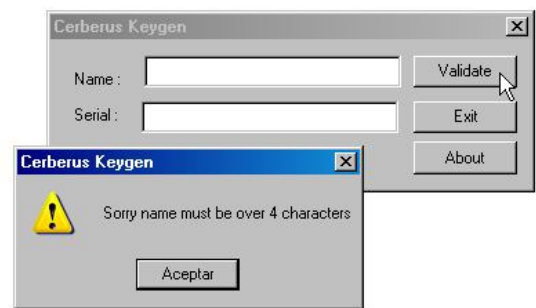
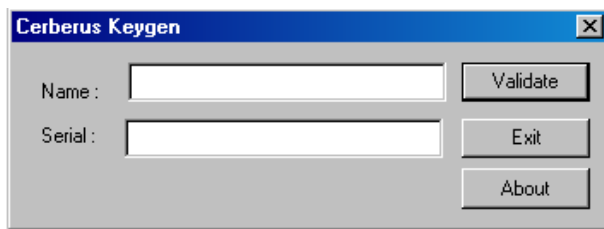


<i>Crackme</i>	<i>Cerberus Keygen</i>
<i>Misión</i>	<i>Buscar Name y Serial Hacer un Selft-Keygen</i>
<i>Compilado</i>	<i>Microsoft Visual C++ ver 5.0/6.0</i>
<i>Protección</i>	<i>Not packed</i>
<i>Herramientas</i>	<i>OllyDbg 1.10 - RDG v0.7.5 - Resource Hacker v4.2.5</i>
<i>Sistema Operativo</i>	<i>Windows Xp SP3</i>
<i>Cracker</i>	<i>QwErTy</i>
<i>Dedicado a</i>	<i>RICNAR - Cerberus - CLS</i>
<i>Link (hay que registrarse)</i>	http://crackmes.de/users/cerberus/keygen/

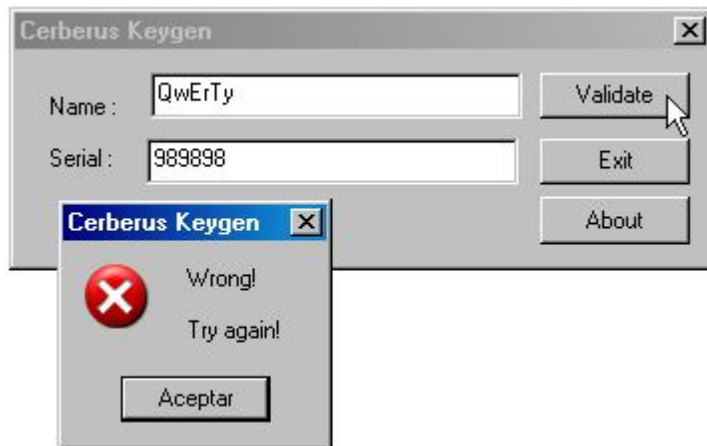
Vaya por delante que para este Crackme es extremadamente fácilón encontrar un serial válido, y sabedor que un buen Cracker no es amante de los Selft-Keygen, (a mí me divierten), creo que también es bueno saber que existe esta técnica, por llamarla de alguna manera. He visto otras soluciones en la Web pero no me convencían del todo. Espero que ésta sea del agrado del lector....

ESTUDIANDO LA VÍCTIMA

Hacemos una copia del Crackme, lo ejecutamos y nos pide un Name y un Code para registrarnos. Si le damos directamente a "Validate" nos salta un mensaje que nos dice que el "Name" debe tener más de 4 caracteres.



Aceptamos, rellenamos datos



Le damos a "Validate" y nos salta el mensaje de chico malo.

CONTINUAMOS ESTUDIANDO LA VÍCTIMA

Abrimos el Crackme con el detector "RDG"

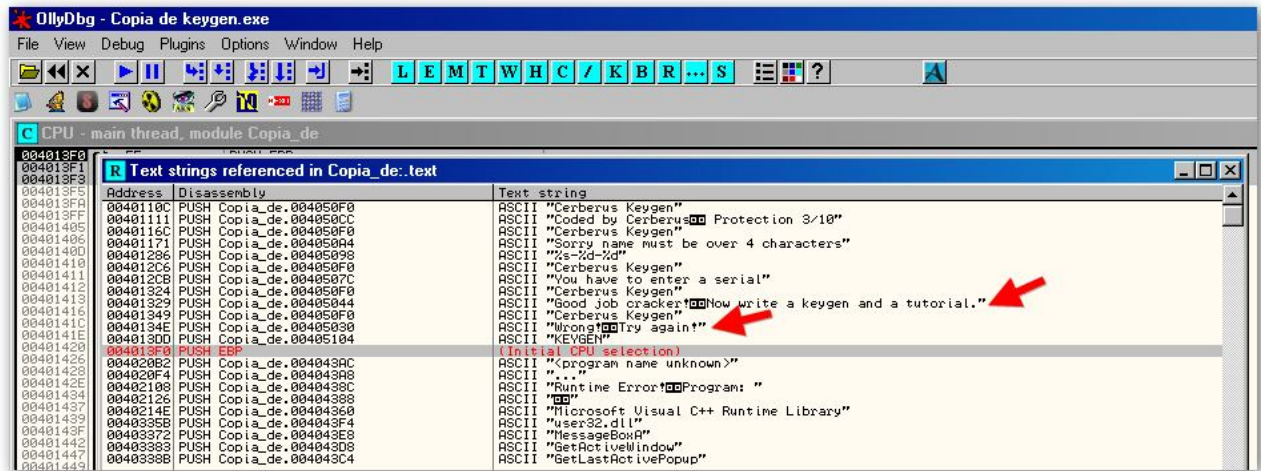


y nos dice que está compilado en "Microsoft Visual C++ 6.0" y que no está empacado. Bien

VAMOS A POR ELLA

PRIMERA MISIÓN: Buscar un Name y un Serial

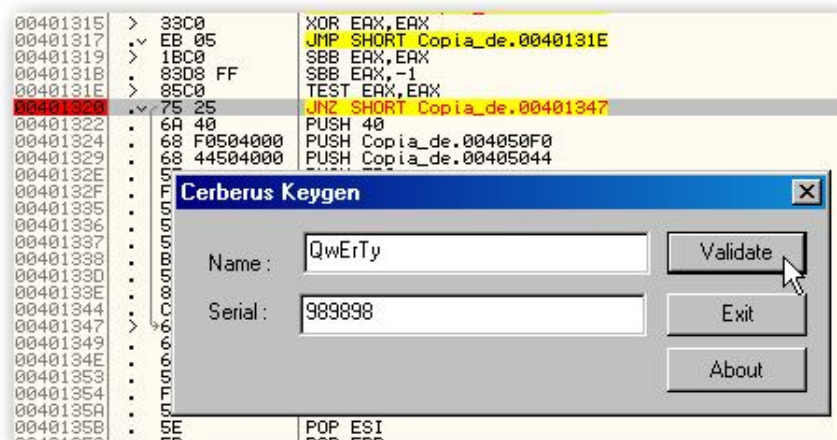
Salimos del "RDG" cargamos el Crackme con Olly, buscamos alguna referencia que nos pueda interesar, y encontramos, entre otras, la de chico bueno, y la de chico malo.



