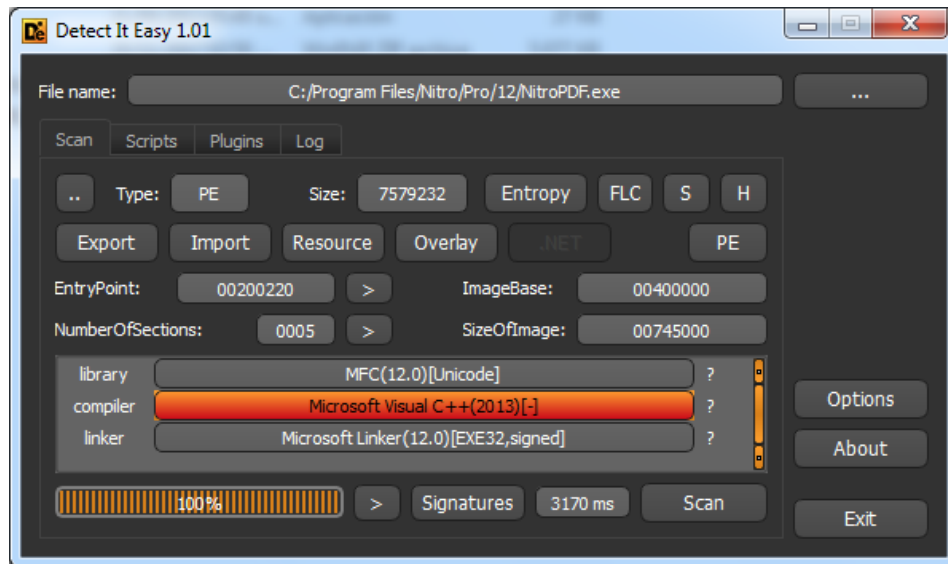
Ustedes podrán comprobar un poco más a fondo esta parte. Para activar el programa no es como antes que solo se debía ingresar un número de serie y listo, ahora la cosa se complicó ya que ese número de serie junto con el ID de instalación debe ser validado a través de Internet para generar un certificado de activación y con eso activarlo; como ven, al atacarle por ahí la cosa se pone más cuesta arriba; creo que si quisiéramos sería posible hacer un KeyGen porque he visto muchos activadores de ese tipo, como por ejemplo el AutoCAD o el SONY Vegas. Claro, "del dicho al hecho, hay mucho trecho" porque debes tener mucho más nivel de tu cracking para entenderlo y poder hacerlo; recuerden el ejemplo que nos pone el maestro Ricardo, "si eres un boxeador novato, no pretendas ganarle al campeón".



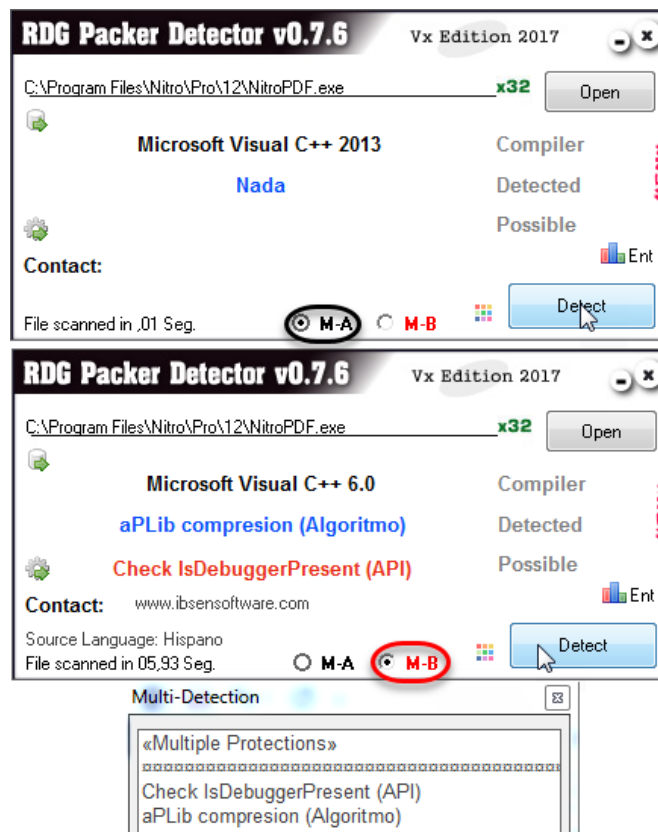
Y para ir terminando esta ojeada superficial, revisamos el "About" que como lo tenemos en español sería: "Acerca de Nitro Pro"; ahí vemos que tenemos un botón que nos invita una vez más a "Activar" el programa.

[Tuto007 - Nitro Pro v12.1.0.195 (Crack.Patch)(OllyDBG v1.10)]

Listo, cerremos el programa porque para la próxima lo cargaremos en nuestro Olly. Vayamos más al fondo. Mirémoslo con el <<**Detect It Easy (DIE) v1.01 Win**>> para ver si está empacado.



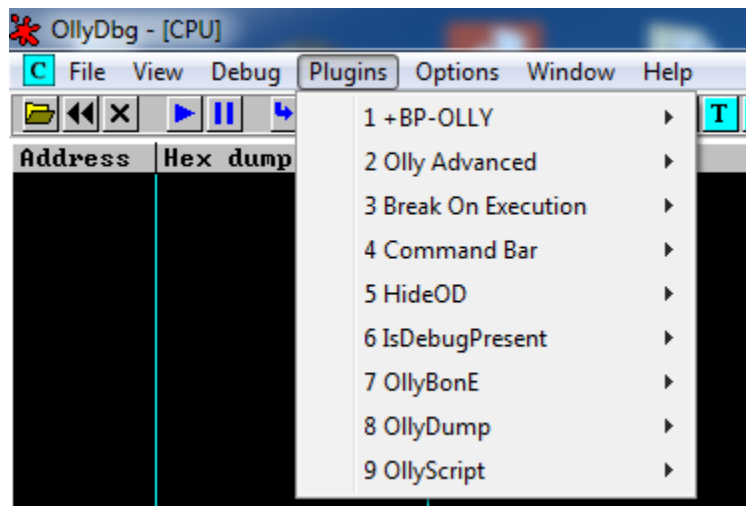
Pues está hecho en <<**Microsoft Visual C++**>> y no está empacado, claro que por ahora en el menú nada de empacados. Y un último vistazo pero con el <<**RDG Packer Detector v0.7.6.2017**>>.



M-A nos confirma lo dicho por el <<**Detect It Easy (DIE) v1.01 Win**>> y el scan con **M-B** nos muestra que lo comprimieron con el algoritmo <<**aPLib compression**>>, este algoritmo por fortuna no se comporta como los odiados Packers, este hace la tarea

[Tuto007 - Nitro Pro v12.1.0.195 (Crack.Patch)(OllyDBG v1.10)]

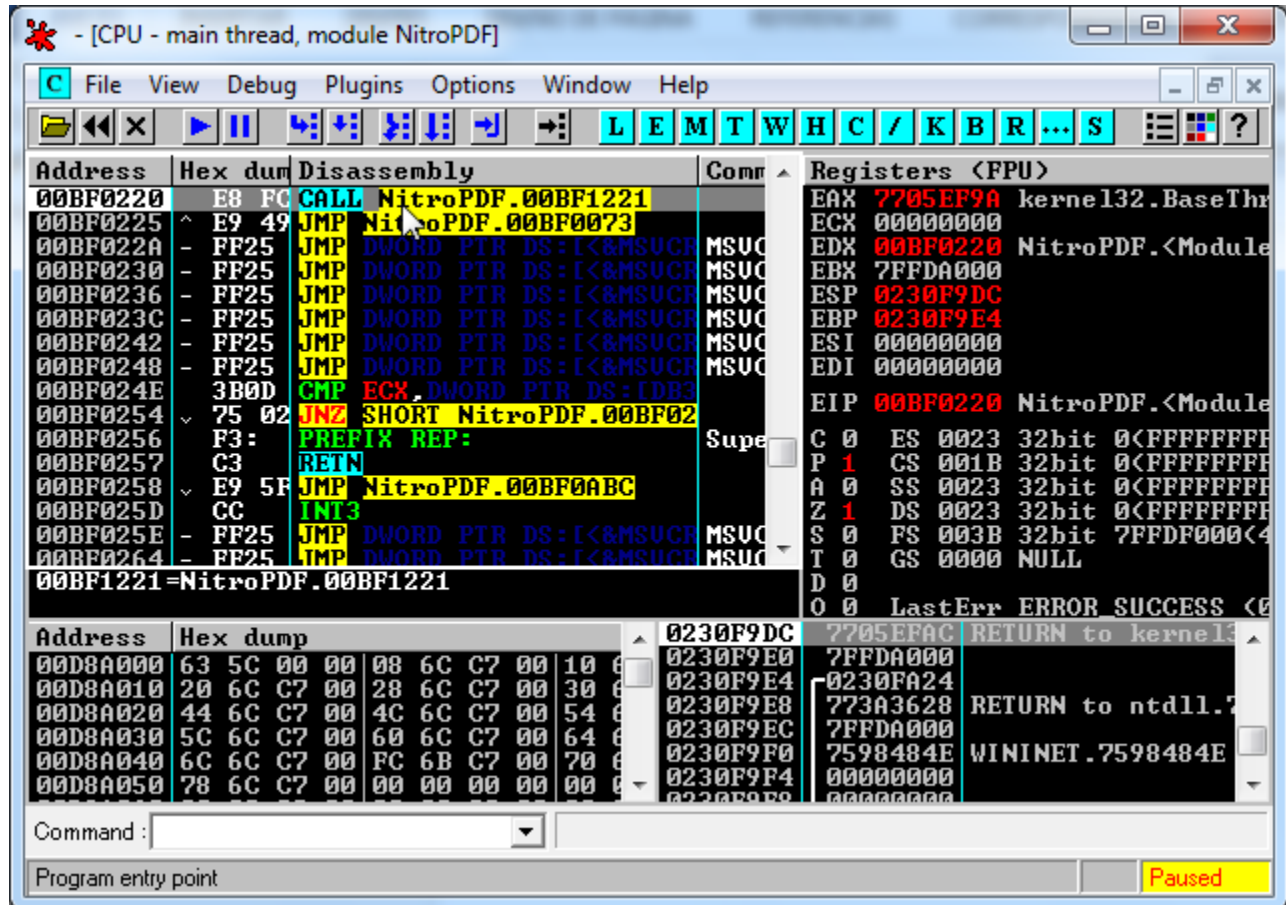
de un buen chico, ayuda a comprimir el ejecutable pero no interfiere para la tarea de crackearlo, y vemos que también tenemos la **API_IsDebuggerPresent**, pero este <<OllyDBG v1.10>> que armé mientras hacía (mejor dicho, que sigo haciendo) el curso **OLLY DESDE CERO** ya viene protegido contra esta API.