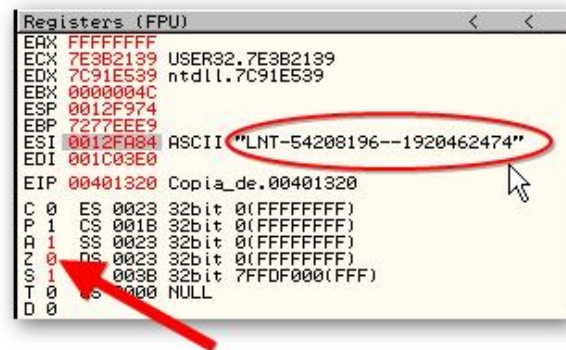
Le damos a cualquiera de las dos y en la ventana del desensamblado aparecemos en la zona caliente, donde vemos los dos "MessageBoxA", un "TEST" y el salto condicional "JNZ" decisivo.



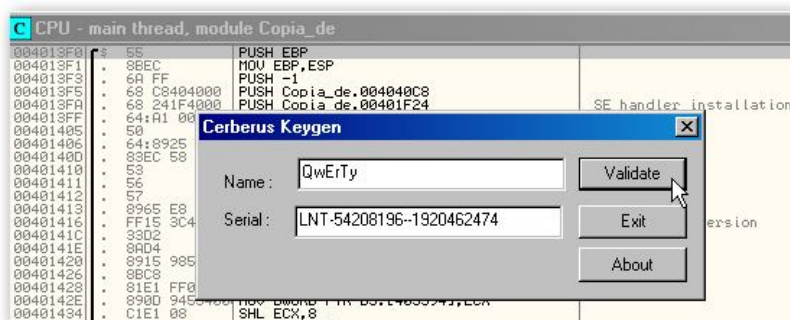
Pues probemos a poner directamente un "Breakpoint" con "F2" a este salto condicional a ver que pasa, je,je,je. Una vez puesto, le damos a "F9" para que corra el Crackme, rellenamos datos



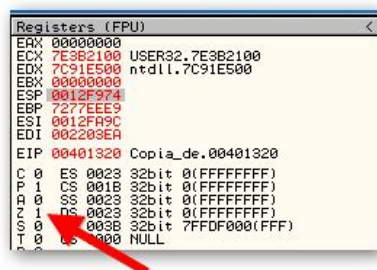
Le damos a "Validate" y en el registro "ESI" aparece un posible "Serial" para nuestro "Name" "LNT-54208196--1920462474". También nos fijamos que al estar el flag "Z" en "0", no se cumplirá la condición del salto y nos mandará a chico malo.



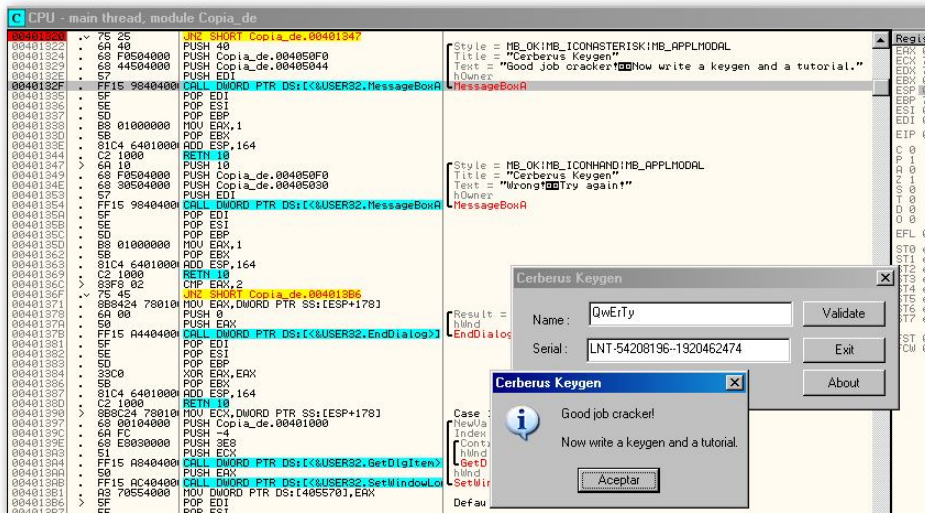
Reiniciamos el Crackme con Olly (tenemos el "BP" puesto en el salto condicional), lo corremos, rellenamos datos, esta vez con el posible serial que obtuvimos anteriormente para nuestro "Name"



Le damos a "Validate" y parados en el "BP", observamos en la ventana "Registers" que ahora el flag "Z" está en "1", y se cumplirá la condición del salto que nos mandará a chico bueno.



Seguimos traceando, y efectivamente nos salta el mensaje de felicitación



Dando con ello nuestra primera misión por finaliza.

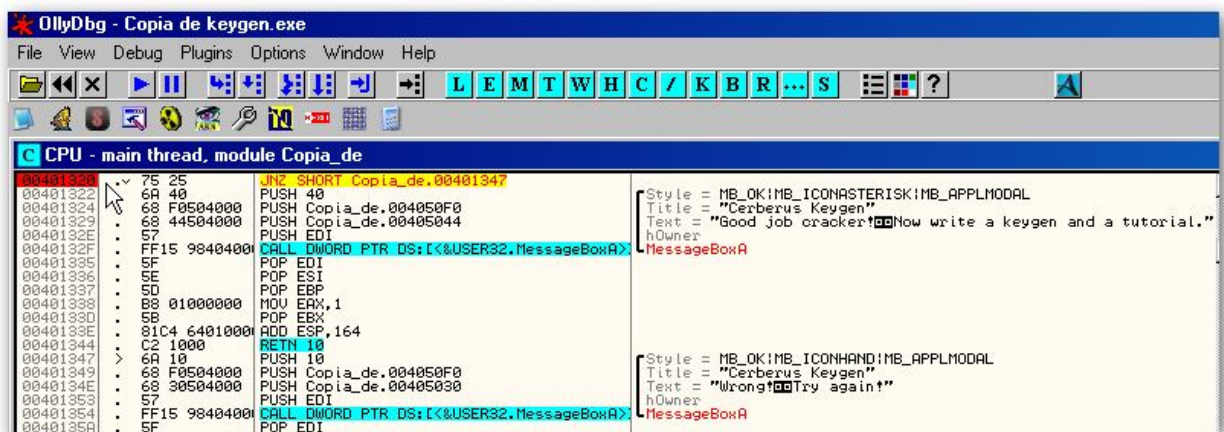
SEGUNDA MISIÓN: Vamos a hacer un Self-Keygen (generador de llaves con la misma víctima)

Para ello utilizaremos la APIs `"User32.SetDlgItemTextA"`. Esta función envía un mensaje `"WM_SETTEXT"` al control especificado, y también establece el título o texto de un control en un cuadro de diálogo.