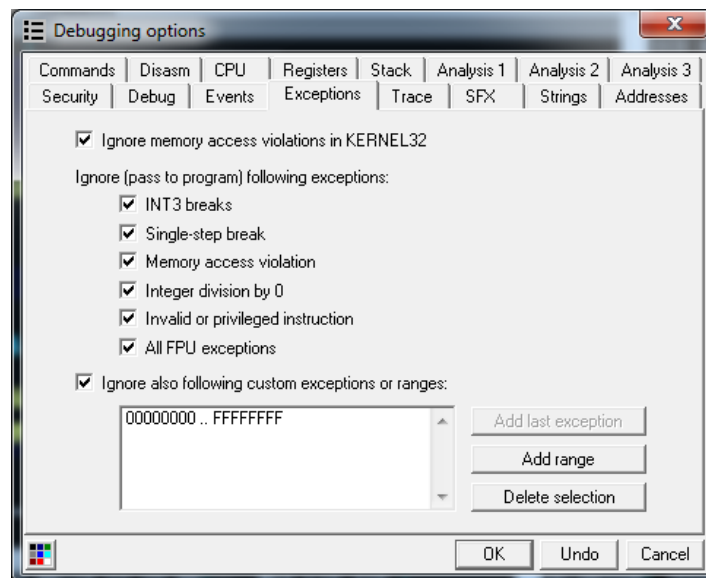
Ya vimos lo necesario, y como conclusión de nuestro análisis y debido a nuestro nivel de Cracking, lo parchearemos. Ya solo nos queda enfrentarnos con él.

AL ATAQUE

Lo cargamos en nuestro Olly. La imagen está muy reducida pero solo quiero mostrar que cargó y llegó a su <ENTRY POINT>.

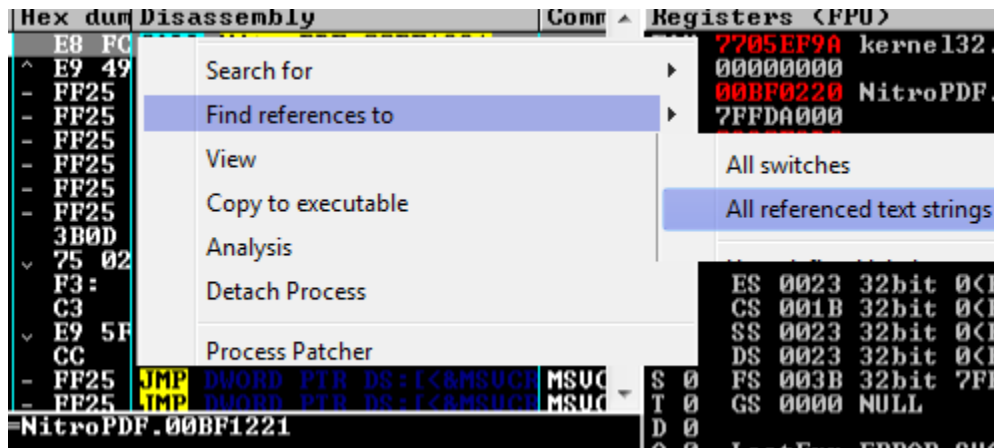


Se me olvidaba, creo que siempre lo doy por obvio pero hoy lo diré en este tuto. Marcamos todas las excepciones, <Options->Debugging options->Exceptions>.

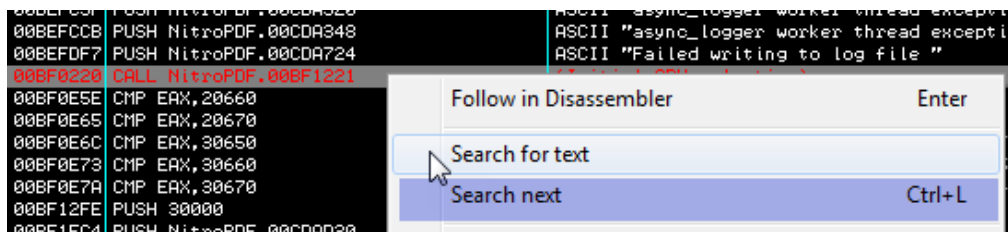


[Tuto007 - Nitro Pro v12.1.0.195 (Crack.Patch)(OllyDBG v1.10)]

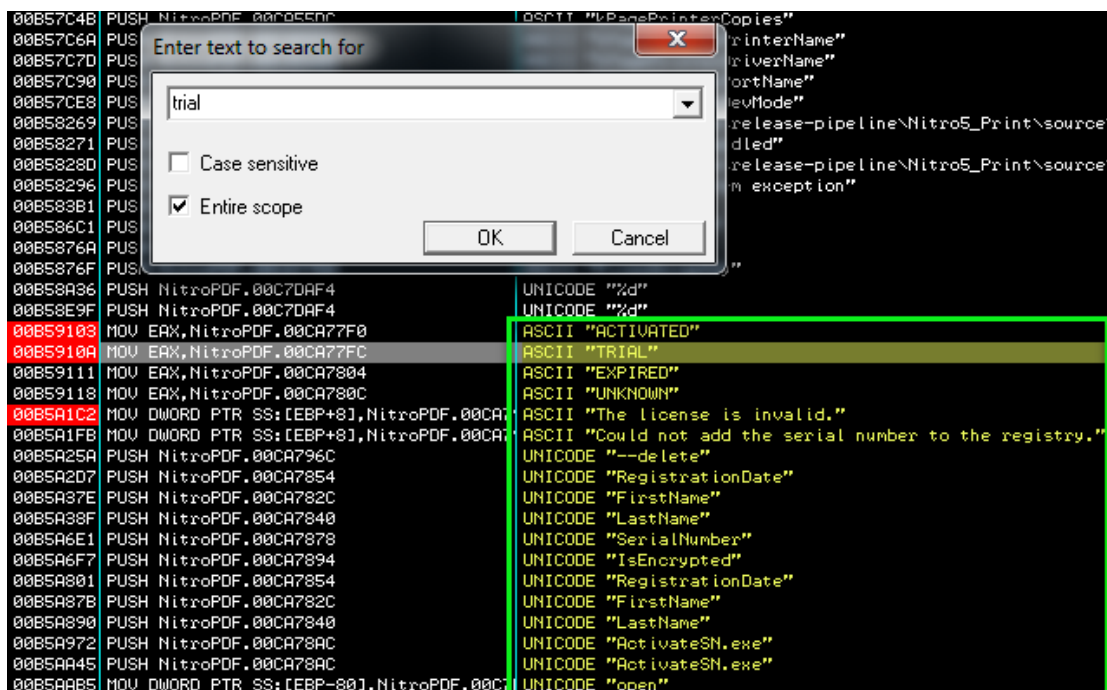
Listo, Yo lo primero que decidí fue hacer una búsqueda de las **strings**. <Click Derecho->Fine references to->All referenced text strings>.



Cosas que pueden ser obvias pero que son buenas contar, que seguro le sirven a uno para saber por dónde buscar. Tenemos el programa en español pero si vemos las **strings** todas están en inglés, entonces debemos realizar nuestra búsqueda en inglés.




Para realizar nuestra búsqueda de **strings**, con <Click Derecho->Search for text> y colocamos lo que queremos buscar con la opción <Entire scope> para realizar una búsqueda completa. Son muchas **strings** pero lo lógico sería buscar el "MODULO PRUEBA" pero en su versión de inglés y de inmediato viene a nuestra cabeza la palabra "TRIAL", además sabemos que ese es el término más usado, hasta por nosotros.



[Tuto007 - Nitro Pro v12.1.0.195 (Crack.Patch)(OllyDBG v1.10)]

Encontramos unas cuantas, y aquí empieza mi "picando aquí, picando allá", porque son hartas, pero no muchísimas como para volver esto imposible. Voy recorriendo una por una y a la que me parecía prometedora le colocaba su **<BREAKPOINT>** (BP). En la imagen de arriba nuestro donde encuentro una de esas **strings** y les coloqué sus **<BREAKPOINTS>**. Aquí es donde pones a prueba ese instinto y ojo de cracker que desarrollas a medida que practicas reversing. También realicé una búsqueda con la string "License" y a la que yo le veía buena pinta entonces ta' que su **<BREAKPOINT>**. Y así fue como terminé según mi **INTRODUCCIÓN**, picando y lleno de BP's.

Address	Module	Active	Disassembly	Comment
00F67F50	NitroPDF	Always	PUSH EBP	Aquiempiezatodo
00F95186	NitroPDF	Always	PUSH NitroPDF.011E0DEC	
00F95764	NitroPDF	Always	PUSH NitroPDF.011E09E4	
00F97050	NitroPDF	Always	PUSH EBP	
00F97242	NitroPDF	Always	JNZ SHORT NitroPDF.00F97256	callstack
00F9720B	NitroPDF	Always	PUSH NitroPDF.011E1D7C	
00FD5BF0	NitroPDF	Always	PUSH EBP	
01017A94	NitroPDF	Always	PUSH NitroPDF.011F58A8	
01017ABC	NitroPDF	Always	PUSH NitroPDF.011F58B8	muestra nag de info
010385C2	NitroPDF	Disabled	CALL NitroPDF.010836A0	
010385C7	NitroPDF	Always	PUSH 1	lleva a licencia perpetua
0103EF9F	NitroPDF	Always	JNZ SHORT NitroPDF.0103EFAA	
0103EFA4	NitroPDF	Always	JE NitroPDF.0103F19D	
0103EFAA	NitroPDF	Always	PUSH NitroPDF.011F7B5C	
0103F22A	NitroPDF	Always	JE SHORT NitroPDF.0103F23D	
0104021F	NitroPDF	Always	JNZ SHORT NitroPDF.0104027B	
01040257	NitroPDF	Always	CALL <JMP.&nfc120u.#4049>	
0108027B	NitroPDF	Always	PUSH NitroPDF.01203108	
010B90B0	NitroPDF	Always	PUSH ESI	
010B90DB	NitroPDF	Always	JNZ SHORT NitroPDF.010B90DF	
010B90F0	NitroPDF	Always	PUSH EBP	interesante1 interesante1
010B9103	NitroPDF	Always	MOV EAX,NitroPDF.012077F0	
010B910A	NitroPDF	Always	MOV EAX,NitroPDF.012077FC	
010BA1C2	NitroPDF	Always	MOV DWORD PTR SS:[EBP+8],Nitro	
010B8550	NitroPDF	Always	MOV EAX,DWORD PTR DS:[ECX+1C]	
010B8A15	NitroPDF	Always	PUSH NitroPDF.01207878	
010B8A5C	NitroPDF	Always	RETN	
010C969E	NitroPDF	Always	PUSH NitroPDF.01209368	
010C97FC	NitroPDF	Always	RETN	
010C9B3B	NitroPDF	Always	PUSH NitroPDF.012099B8	
010C9B68	NitroPDF	Always	PUSH NitroPDF.01209A20	
010C9E6A	NitroPDF	Always	PUSH NitroPDF.0120A388	
5C5B8463	metrics	Always	PUSH metrics.5C5DBBFC	
5C5B8535	metrics	Always	RETN 0C	
5C5B853D	metrics	Always	RETN	

Y ahí estaba yo, con el ojo lleno de **<BREAKPOINTS>** y como la idea de esto no es solo decirte, "mira ve a la dirección tal y cambias esto por esto, guarda los cambios y listo", sino tratar de compartir la experiencia y lo poquito que se va aprendiendo. Así que solo nos queda es ejecutar el programa con **<F9>** o , y tracear y revisar con detenimiento las veces que sea necesario hasta ir reduciendo nuestra búsqueda y llegar a la "ZONA CALIENTE". Como vemos en la imagen anterior a medida que iba avanzando agregaba comentarios para mejor comprensión.

[Tuto007 - Nitro Pro v12.1.0.195 (Crack.Patch)(OllyDBG v1.10)]

Address	Hex dump	Disassembly	Comment
010B90B0	56	PUSH ESI	interesante1
010B90B1	E8 3AFEFFFF	CALL NitroPDF.010B8EF0	
010B90B6	8BF0	MOV ESI,EAX	
010B90B8	8B0E	MOV ECX,DWORD PTR DS:[ESI]	
010B90BA	8B11	MOV EDX,DWORD PTR DS:[ECX]	
010B90BC	8B52 04	MOV EDX,DWORD PTR DS:[EDX+4]	
010B90BF	FFD2	CALL EDX	Valida FileChck.dll
010B90C1	84C0	TEST AL,AL	
010B90C3	74 1A	JE SHORT NitroPDF.010B90DF	
010B90C5	8B0E	MOV ECX,DWORD PTR DS:[ESI]	
010B90C7	8B01	MOV EAX,DWORD PTR DS:[ECX]	
010B90C9	FF50 08	CALL DWORD PTR DS:[EAX+8]	
010B90CC	85C0	TEST EAX,EAX	
010B90CE	75 0F	JNZ SHORT NitroPDF.010B90DF	
010B90D0	8B0E	MOV ECX,DWORD PTR DS:[ESI]	
010B90D2	8B01	MOV EAX,DWORD PTR DS:[ECX]	
010B90D4	8B40 18	MOV EAX,DWORD PTR DS:[EAX+18]	
010B90D7	FFD0	CALL EAX	
010B90D9	84C0	TEST AL,AL	
010B90DB	75 02	JNZ SHORT NitroPDF.010B90DF	interesante1
010B90DD	5E	POP ESI	
010B90DE	C3	RETN	
010B90DF	B0 01	MOV AL,1	
010B90E1	5E	POP ESI	
010B90E2	C3	RETN	


Ahí paramos en el primero de nuestros <BREAKPOINTS> y podemos ver que hace unos **TEST AL,AL** y un **TEST EAX,EAX** y que según el resultado (según el FLAG-Z), si **AL=0** y **EAX=0** o no, iría a **NitroPDF.010B90DF**. También tenemos un **CALL** interesante en la dirección **010B90BF CALL EDX**. Sea esta la oportunidad para hacer notar esto, y como vemos ese **CALL EDX** irá siempre a una misma dirección pero no lo sabremos hasta que no lleguemos ahí traceando. Lleguemos ahí.

010B90B0	56	PUSH ESI	interesante1
010B90B1	E8 3AFEFFFF	CALL NitroPDF.010B8EF0	
010B90B6	8BF0	MOV ESI,EAX	ESI DA LA POSICIÓN PARA LOS CALL
010B90B8	8B0E	MOV ECX,DWORD PTR DS:[ESI]	
010B90BA	8B11	MOV EDX,DWORD PTR DS:[ECX]	
010B90BC	8B52 04	MOV EDX,DWORD PTR DS:[EDX+4]	
010B90BF	FFD2	CALL EDX	Valida FileChck.dll
010B90C1	84C0	TEST AL,AL	
010B90C3	74 1A	JE SHORT NitroPDF.010B90DF	
010B90C5	8B0E	MOV ECX,DWORD PTR DS:[ESI]	
010B90C7	8B01	MOV EAX,DWORD PTR DS:[ECX]	
010B90C9	FF50 08	CALL DWORD PTR DS:[EAX+8]	
010B90CC	85C0	TEST EAX,EAX	
010B90CE	75 0F	JNZ SHORT NitroPDF.010B90DF	
010B90D0	8B0E	MOV ECX,DWORD PTR DS:[ESI]	
010B90D2	8B01	MOV EAX,DWORD PTR DS:[ECX]	
010B90D4	8B40 18	MOV EAX,DWORD PTR DS:[EAX+18]	
010B90D7	FFD0	CALL EAX	
010B90D9	84C0	TEST AL,AL	
010B90DB	75 02	JNZ SHORT NitroPDF.010B90DF	
010B90DD	5E	POP ESI	
010B90DE	C3	RETN	
010B90DF	B0 01	MOV AL,1	
010B90E1	5E	POP ESI	
010B90E2	C3	RETN	


EDX=010BB640 (NitroPDF.010BB640)

Con la imagen de arriba podemos analizar para entender un poco la lógica de las cosas. Como podemos ver no es solo uno si no varios **CALL** que todavía no se sabe a dónde saltarán, pero saltarán teniendo como base el valor retornado por **010B90B1 CALL NitroPDF.010B8EF0** en **EAX** que luego será movido a **ESI** con **MOV ESI,EAX**. Si miramos estamos detenidos en **010B90BF CALL EDX** y ahora si sabemos a dónde saltará y será **EDX=010BB640 (NitroPDF.010BB640)**. Supongo que es hecho así por parte del programador para maximizar el código o para ocultar esas rutinas de traceos indebidas; vaya uno a saber por qué funciona así, yo de programador no tengo nada. Todo lo anterior lo explico porque de esta misma forma accede a la "ZONA DESCICIVA"


[Tuto007 - Nitro Pro v12.1.0.195 (Crack.Patch)(OllyDBG v1.10)]

en donde se lee un valor guardado previamente, y si es el valor es el indicado tomará siempre el camino correcto. Sigamos con <F9> o .

Address	Hex dump	Disassembly	Comment
010C969E	68 68932001	PUSH NitroPDF.01209368	ASCII "CheckLicense"
010C96A3	50	PUSH EAX	
010C96A4	FF15 8C421D00	CALL DWORD PTR DS:[<&KERNEL32	kernel32.GetProcAddress
010C96AA	85C0	TEST EAX,EAX	
010C96AC	75 3F	JNZ SHORT NitroPDF.010C96ED	
010C96AE	FF37	PUSH DWORD PTR DS:[EDI]	
010C96B0	FF15 B4411D00	CALL DWORD PTR DS:[<&KERNEL32	kernel32.FreeLibrary
010C96B6	57	PUSH EDI	
010C96B7	C707 00000000	MOV DWORD PTR DS:[EDI],0	
010C96BD	E8 686B0800	CALL <JMP.&MSUCR120.??3EYAXPAX	
010C96C2	83C4 04	ADD ESP,4	
010C96C5	6A 10	PUSH 10	
010C96C7	68 1C7A2001	PUSH NitroPDF.01207A1C	UNICODE "ERROR"
010C96CC	68 78932001	PUSH NitroPDF.01209378	UNICODE "0000 Filechck.dll is missing or the wrong version."
010C96D1	6A 00	PUSH 0	
010C96D3	FF15 704A1D00	CALL DWORD PTR DS:[<&USER32	USER32.MessageBoxW
010C96D9	5E	POP ESI	
010C96DB	5E	POP EDI	


Más claro no canta un gallo, va revisar nuestra licencia y en esta misma rutina realiza una comprobación de si tenemos la .DLL Filechck.dll. Estas son zonas de mucho interés si quisiéramos comprender cómo funciona el rollo de la **Licencia** para poder crear una (**Licencia**) a nuestro gusto y hacer un KeyGen. Sigamos con <F9> o .

Address	Hex dump	Disassembly	Comment
010BBA15	68 78782001	PUSH NitroPDF.01207878	UNICODE "SerialNumber"
010BBA1A	8D8D 34FEFF	LEA ECX,DWORD PTR SS:[EBP-1CC]	
010BBA20	E8 AB59EDFF	CALL NitroPDF.00F913D0	
010BBA25	6A FF	PUSH -1	
010BBA27	8D4E 0C	LEA ECX,DWORD PTR DS:[ESI+C]	
010BBA2A	FF15 3C4D1D00	CALL DWORD PTR DS:[<&mfc120u.1	mfc120u.571D0211
010BBA30	8B85 34FEFF	MOV EAX,DWORD PTR SS:[EBP-1CC]	
010BBA36	85C0	TEST EAX,EAX	
010BBA38	74 07	JE SHORT NitroPDF.010BBA41	
010BBA3A	50	PUSH EAX	
010BBA3B	FF15 80101D00	CALL DWORD PTR DS:[<&ADUAPI32	ADUAPI32.RegCloseKey
010BBA41	B0 01	MOV AL,1	
010BBA43	8B4D F4	MOV ECX,DWORD PTR SS:[EBP-C]	
010BBA46	64:890D 0000	MOV DWORD PTR FS:[0],ECX	
010BBA4D	59	POP ECX	
010BBA4E	5E	POP ESI	
010BBA4F	8B4D F0	MOV ECX,DWORD PTR SS:[EBP-10]	
010BBA52	33CD	XOR ECX,EBP	
010BBA54	E8 F5470900	CALL NitroPDF.0115024E	
010BBA59	8BE5	MOV ESP,EBP	
010BBA5B	5D	POP EBP	
010BBA5C	C3	RETN	


Lo mismo que la anterior. Sería de utilidad si quisiéramos hallar nuestra **Licencia** y nosotros aquí estamos es tirando directamente, y a la yugular. Sigamos como veníamos <F9> o , hasta llegar a la imagen de abajo.