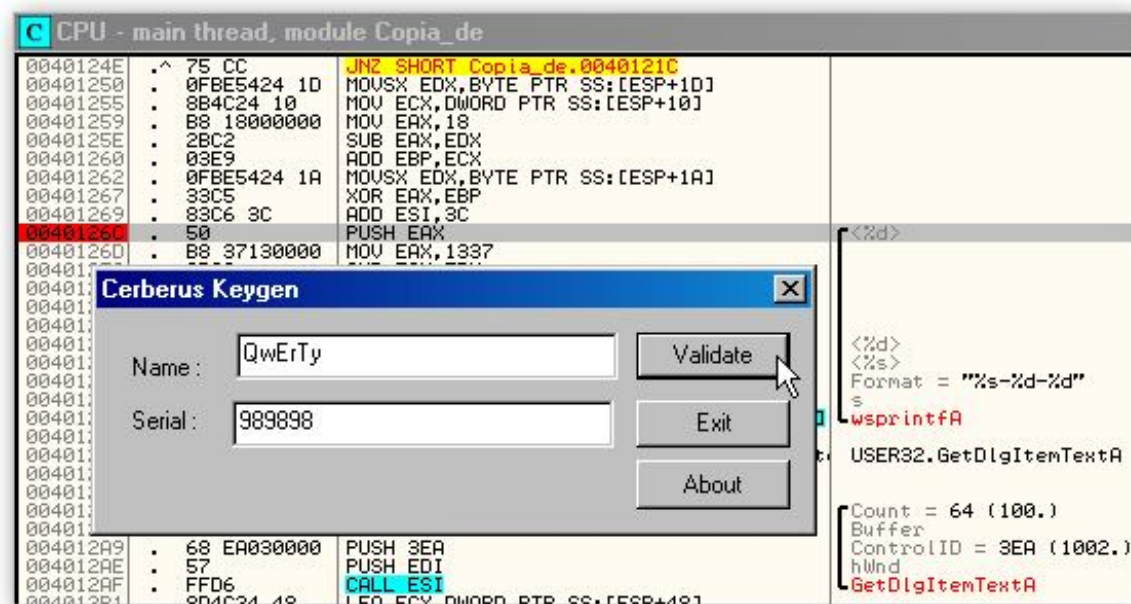
Debemos saber que para que esta APIs funcione, necesitamos el `"handle"` de la ventana, el `"ID"` de la caja de texto y la `"String"` a mostrar.

Pues vamos a buscar todos estos datos.

Reiniciamos el Crackme con Olly, y nos vamos directamente a la address `"00401320"` donde tenemos el `"Breakpoint"`



Subimos haciendo `"scroll"` hasta `"0040126C"`, y le ponemos otro Breakpoint con `"F2"`, corremos el Crackme, rellenamos datos



Le damos a `"Validate"` y Olly para en nuestro `"BP"`.

OllyDbg - Copia de keygen.exe

File View Debug Plugins Options Window Help

CPU - main thread, module Copia_de

0040124E	75 CC	JNZ SHORT Copia_de.0040121C	
00401250	0FB5424 1D	MOVSX EDX, BYTE PTR SS:[ESP+10]	
00401255	8B4C24 10	MOV ECX, DWORD PTR SS:[ESP+10]	
00401259	B8 18000000	MOV EAX, 18	
0040125E	2BC2	SUB EAX, EDX	
00401260	03E9	ADD EBP, ECX	
00401262	0FB5424 1A	MOVSX EDX, BYTE PTR SS:[ESP+1A]	
00401267	33C5	XOR EAX, EBP	
00401269	83C6 3C	ADD ESI, 3C	
0040126C	50	PUSH EAX	<%d> = 8D881176 (-1920462474.)
0040126D	B8 37130000	MOV EAX, 1337	
00401272	2BC2	SUB EAX, EDX	
00401274	8D8C24 B0000000	LEA ECX, DWORD PTR SS:[ESP+B0]	
00401278	33C6	XOR EAX, ESI	
0040127D	8D9424 14010000	LEA EDX, DWORD PTR SS:[ESP+114]	
00401284	50	PUSH EAX	<%d>
00401285	51	PUSH ECX	<%s>
00401286	68 98504000	PUSH Copia_de.00405098	Format = \"%s-%d-%d\"
00401288	52	PUSH EDX	s
0040128C	FF15 A0404000	CALL DWORD PTR DS:[&USER32.wsprintfA]	wsprintfA
00401292	8B8C24 8C010000	MOV EDI, DWORD PTR DS:[ESP+18C]	

Ahora vamos traceando con "F8" hasta llegar a la address "00401292", y aquí parados, en la ventana "Stack" vemos el Serial válido

00401267	33C5	XOR EAX, EBP	
00401269	83C6 3C	ADD ESI, 3C	
0040126C	50	PUSH EAX	
0040126D	B8 37130000	MOV EAX, 1337	
00401272	2BC2	SUB EAX, EDX	
00401274	8D8C24 B0000000	LEA ECX, DWORD PTR SS:[ESP+B0]	
00401278	33C6	XOR EAX, ESI	
0040127D	8D9424 14010000	LEA EDX, DWORD PTR SS:[ESP+114]	
00401284	50	PUSH EAX	<%d>
00401285	51	PUSH ECX	<%s>
00401286	68 98504000	PUSH Copia_de.00405098	Format = \"%s-%d-%d\"
00401288	52	PUSH EDX	s
0040128C	FF15 A0404000	CALL DWORD PTR DS:[&USER32.wsprintfA]	wsprintfA
00401292	8B8C24 8C010000	MOV EDI, DWORD PTR DS:[ESP+18C]	
00401299	8B35 9C404000	MOV ESI, DWORD PTR DS:[&USER32.GetDlgItemTextA]	USER32.GetDlgItemTextA
0040129F	83C4 14	ADD ESP, 14	

0012F960	0012FA84	ASCII "LNT-54208196--1920462474"
0012F964	0012FA88	ASCII "%s-%d-%d"
0012F968	0012FA8C	ASCII "LNT"
0012F96C	033B28C4	
0012F970	8D881176	
0012F974	0012FB4C	
0012F978	00401080	Copia_de.00401080
0012F97C	0012FB10	
0012F980	00000000	
0012F984	7277297B	
0012F988	0012F98C	ASCII "QwErTy"
0012F98C	72457751	
0012F990	00007954	

Bien, ya tenemos nuestro primer dato que buscamos, nos quedamos con "0012FA84" que es uno de los lugares donde guarda el serial válido en memoria, y lo apuntamos en un "Notepad"

Sin título - Bloc de notas

Archivo Edición Formato Ver Ayuda

0012FA84 -----> guarda el serial válido en memoria

Continuamos traceando hasta la address "004012B5", y en la ventana "Registers" aparece nuestro Serial falso

00401267	33C5	XOR EAX,EBP	
00401269	83C6 3C	ADD ESI,3C	
0040126C	50	PUSH EAX	<%d>
0040126D	B8 37130000	MOV EAX,1337	
00401272	2BC2	SUB EAX,EDX	<%s>
00401274	8D8C24 B0000000	LEA ECX,DWORD PTR SS:[ESP+B0]	Format = "%s-%d-%d"
0040127B	33C6	XOR EAX,ESI	s
0040127D	8D9424 14010000	LEA EDX,DWORD PTR SS:[ESP+114]	
00401284	50	PUSH EAX	
00401285	51	PUSH ECX	
00401286	68 98504000	PUSH Copia_de.00405098	
0040128B	52	PUSH EDX	
0040128C	FF15 A0404000	CALL DWORD PTR DS:[&USER32.wsprintfA]	wsprintfA
00401292	8B8C24 8C010000	MOV EDI,DWORD PTR SS:[ESP+18C]	
00401299	8B35 9C404000	MOV ESI,DWORD PTR DS:[&USER32.GetDlgItemTextA]	USER32.GetDlgItemTextA
0040129F	83C4 14	ADD ESP,14	
004012A2	8D4424 48	LEA EAX,DWORD PTR SS:[ESP+48]	
004012A6	6A 64	PUSH 64	Count = 64 (100.)
004012A8	50	PUSH EAX	Buffer
004012A9	68 EA030000	PUSH 3EA	ControlID = 3EA (1002.)
004012AE	57	PUSH EDI	hWnd
004012AF	FFD6	CALL ESI	GetDlgItemTextA
004012B1	8D4C24 48	LEA ECX,DWORD PTR SS:[ESP+48]	
004012B5	6A 64	PUSH 64	Count = 64 (100.)
004012B7	51	PUSH ECX	Buffer
004012B8	68 EA030000	PUSH 3EA	ControlID = 3EA (1002.)
004012BD	57	PUSH EDI	hWnd
004012BE	FFD6	CALL ESI	GetDlgItemTextA
004012C0	85C0	TEST EAX,EAX	
004012C2	75 22	JNZ SHORT Copia_de.004012E6	
004012C4	6A 30	PUSH 30	Style = MB_OK MB_ICONEXCLAMATION MB_APPLMODAL
004012C6	68 F0504000	PUSH Copia_de.004050F0	Title = "Cerberus Keygen"
004012CB	68 7C504000	PUSH Copia_de.0040507C	Text = "You have to enter a serial"
004012D0	57	PUSH EDI	hOwner
004012D1	FF15 98404000	CALL DWORD PTR DS:[&USER32.MessageBoxA]	MessageBoxA
004012D7	5F	POP EDI	

Registers (FPU)	
EAX	00000006
ECX	0012F98C ASCII "989898"
EDX	7C91E514 ntdll.KiFastSystemCallRet
EBX	00000000
ESP	0012F974
EBP	727EEEE9
ESI	7E3EB05E USER32.GetDlgItemTextA
EDI	0019036E
EIP	004012B5 Copia_de.004012B5

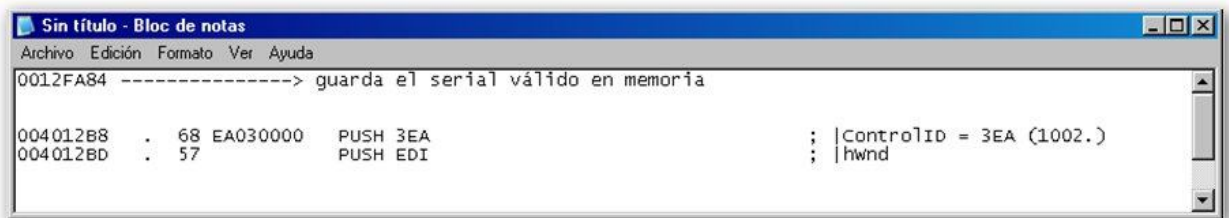
Con este detalle, intuimos que estamos en la parte de código correcto, y de que también estamos dentro del evento que nos interesa para nuestro propósito, donde deberemos modificar la instrucción para que en la caja de texto nos muestre el serial válido en lugar de leer el falso.

También hemos encontrado dentro del evento los demás datos que nos faltaban, o sea el Control "ID" y el "handle", que también los apuntaremos en el "Notepad":

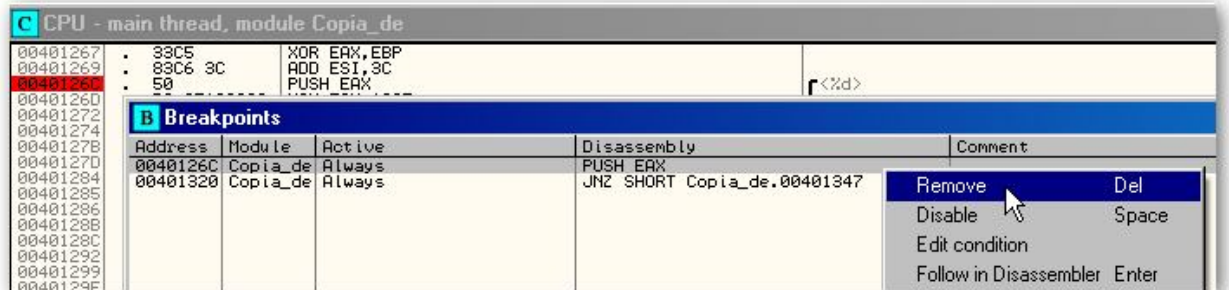
El Control "ID" de la caja de texto, es "PUSH 3EA" y el "handle" de la ventana, es "PUSH EDI".

00401267	33C5	XOR EAX,EBP	
00401269	83C6 3C	ADD ESI,3C	
0040126C	50	PUSH EAX	<%d>
0040126D	B8 37130000	MOV EAX,1337	
00401272	2BC2	SUB EAX,EDX	<%s>
00401274	8D8C24 B0000000	LEA ECX,DWORD PTR SS:[ESP+B0]	Format = "%s-%d-%d"
0040127B	33C6	XOR EAX,ESI	s
0040127D	8D9424 14010000	LEA EDX,DWORD PTR SS:[ESP+114]	
00401284	50	PUSH EAX	
00401285	51	PUSH ECX	
00401286	68 98504000	PUSH Copia_de.00405098	
0040128B	52	PUSH EDX	
0040128C	FF15 A0404000	CALL DWORD PTR DS:[&USER32.wsprintfA]	wsprintfA
00401292	8B8C24 8C010000	MOV EDI,DWORD PTR SS:[ESP+18C]	
00401299	8B35 9C404000	MOV ESI,DWORD PTR DS:[&USER32.GetDlgItemTextA]	USER32.GetDlgItemTextA
0040129F	83C4 14	ADD ESP,14	
004012A2	8D4424 48	LEA EAX,DWORD PTR SS:[ESP+48]	
004012A6	6A 64	PUSH 64	Count = 64 (100.)
004012A8	50	PUSH EAX	Buffer
004012A9	68 EA030000	PUSH 3EA	ControlID = 3EA (1002.)
004012AE	57	PUSH EDI	hWnd
004012AF	FFD6	CALL ESI	GetDlgItemTextA
004012B1	8D4C24 48	LEA ECX,DWORD PTR SS:[ESP+48]	
004012B5	6A 64	PUSH 64	Count = 64 (100.)
004012B7	51	PUSH ECX	Buffer
004012B8	68 EA030000	PUSH 3EA	ControlID = 3EA (1002.)
004012BD	57	PUSH EDI	hWnd
004012BE	FFD6	CALL ESI	GetDlgItemTextA
004012C0	85C0	TEST EAX,EAX	
004012C2	75 22	JNZ SHORT Copia_de.004012E6	
004012C4	6A 30	PUSH 30	Style = MB_OK MB_ICONEXCLAMATION MB_APPLMODAL
004012C6	68 F0504000	PUSH Copia_de.004050F0	Title = "Cerberus Keygen"
004012CB	68 7C504000	PUSH Copia_de.0040507C	Text = "You have to enter a serial"
004012D0	57	PUSH EDI	hOwner
004012D1	FF15 98404000	CALL DWORD PTR DS:[&USER32.MessageBoxA]	MessageBoxA
004012D7	5F	POP EDI	