[Tuto007 - Nitro Pro v12.1.0.195 (Crack.Patch)(OllyDBG v1.10)]

010B90B0	56	PUSH ESI	interesante1
010B90B1	E8 3AFEFFFF	CALL NitroPDF.010B8EF0	
010B90B6	8BF0	MOV ESI,EAX	
010B90B8	8B0E	MOV ECX,DWORD PTR DS:[ESI]	
010B90BA	8B11	MOV EDX,DWORD PTR DS:[ECX]	
010B90BC	8B52 04	MOV EDX,DWORD PTR DS:[EDX+4]	
010B90BF	FFD2	CALL EDX	Valida FileChck.dll
010B90C1	84C0	TEST AL,AL	
010B90C3	74 1A	JE SHORT NitroPDF.010B90DF	
010B90C5	8B0E	MOV ECX,DWORD PTR DS:[ESI]	
010B90C7	8B01	MOV EAX,DWORD PTR DS:[ECX]	
010B90C9	FF50 08	CALL DWORD PTR DS:[EAX+8]	
010B90CC	85C0	TEST EAX,EAX	
010B90CE	75 0F	JNZ SHORT NitroPDF.010B90DF	
010B90D0	8B0E	MOV ECX,DWORD PTR DS:[ESI]	
010B90D2	8B01	MOV EAX,DWORD PTR DS:[ECX]	
010B90D4	8B40 18	MOV EAX,DWORD PTR DS:[EAX+18]	
010B90D7	FFD0	CALL EAX	
010B90D9	84C0	TEST AL,AL	
010B90DB	75 02	JNZ SHORT NitroPDF.010B90DF	interesante1
010B90DD	5E	POP ESI	
010B90DE	C3	RETN	


Pues podemos notar que volvimos a la rutina inicial, por eso después de muchísimos traceos tras traceos les puse comentarios como interesante. Efectivamente es interesante si quisiéramos crear un KeyGen. Y yo aquí le cambiaba los saltos del FLAG-Z y los seguía y me iba armando el camino correcto. Sigamos con <F9> o .

001F7F50	55	PUSH EBP	
001F7F51	8BEC	MOV EBP,ESP	
001F7F53	51	PUSH ECX	
001F7F54	8B49 04	MOV ECX,DWORD PTR DS:[ECX+4]	
001F7F57	56	PUSH ESI	
001F7F58	C745 FC 0000	MOV DWORD PTR SS:[EBP-4],0	
001F7F5F	8B01	MOV EAX,DWORD PTR DS:[ECX]	
001F7F61	FF50 0C	CALL DWORD PTR DS:[EAX+C]	
001F7F64	50	PUSH EAX	
001F7F65	E8 86111500	CALL NitroPDF.003490F0	
001F7F6A	8B75 08	MOV ESI,DWORD PTR SS:[EBP+8]	
001F7F6D	8BD0	MOV EDX,EAX	
001F7F6F	83C4 04	ADD ESP,4	
001F7F72	C746 14 0F00	MOV DWORD PTR DS:[ESI+14],0F	
001F7F79	C746 10 0000	MOV DWORD PTR DS:[ESI+10],0	
001F7F80	C606 00	MOV BYTE PTR DS:[ESI],0	
001F7F83	803A 00	CMP BYTE PTR DS:[EDX],0	
001F7F86	75 14	JNZ SHORT NitroPDF.001F7F9C	
001F7F88	33C9	XOR ECX,ECX	
001F7F8A	51	PUSH ECX	
001F7F8B	52	PUSH EDX	
001F7F8C	8BCE	MOV ECX,ESI	
001F7F8E	E8 BDCCFFFF	CALL NitroPDF.001F4C50	
001F7F93	8BC6	MOV EAX,ESI	
001F7F95	5E	POP ESI	
001F7F96	8BE5	MOV ESP,EBP	


Este <BREAKPOINT> en 001F7F50, debo decirlo, lo ubiqué mientras traceaba y gracias a este halle la "ZONA DESCICIVA" que está en ese 001F7F61 CALL y como lo explicamos al inicio del traceo viene dado en este caso por el valor en EAX. Si siguiéramos traceando hasta llegar a ese CALL y entráramos tendríamos la rutina a parchear, pero lo haremos después porque nos falta más análisis porque debo explicar cómo ubiqué ese lugar. Sigamos otra vez con <F9> o , porque esto ya se puso bueno.

[Tuto007 - Nitro Pro v12.1.0.195 (Crack.Patch)(OllyDBG v1.10)]


Address	Hex dump	Disassembly	Comment	Registers (FPU)
003490F0	55	PUSH EBP		EAX 00000002
003490F1	8BEC	MOV EBP, ESP		ECX 049BE2E0
003490F3	8B45 08	MOV EAX, DWORD PTR SS:[EBP+8]		EDX 00467100
003490F6	48	DEC EAX		EBX 01A2D4A0
003490F7	83F8 03	CMP EAX, 3		ESP 01A2D3F0
003490FA	77 1C	JA SHORT NitroPDF.00349118		EBP 01A2D400
003490FC	FF2485 20913	JMP DWORD PTR DS:[EAX*4+349120]		ESI 01A2D438
00349103	B8 F0774900	MOV EAX, NitroPDF.004977F0	ASCII "ACTIVATED"	EDI 01A2D4A8
00349108	5D	POP EBP		EIP 003490F0
00349109	C3	RETN		
0034910A	B8 FC774900	MOV EAX, NitroPDF.004977FC	ASCII "TRIAL"	C 0 ES 0023
0034910F	5D	POP EBP		P 1 CS 001B
00349110	C3	RETN		A 0 SS 0023
00349111	B8 04784900	MOV EAX, NitroPDF.00497804	ASCII "EXPIRED"	Z 0 DS 0023
00349116	5D	POP EBP		S 0 FS 003B
00349117	C3	RETN		T 0 GS 0000
00349118	B8 0C784900	MOV EAX, NitroPDF.0049780C	ASCII "UNKNOWN"	D 0
0034911D	5D	POP EBP		O 0 LastErr
0034911E	C3	RETN		EFL 00200206

Y esta es la rutina que me mostró el camino y si recuerdan estos son los **<BREAKPOINTS>** que colocamos cuando buscamos las **strings**. Por aquí pasé muchas veces, pasaba y pasaba y no lo pillaba, pero ya lo dice el dicho: "tanto va el cántaro al agua, que por fin se rompe". Traciamos con <F7> o , hasta llegar a 003490F3.

003490EF	CC	INT3	Registers (FPU)
003490F0	55	PUSH EBP	EAX 00000002
003490F1	8BEC	MOV EBP, ESP	ECX 049BE2E0
003490F3	8B45 08	MOV EAX, DWORD PTR SS:[EBP+8]	EDX 00467100 NitroPDF.00467100
003490F6	48	DEC EAX	EBX 01A2D4A0
003490F7	83F8 03	CMP EAX, 3	ESP 01A2D3EC
003490FA	77 1C	JA SHORT NitroPDF.00349118	EBP 01A2D3EC
003490FC	FF2485 20913	JMP DWORD PTR DS:[EAX*4+349120]	ESI 01A2D438
003490FD	B8 F0774900	MOV EAX, NitroPDF.004977F0	EDI 01A2D4A8 ASCII "ES"
Stack SS:[01A2D3F4]=00000002			EIP 003490F3 NitroPDF.003490F3
EAX=00000002			

Vemos en los registros que **EAX=00000002**, también podemos ver el valor de **EAX** en las observaciones del Olly. Estamos parados en 003490F3 y moveremos el valor que se encuentra en **[EBP+8]** a **EAX**, y otra vez las observaciones no dan más información y **[01A2D3F4]=00000002**. Entonces no cambia nada, siempre será el mismo valor. Pasemos a la siguiente dirección 003490F6 con <F7> o .

003490F0	55	PUSH EBP
003490F1	8BEC	MOV EBP, ESP
003490F3	8B45 08	MOV EAX, DWORD PTR SS:[EBP+8]
003490F6	48	DEC EAX
003490F7	83F8 03	CMP EAX, 3
003490FA	77 1C	JA SHORT NitroPDF.00349118
003490FC	FF2485 20913	JMP DWORD PTR DS:[EAX*4+349120]
003490FD	B8 F0774900	MOV EAX, NitroPDF.004977F0
EAX=00000002		

En 003490F6 le restaremos **0x1** a **EAX** con **DEC EAX**. Como **EAX=00000002**, entonces cuando pasemos esta instrucción **EAX=00000001**. Sigamos con <F7> o  hasta la siguiente instrucción.

[Tuto007 - Nitro Pro v12.1.0.195 (Crack.Patch)(OllyDBG v1.10)]

003490F0	55	PUSH EBP
003490F1	8BEC	MOV EBP,ESP
003490F3	8B45 08	MOV EAX,DWORD PTR SS:[EBP+8]
003490F6	48	DEC EAX
003490F7	83F8 03	CMP EAX,3
003490FA	77 1C	JL SHORT NitroPDF.00349118
003490FC	FF2485 20913	JMP DWORD PTR DS:[EAX*4+349120]
00349100		
EAX=00000001		

En 003490F7 comparará EAX con 0x3 (CMP EAX,3) y abajo tenemos el salto en 003490FA JA. Mientras EAX sea menor o igual a 0x3 el salto no se tomará y pasaremos a 003490FC donde hay un JMP que saltará a una dirección dependiendo del valor de EAX. Lleguemos hasta el JMP.