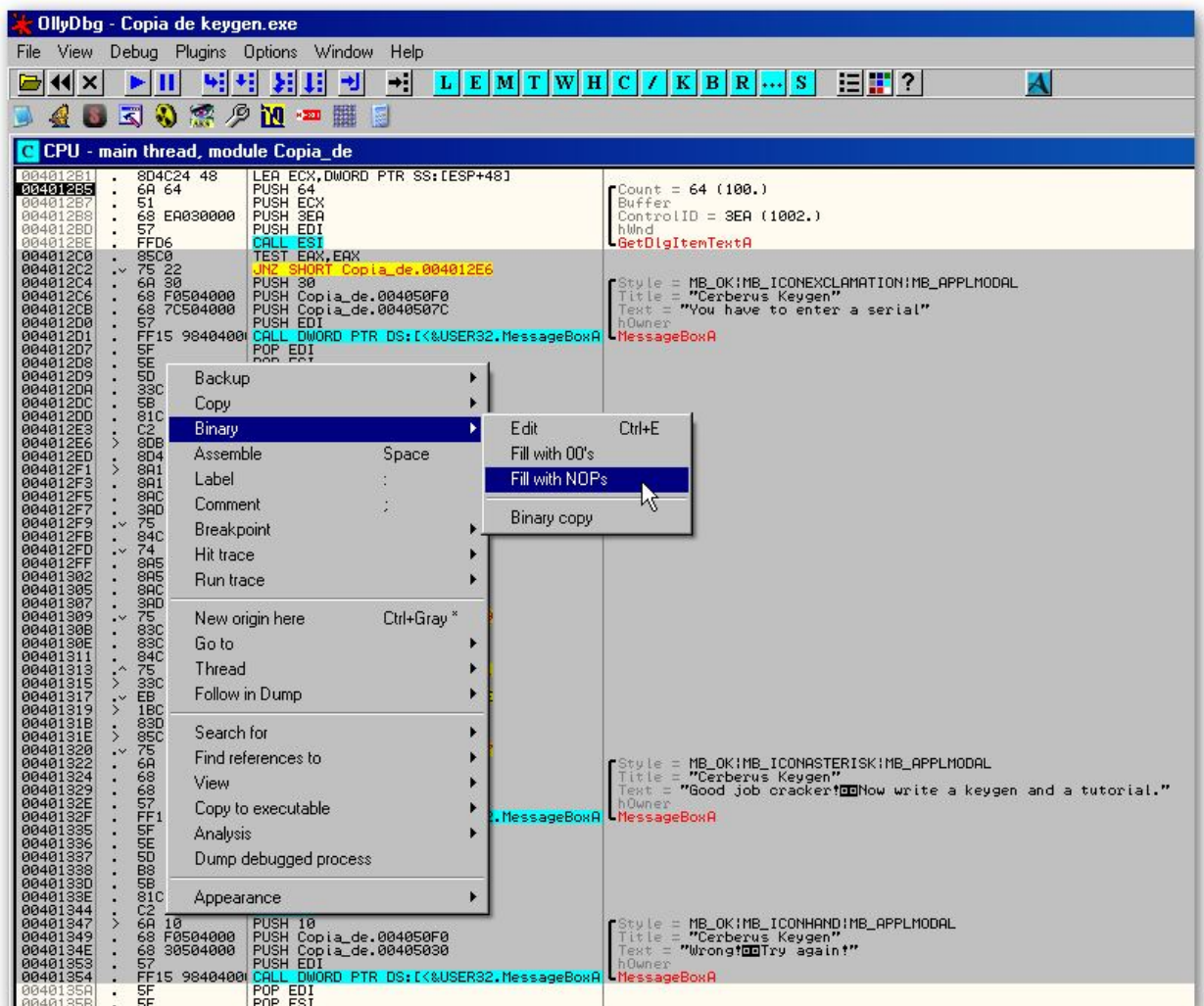
Nuestro "Notepad" queda ahora de la siguiente manera



Acto seguido ya podemos borrar los dos "BP"

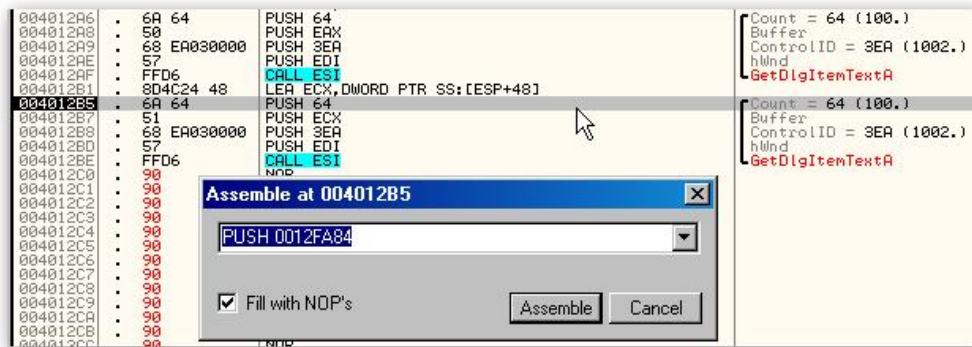


Y NOPeamos la parte de código que no nos interesa para nada, y que va desde la address "004012C0" hasta la "00401354"

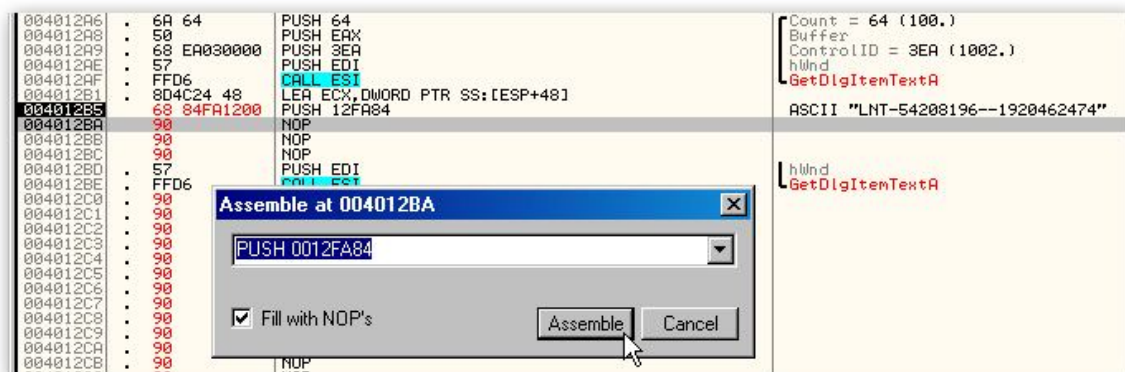


Analizamos código para que se vea todo mejor, y haciendo "scroll" volvemos a la posición donde estábamos parados, y que era la address "004012B5"

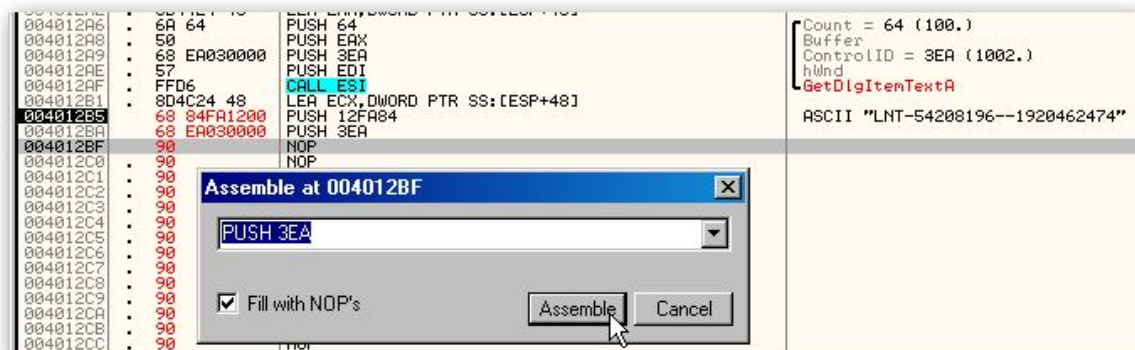
Una vez aquí, primero cambiamos la instrucción "PUSH 64" por "PUSH 0012FA84" que es como vimos antes, donde guarda el serial válido en memoria para nuestro "Name"



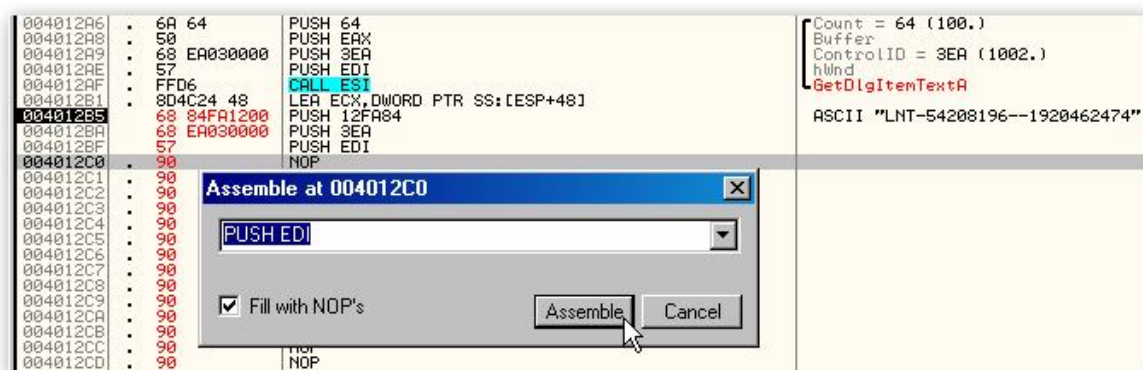
Quedando así



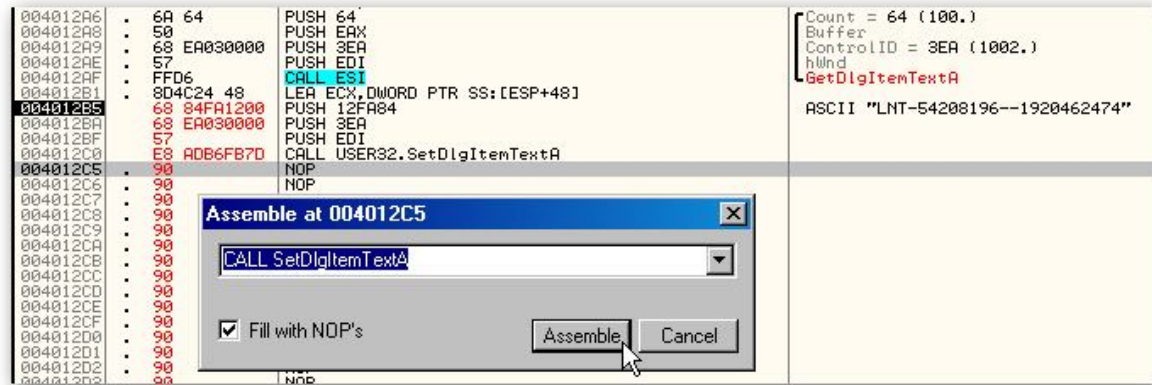
Segundo, insertamos "PUSH 3EA" que es el Control "ID" de la caja de texto



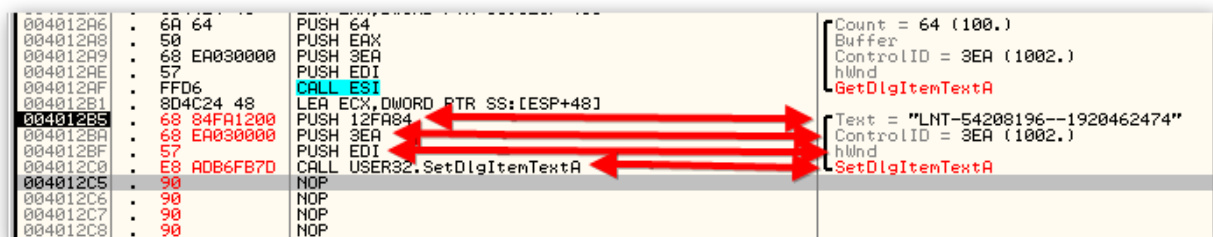
Tercero, insertamos "PUSH EDI" que es el "handle" de la ventana



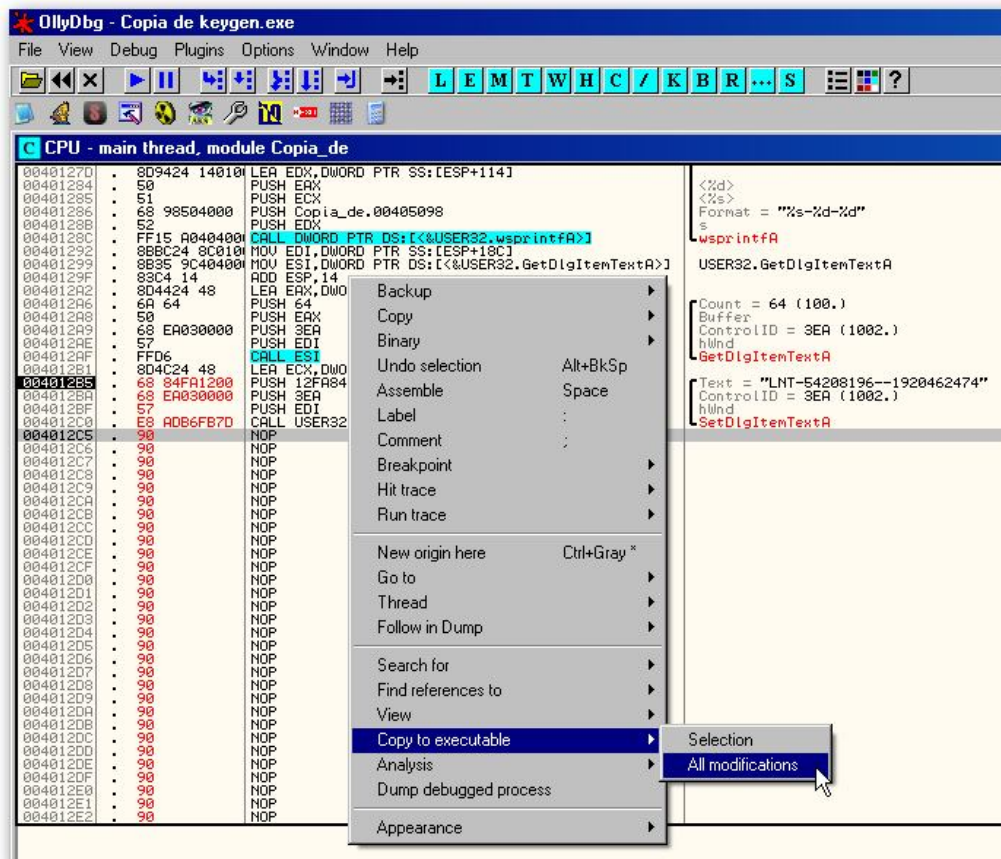
Y, cuarto, insertamos la función APIs "**CALL SetDlgItemTextA**" que se encargará de mostrarnos el Serial válido para nuestro Name en la caja de texto.



Analizamos código para que se reorganice todo, y nos queda así de guapo...



Por último, guardamos todas las modificaciones



Renombramos , yo le he llamado, **Selft-Keygen.exe** y le damos a "Guardar"

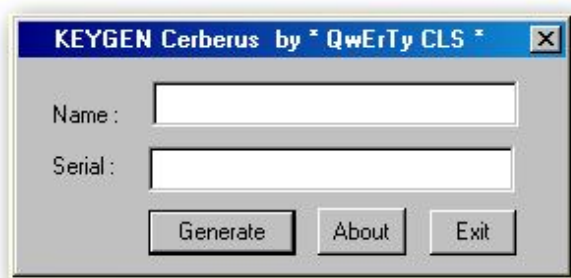


Ahora ya podemos salir de Olly, nos vamos a la ruta donde lo hemos guardado, y comprobamos que funcione.