003490F0	55	PUSH EBP	
003490F1	8BEC	MOV EBP,ESP	
003490F3	8B45 08	MOV EAX,DWORD PTR SS:[EBP+8]	
003490F6	48	DEC EAX	
003490F7	83F8 03	CMP EAX,3	
003490FA	77 1C	JL SHORT NitroPDF.00349118	
003490FC	FF2485 20913	JMP DWORD PTR DS:[EAX*4+349120]	NitroPDF.0034910A
00349103	B8 F0774900	MOV EAX,NitroPDF.004977F0	ASCII "ACTIVATED"
00349108	5D	POP EBP	
00349109	C3	RETN	
0034910A	B8 FC774900	MOV EAX,NitroPDF.004977FC	ASCII "TRIAL"
0034910F	5D	POP EBP	
00349110	C3	RETN	
00349111	B8 04784900	MOV EAX,NitroPDF.00497804	ASCII "EXPIRED"
00349116	5D	POP EBP	
00349117	C3	RETN	
00349118	B8 0C784900	MOV EAX,NitroPDF.0049780C	ASCII "UNKNOWN"
0034911D	5D	POP EBP	
0034911E	C3	RETN	
0034911F	90	NOP	

El salto nos envía a 0034910A y es al "TRIAL", entonces sabemos que EAX empezó con un valor de 0x2 y al restarle 0x1 quedó EAX=00000001; luego para saltar a "ACTIVATED" EAX debe ser igual a 0x0 y para que eso suceda EAX debe valer al inicio de la rutina 0x1 para que al restarle 0x1 nos quedará EAX=0. Entonces vamos entendiendo que cuando tengamos EAX=0x1 iremos por el camino correcto. Sigamos por donde veníamos la ruta del "TRIAL" hasta llegar al RETN.

00F690F7	83F8 03	CMP EAX,3	
00F690FA	77 1C	JL SHORT NitroPDF.00F69118	
00F690FC	FF2485 2091F60	JMP DWORD PTR DS:[EAX*4+F69120]	
00F69103	B8 F0770B01	MOV EAX,NitroPDF.010B77F0	ASCII "ACTIVATED"
00F69108	5D	POP EBP	
00F69109	C3	RETN	
00F6910A	B8 FC770B01	MOV EAX,NitroPDF.010B77FC	ASCII "TRIAL"
00F6910F	5D	POP EBP	
00F69110	C3	RETN	
00F69111	B8 04780B01	MOV EAX,NitroPDF.010B7804	ASCII "EXPIRED"
00F69116	5D	POP EBP	
00F69117	C3	RETN	
00F69118	B8 0C780B01	MOV EAX,NitroPDF.010B780C	ASCII "UNKNOWN"
00F6911D	5D	POP EBP	
00F6911E	C3	RETN	

025AD5CC	00E17F6A	RETURN to NitroPDF.00E17F6A from NitroPDF.00F690F0
025AD5D0	00000002	
025AD5D4	025AD614	
025AD5D8	00000000	
025AD5DC	025AD5F0	
025AD5E0	0F1D839C	
025AD5E4	025AD614	
025AD5E8	00260958	
025AD5EC	00000000	
025AD5F0	025AD658	
025AD5F4	0F1D8451	RETURN to metrics.0F1D8451 from metrics.0F1D8380

[Tuto007 - Nitro Pro v12.1.0.195 (Crack.Patch)(OllyDBG v1.10)]


En el **STACK** podemos ver que retornará a **NitroPDF.00E17F6A**. Vallamos a esa dirección, habiéndola seleccionado en el **STACK** <Clic Derecho->Follow in Disassembler> o <ENTER>.

00E17F50	55	PUSH EBP
00E17F51	8BEC	MOV EBP,ESP
00E17F53	51	PUSH ECX
00E17F54	8B49 04	MOV ECX,DWORD PTR DS:[ECX+4]
00E17F57	56	PUSH ESI
00E17F58	C745 FC 00000000	MOV DWORD PTR SS:[EBP-4],0
00E17F5F	8B01	MOV EAX,DWORD PTR DS:[EAX+1]
00E17F61	FF50 0C	CALL DWORD PTR DS:[EAX+C]
00E17F64	50	PUSH EAX
00E17F65	F8 86111500	CALL NitroPDF.00E690F0
00E17F6A	8B75 08	MOV ESI,DWORD PTR SS:[EBP+8]
00E17F6D	8BD0	MOV EDX,EAX
00E17F6F	83C4 04	ADD ESP,4
00E17F72	C746 14 0F000000	MOV DWORD PTR DS:[ESI+14],0F
00E17F79	C746 10 00000000	MOV DWORD PTR DS:[ESI+10],0
00E17F80	C606 00	MOV BYTE PTR DS:[ESI],0
00E17F83	803A 00	CMP BYTE PTR DS:[EDX],0
00E17F86	75 14	JNZ SHORT NitroPDF.00E17F9C
00E17F88	33C9	XOR ECX,ECX
00E17F8A	51	PUSH ECX

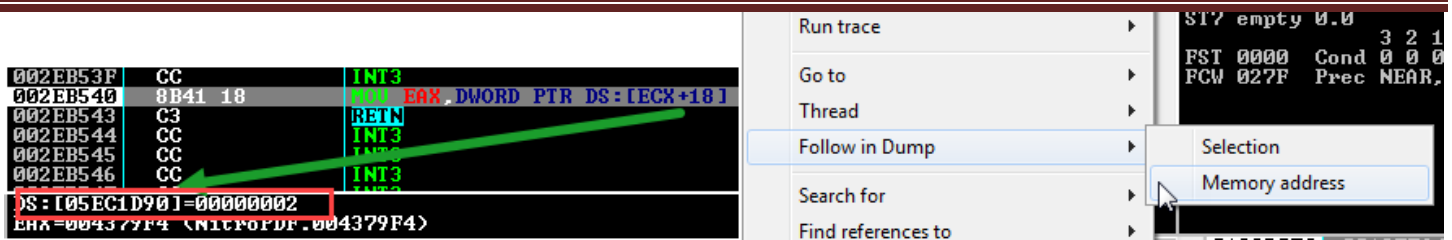
AQUÍ SE CARGA EL VALOR DE EAX

Pues retornamos a **00E17F6A**, y si miramos un poco más arriba podemos ver que es la rutina que ya habíamos visto anteriormente, en donde habíamos dicho que ahí obteníamos la dirección de ese **CALL** donde el programa pasará infinidad de veces para validar si es "FULL". Como vamos ya por mal camino, reiniciemos todo y lleguemos de nuevo a ese <BREAKPOINT> **00E17F50** y luego de ahí traceamos con <F7> hasta quedar en ese **00E17F61 CALL**.

Address	Hex dump	Disassembly	Comment
00197F4E	CC	INT3	
00197F4F	CC	INT3	
00197F50	55	PUSH EBP	
00197F51	8BEC	MOV EBP,ESP	
00197F53	51	PUSH ECX	
00197F54	8B49 04	MOV ECX,DWORD PTR DS:[ECX+4]	
00197F57	56	PUSH ESI	
00197F58	C745 FC 00000000	MOV DWORD PTR SS:[EBP-4],0	
00197F5F	8B01	MOV EAX,DWORD PTR DS:[EAX+1]	
00197F61	FF50 0C	CALL DWORD PTR DS:[EAX+C]	NitroPDF.002EB540
00197F64	50	PUSH EAX	
00197F65	E8 86111500	CALL NitroPDF.002E90F0	
00197F6A	8B75 08	MOV ESI,DWORD PTR SS:[EBP+8]	
00197F6D	8BD0	MOV EDX,EAX	
00197F6F	83C4 04	ADD ESP,4	
00197F72	C746 14 0F000000	MOV DWORD PTR DS:[ESI+14],0F	
00197F79	C746 10 00000000	MOV DWORD PTR DS:[ESI+10],0	
00197F80	C606 00	MOV BYTE PTR DS:[ESI],0	
00197F83	803A 00	CMP BYTE PTR DS:[EDX],0	
00197F86	75 14	JNZ SHORT NitroPDF.00197F9C	
00197F88	33C9	XOR ECX,ECX	

Ya estamos parados en ese **CALL** y como dato curioso las direcciones han cambiado, ahora empiezan con **0019XXXX** y los <BREAKPOINTS> funcionan sin problemas. Supongo que puede ser por el algoritmo <<aPLib compresion>>; bueno, mientras no sea inconveniente, sigamos adelante. Ya sabemos la dirección de ese **CALL** y es a **NitroPDF.002EB540** y que puede cambiar por lo menos la primer parte de la dirección como lo vimos ahora. Vallamos a ese **CALL** con <F7> o .

[Tuto007 - Nitro Pro v12.1.0.195 (Crack.Patch)(OllyDBG v1.10)]



Ahí está el corazón de tener siempre la versión "FULL" (ACTIVATED). Leerá en memoria lo que hay en `[ECX+18]` y en las observaciones vemos que `[05EC1D90]=00000002` que es el valor para la versión "TRIAL". Parados en la instrucción `MOV EAX, DWORD PTR DS:[ECX+18]`, <Clic Derecho->Follow in Dump->Memory address>.

Address	Hex dump
05DAF818	02 00 00 00 01 00 00 00
05DAF828	2A 09 00 00 BF E8 00 00
05DAF838	84 42 77 28 C2 0A 20 0C
05DAF848	98 26 C2 00 30 26 C2 00
05DAF858	74 A8 BF 00 00 CA D4 00

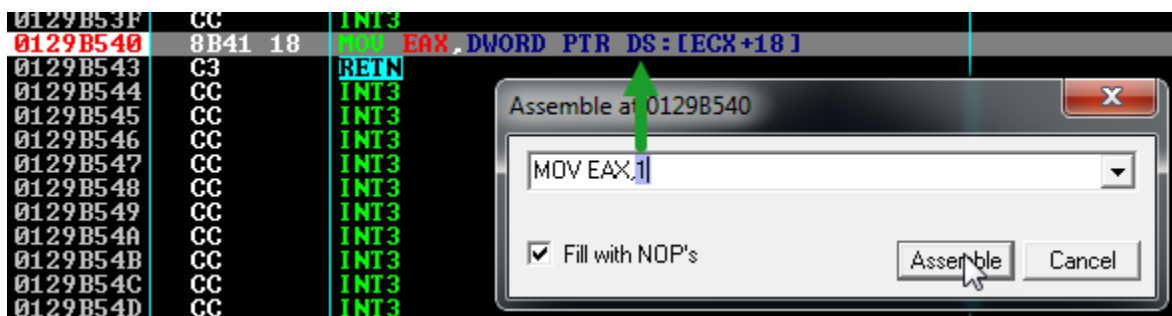
02 --> TRIAL

01 --> FULL

Pueden notar que la dirección en el DUMP no es igual a las observaciones del Olly, y es porque reinicie todo de nuevo. Ya todo ha sido explicado y muchos ya sabrán el próximo paso, y es que a `EAX` siempre se mueva el `0x1`. Entonces parcheamos esa instrucción, <Clic Derecho->Assemble> o <SPACE>



Y solo debemos cambiarlo por `0x1`.

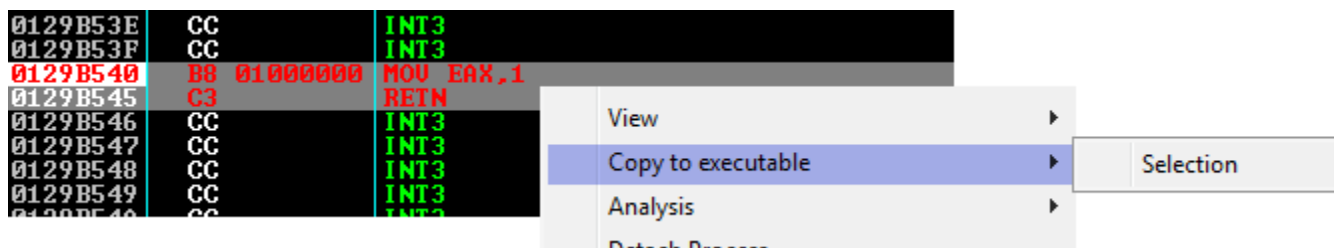


Con ese `MOV EAX,1` siempre tendremos la versión "FULL". Miremos cómo nos quedó.

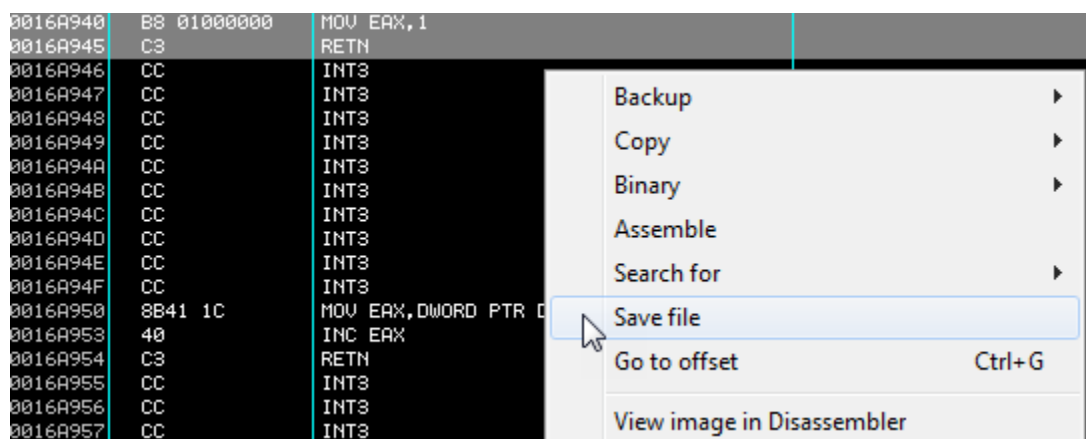
0129B53F	CC	INT3
0129B540	B8 01000000	MOV EAX,1
0129B545	CC	INT3
0129B546	CC	INT3
0129B547	CC	INT3
0129B548	CC	INT3

[Tuto007 - Nitro Pro v12.1.0.195 (Crack.Patch)(OllyDBG v1.10)]

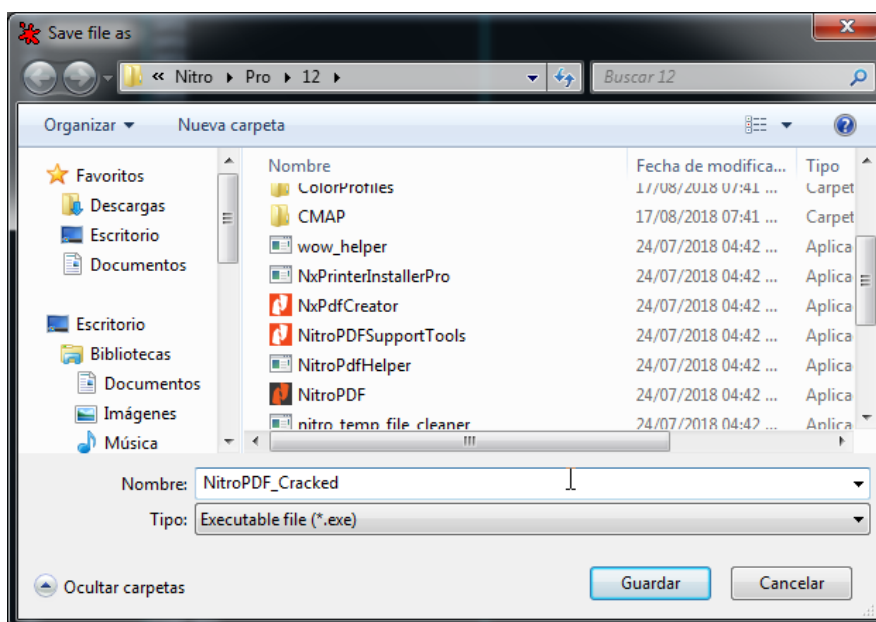
Se nos comió el **RETN** y es porque inicialmente la instrucción ocupaba 3 bytes (8B41 18) y ahora ocupa 5 bytes (B8 01000000). Por fortuna hay espacio de sobra para agregar el **RETN** de nuevo.



Listo, seleccionamos los cambios que hicimos, **<Clic Derecho->Copy to executable->Selection>**.

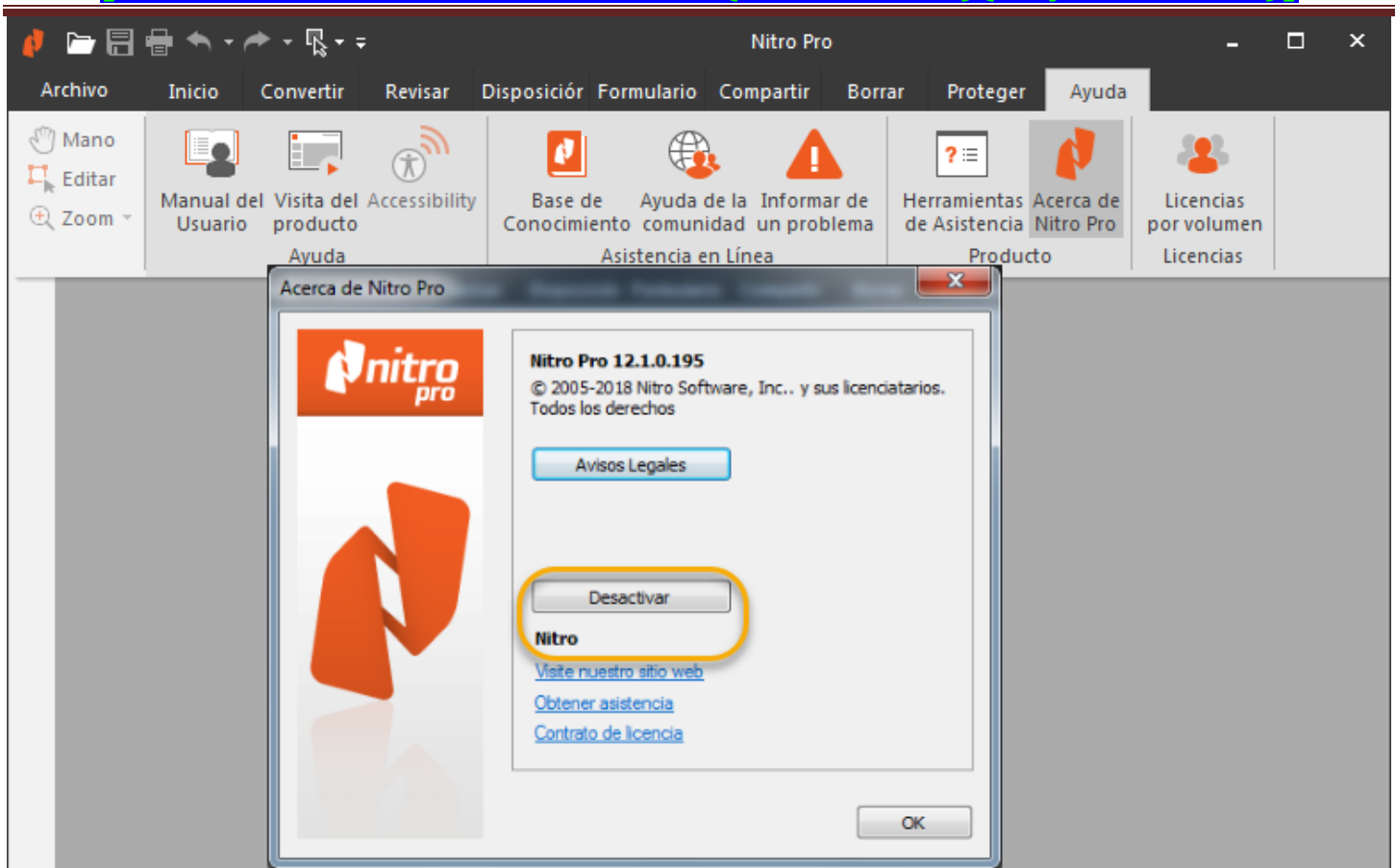


De nuevo, **<Clic Derecho->Save file>** y por fin guardamos nuestra archivo crackeado. Yo lo guarde como **"NitroPDF_Cracked.exe"**.



Solo nos queda probarlo.

[Tuto007 - Nitro Pro v12.1.0.195 (Crack.Patch)(OllyDBG v1.10)]



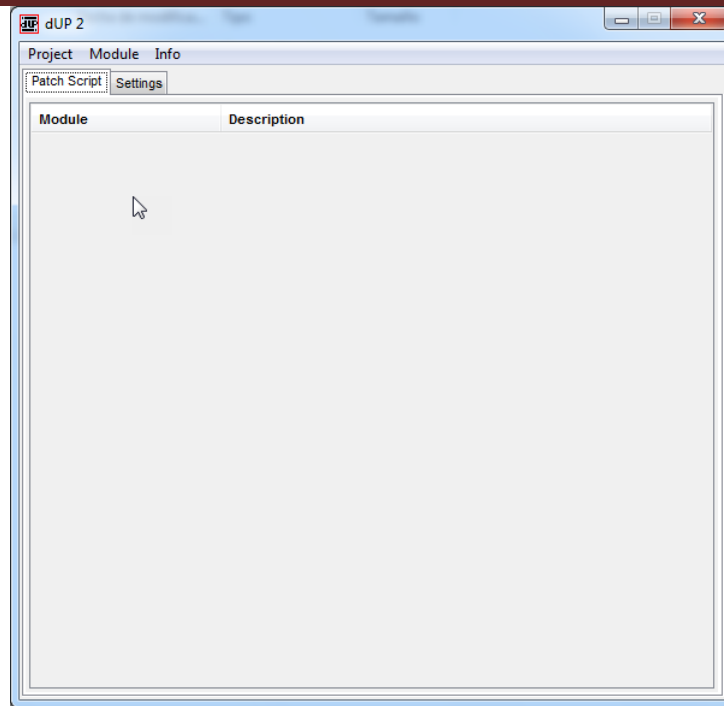
Todas las huellas del "MODO PRUEBA" desaparecieron, hasta el botón que nos decía "Activar" ahora nos dice "Desactivar".