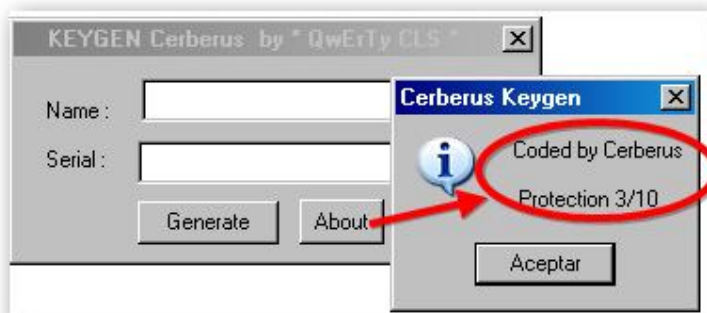


Yfunciona de maravilla

Ahora, su aspecto exterior no queda muy serio que digamos para un auténtico generador de llaves. Salimos de nuestro "Selft-Keygen.exe" lo maqueamos con la Tool "Resource Hacker" y lo dejamos así:

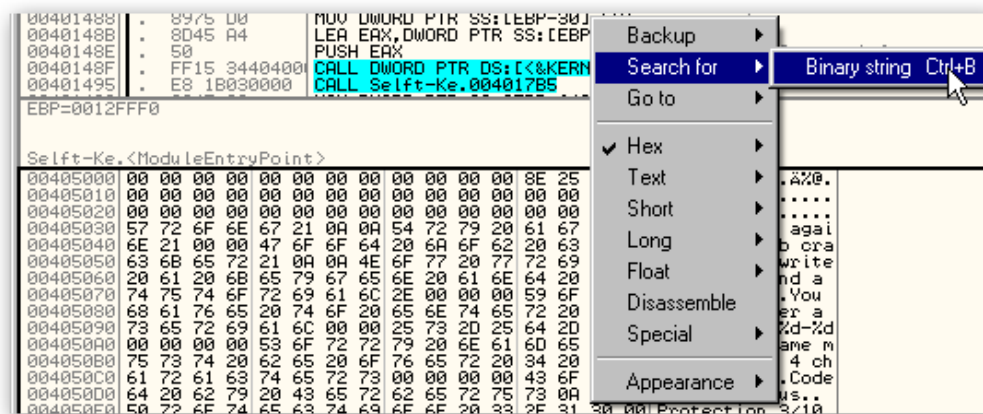


Pero aún nos queda un último detalle..... si le damos a la tecla "About" nos muestra el cartelito de

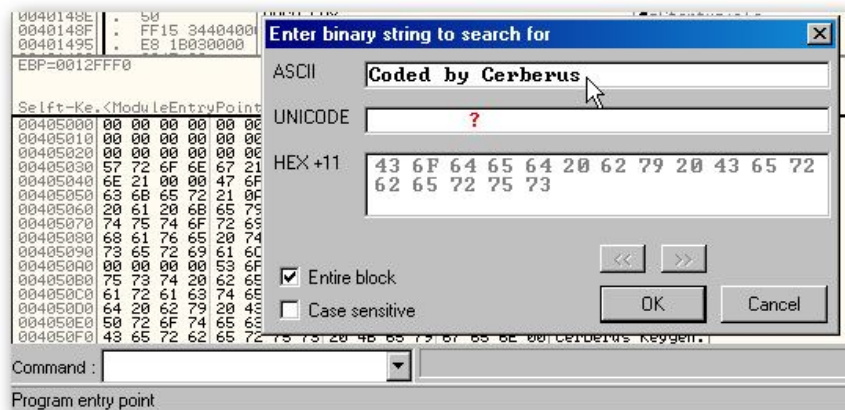


Pues vamos a cambiar ese texto con Olly (aunque también podríamos hacerlo con un Editor Hexadecimal).

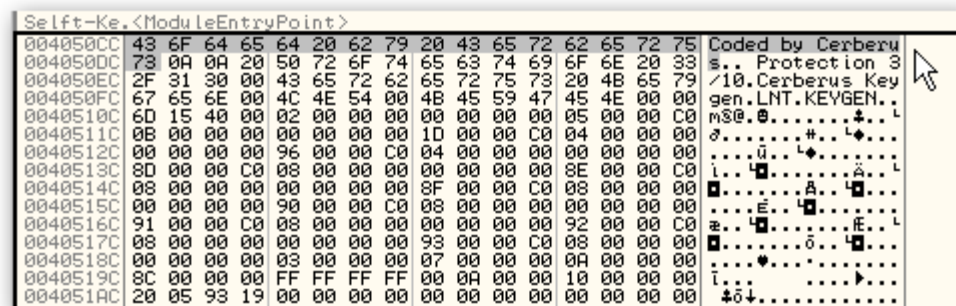
Salimos de nuestro "Selft-Keygen.exe". Lo cargamos con Olly, nos posicionamos sobre la ventana "Dump" hacemos "Search for - Binary string "



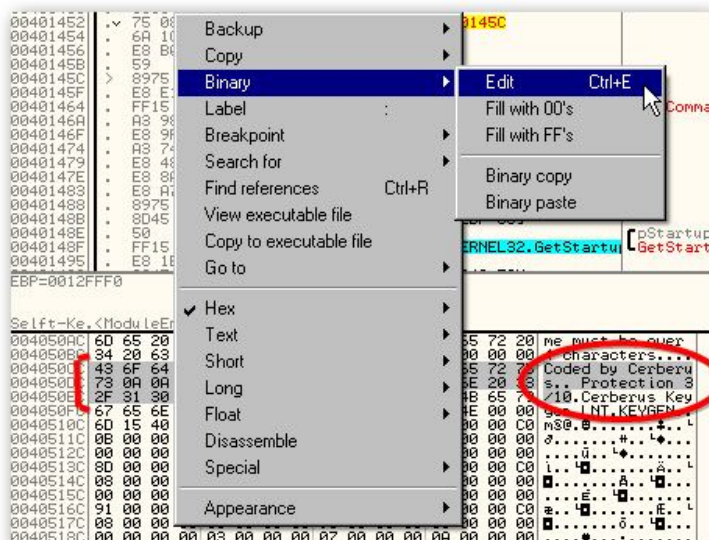
Ahora en la caja de texto "ASCII" tipeamos el mensaje a buscar o parte del mismo



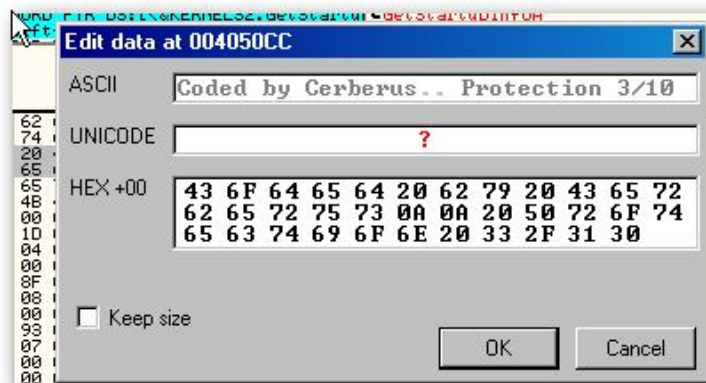
Le damos a "OK" y aparecemos aquí, justo lo que andamos buscando