Listo, el ataque ha terminado y la víctima ha caído. Con el Crack es suficiente, pero eso es la mitad de lo propuesto en este tuto, ya que nos queda faltando el Patch, así que con ese tema seguiremos.

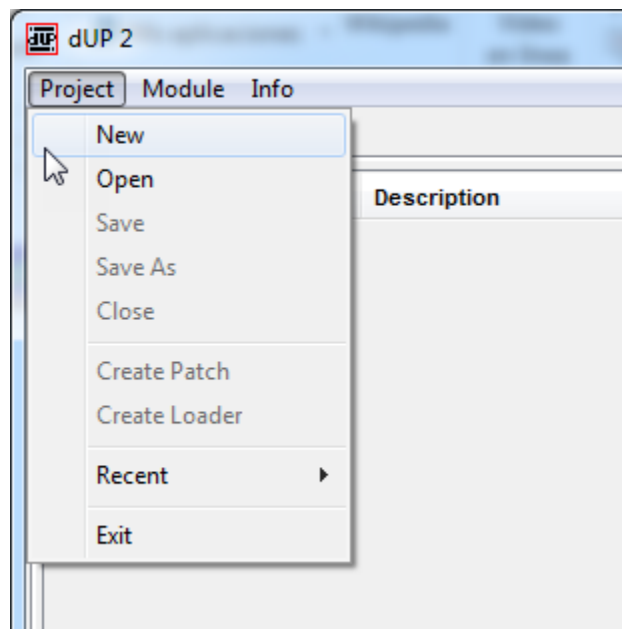
HACIENDO EL PATCH

Les dejo un tuto muy bueno hecho por ShaDDy donde nos da una buena introducción con el dUP2, [\[1009\] dUP2.Diablo's.Universal.Patcher.v2.17.By.ShaDDy](#). Nosotros utilizaremos una versión más reciente pero para el estudio de crear el Patch es muy bueno.

[Tuto007 - Nitro Pro v12.1.0.195 (Crack.Patch)(OllyDBG v1.10)]



Ahí tenemos nuestro <**dUP2 Diablo's Universal Patcher v2.26**> abierto y ahora vamos a crear nuestro proyecto.



Vamos a <**Project->New**> para abrir nuestra ventana de proyecto donde agregaremos información de nuestro **Patch**.

[Tuto007 - Nitro Pro v12.1.0.195 (Crack.Patch)(OllyDBG v1.10)]

Patch Info

Patcher Caption: Patch.Nitro.Pro.v12.xx.por.LUISFECAB

Application: Patch.Nitro.Pro.v12.xx.por.LUISFECAB

Filename (s): NitroPDF.exe

URL: www.ricardonarvaja.info

Author: LUISFECAB

Release Date: August 20, 2018

Release Info: Cracked by LUISFECAB

About Box Message: Cracked by LUISFECAB
CrackSLatinoS
www.ricardonarvaja.info
CLS

Scrolltext:

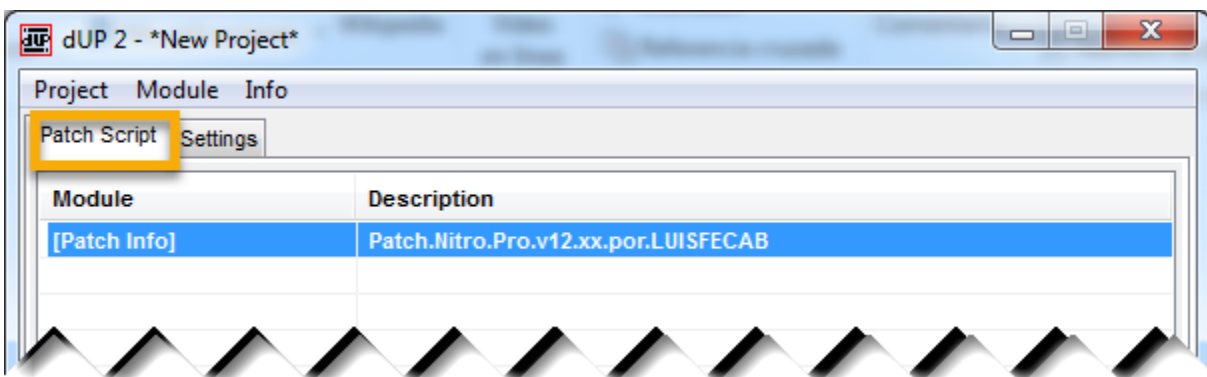
☒ Show this dialog when create a new project

☒ Run patch with administrator rights

☐ No Backup by default

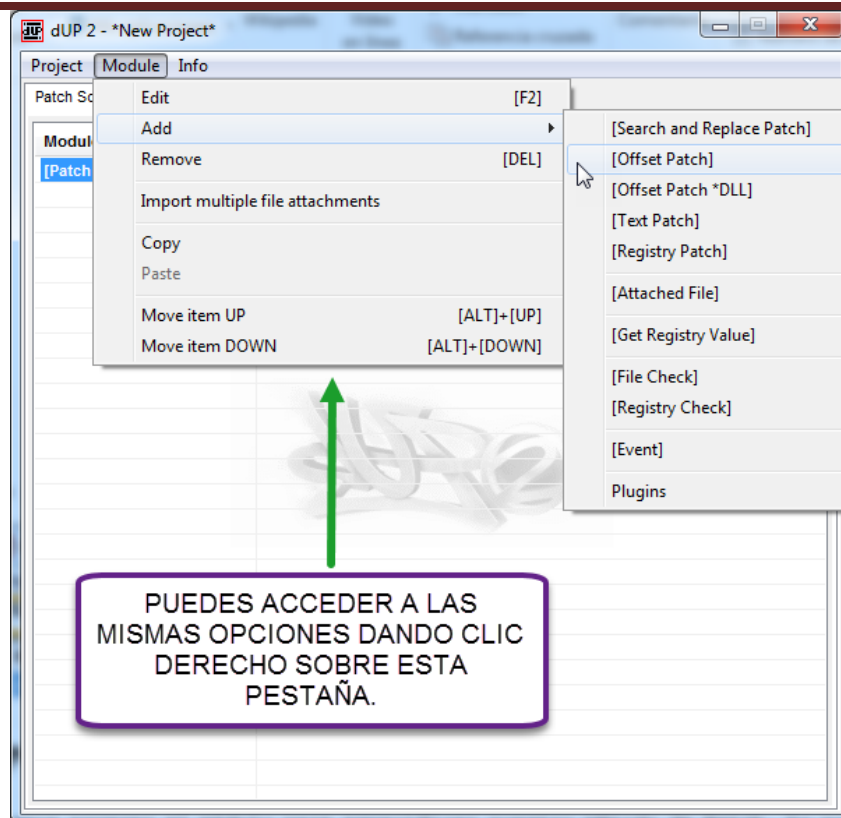
Cancel Save

Guardamos y con eso se nos agrega el primer módulo a la pestaña "**Patch Script**", que es la Info de nuestro **Patch**.

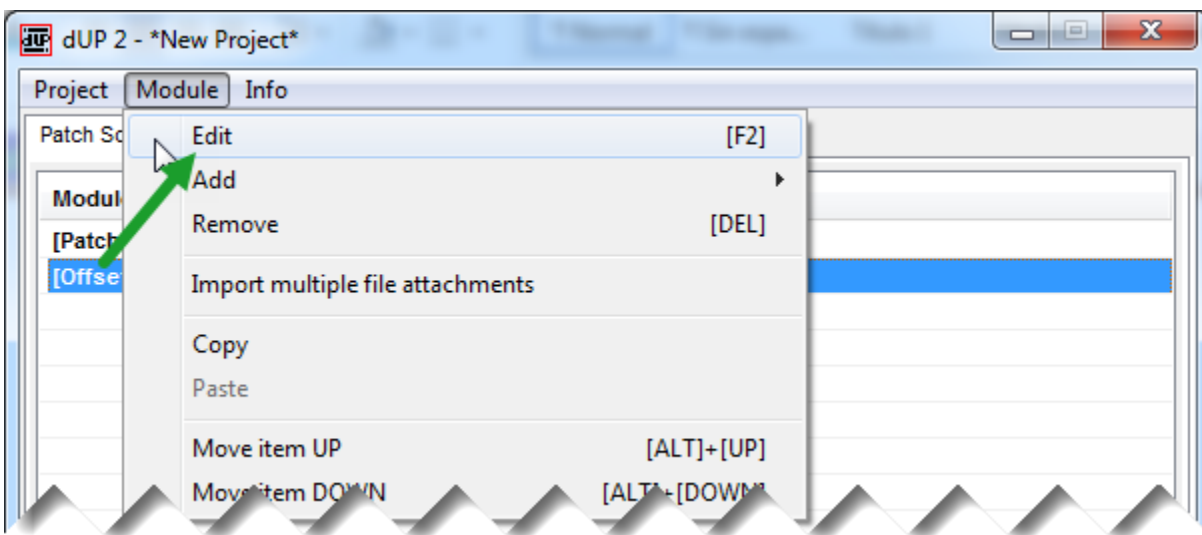


Ahora debemos agregar el módulo para especificar nuestro método de Patch. Lo podemos hacer desde <Module->Add->"Escoger tu forma de Patch">.