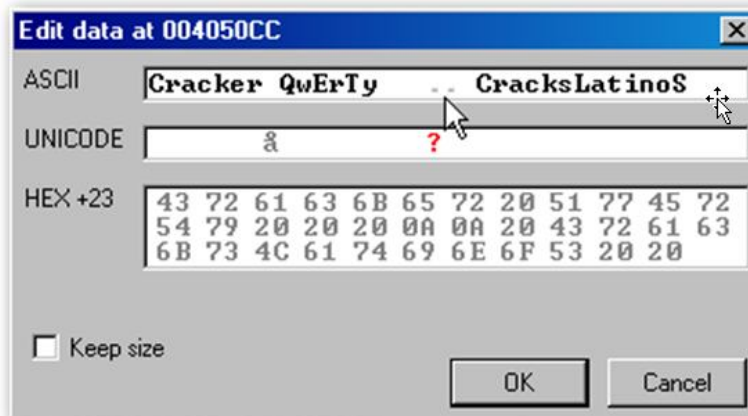
Sombreamos lo que queremos cambiar, editamos el Binario "Binary - Edit"



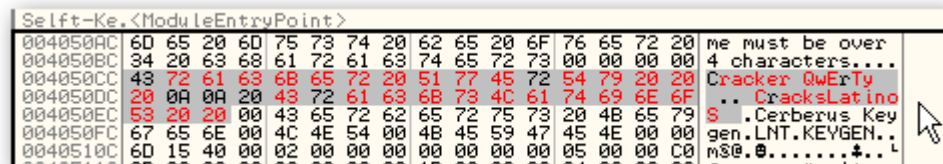
Cambiamos el texto por el que queramos (Respetando el largo)



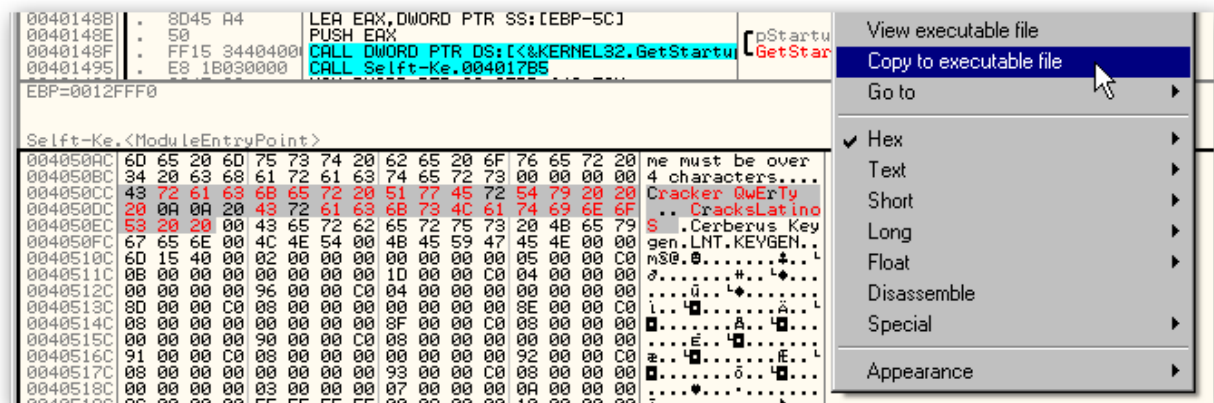
Yo tipeo



Le damos a "OK" y en la ventana "Dump" nos quedará así:



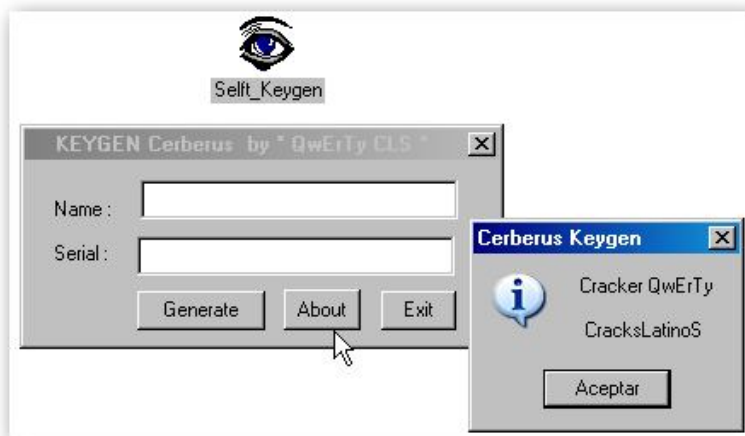
Ahora para guardar los cambios sobre la misma ventana del "Dump" y con el texto marcado, damos clic derecho de ratón y "Copy to executable file"



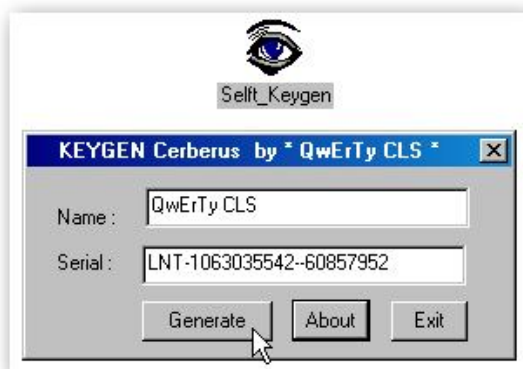
Renombramos el nombre del ".exe" con el que estamos trabajando por si las moscas, (si algo va mal siempre tendremos el ".exe" antes de la modificación), yo he cambiado el guión medio por guión bajo, y le damos a "Guardar".



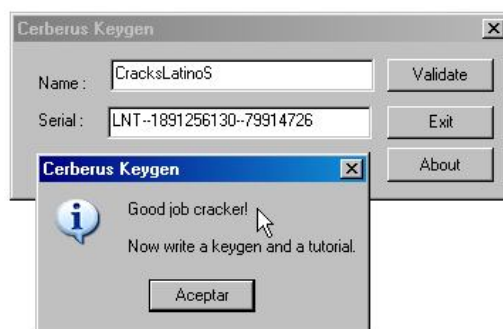
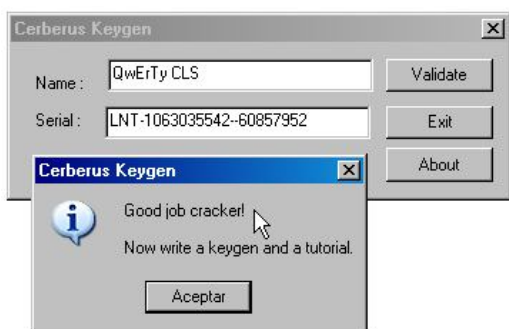
Ya podemos salir de Olly, vamos a la ruta donde tenemos nuestro flamante "Selft_Keygen.exe" final, lo ejecutamos, le damos a "About" y comprobamos que el cambio que hemos hecho haya surtido efecto..... y aquí lo tenemos.....



Y generando seriales a la perfección:



Comprobamos los Serials generados y.....



Se nos cae la baba sobre el teclado de lo orgullosos que estamos de nosotros mismos al ver finalizada y con éxito nuestra esplendida obra, decimos en voz alta QUE BIEN VA, los labios se nos ensanchan de oreja a oreja, y nuestra felicidad no tiene precio al ver también nuestra segunda misión finalizada y descaradamente recompensada.

A, se me olvidaba, ya podemos cerrar el Notepad....

.....MISIÓN CUMPLIDA.....



Mis agradecimientos infinitos a

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

La lectura solo proporciona a la mente material de conocimiento;
es el pensamiento lo que hace lo que leemos nuestro.

John Locke

5 de Noviembre de 2016