[Tuto007 - Nitro Pro v12.1.0.195 (Crack.Patch)(OllyDBG v1.10)]

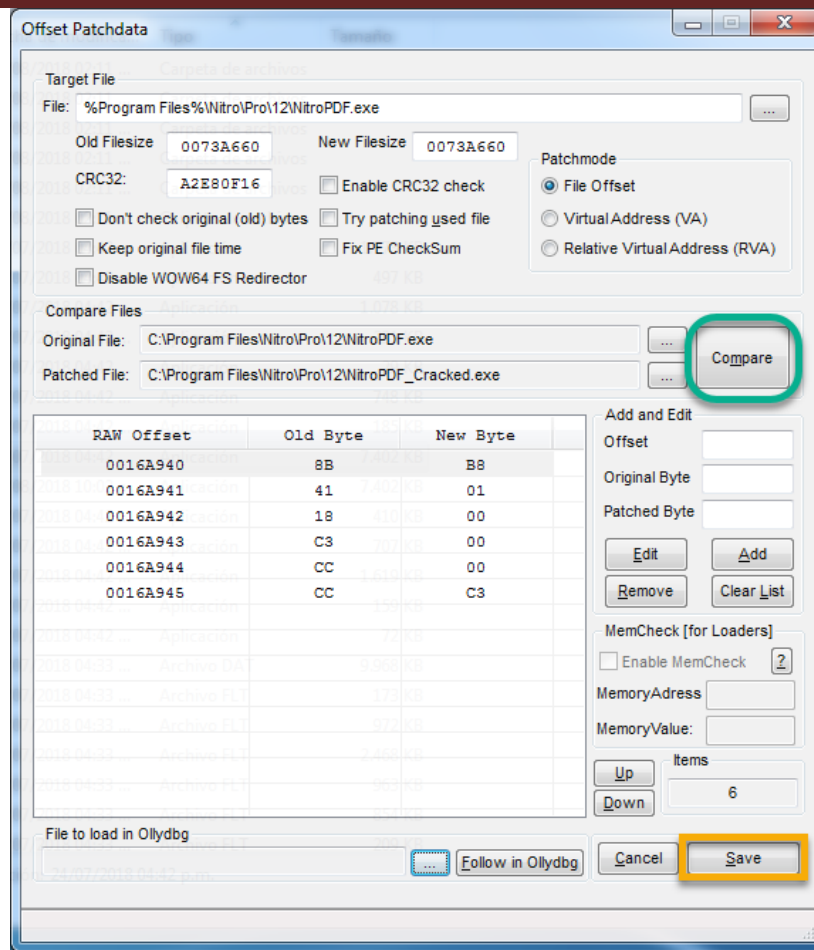


Escogemos la opción "[**Offset Patch**]" para utilizar el método por comparación para hallar los bytes a parchear.

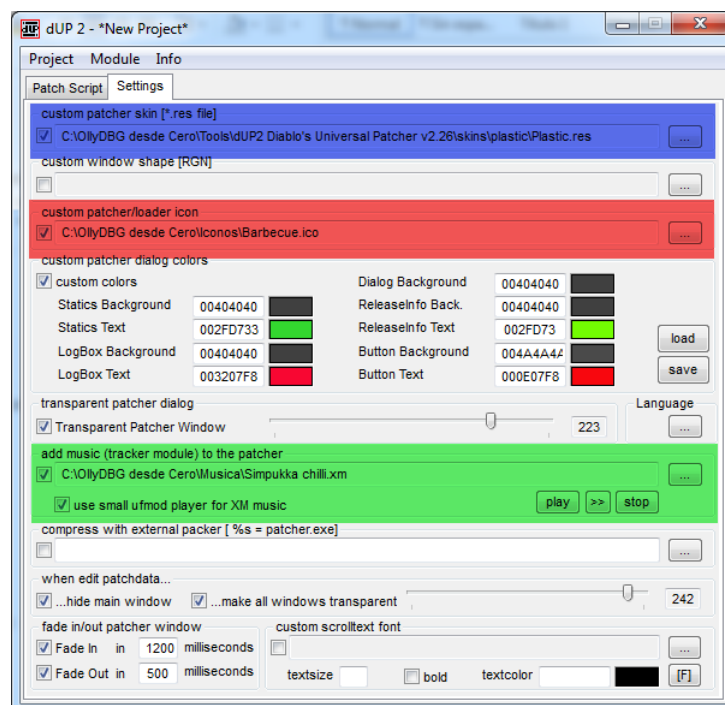


El módulo se agrega, ahora lo seleccionamos para poder agregar nuestra configuración. Lo seleccionamos, <**Module**-><**Edit**> o <**F2**>, también con <**Clic Derecho**-><**Edit**>.

[Tuto007 - Nitro Pro v12.1.0.195 (Crack.Patch)(OllyDBG v1.10)]

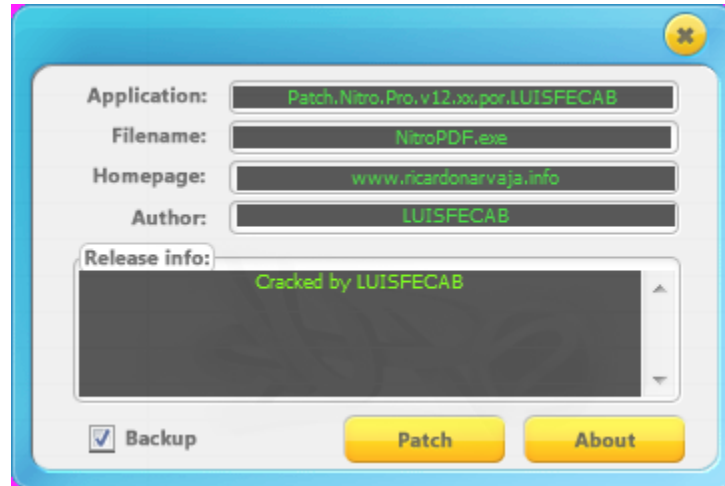


En <Target File> agregamos la ruta completa de nuestra víctima. En <Compare Files> cargamos nuestro archivo original y el crackeado para realizar la comparación con el botón "Compare" y con eso podemos ver que son los mismos bytes que cambiamos nosotros. Solo resta guardar los cambios de este módulo con "Save".

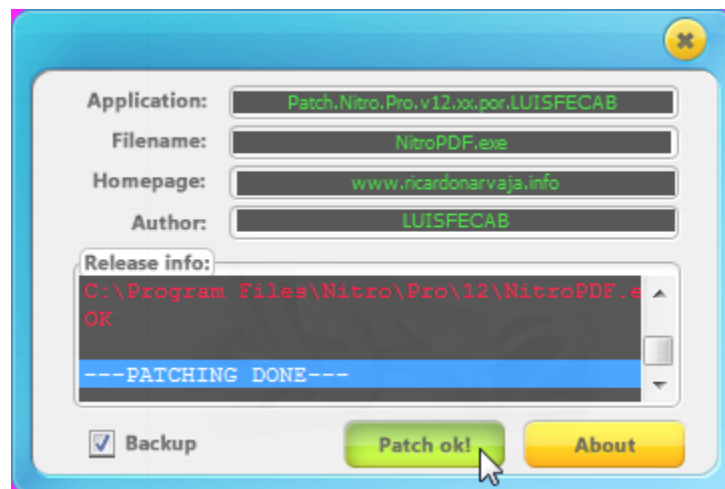


[Tuto007 - Nitro Pro v12.1.0.195 (Crack.Patch)(OllyDBG v1.10)]

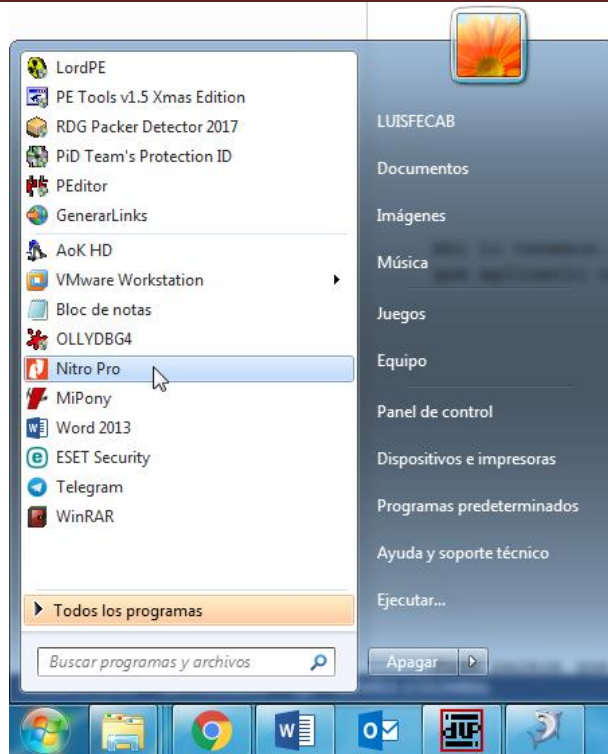
Ahora en la pestaña "**Settings**" le colocamos unos detalles para que quede más chulo. El **SKIN** es uno que venía con este dUP2 cuando lo descargué, el **ICONO** y **MÚSICA** los saqué del "**Xylitol Crypto-KeygenMe 1: BarbecueMe**" con el cual QwErTy hizo un tutorial muy bueno, [1649](#). Solo nos queda guardar el proyecto y generar nuestro Patch. Eso lo haces desde <**Project-Save As**> y <**Project-Create Patch**>. Ahora a ejecutarlo para aplicarlo, a cruzar los dedos.



Ahí lo tenemos. Esta bonito este Skin y con buena música. Bueno, nada más por hacer que aplicarlo con su botón "Patch".



Pues parece que todo OK. Voy a ejecutar el programa a ver qué me sale.



Joder amigos, esto para mí es tremendamente gratificante, me funcionó a la perfección. Cada proyecto de cracking que saco adelante me llena de mucha alegría y con estos tutoriales la deseo transmitir y compartir con ustedes.

PARA TERMINAR

Creo que empezaré diciéndoles que les quedo debiendo el Crack y Patch para la versión de 64 Bits contrario a lo que les dije en la **INTRODUCCIÓN** pero es que por aquí en este momento no dispongo de un Windows de 64 Bits, quedo comprometido con eso y sobre todo para ya irme acoplando también al <x64DBG> que creo ya es hora de abordarlo.

Ahora, hablemos de la protección que ofrece el programa, si bien para crackearlo con serial y para mi nivel es muy difícil pero que pude crackearlo de otra forma, pienso yo, que ellos en si utilizan esa forma porque ahora están ofreciendo un portafolio más amplio de servicios y asesoría, entonces requieren tener control sobre sus clientes para asesorarlos.

Contento como ya les dije hace ratito y como siempre, espero reciban este mi pequeño escrito con agrado, y no duden en corregirme en cualquier fallo de mis interpretaciones.

Felicidades a la lista de CrackSLatinoS que por estos días cumple 18 años de existencia y vigencia. Gracias a Ricardo y a todos los que colaboraron para que eso fuera posible.