Victima	Un par de bpls
Protección	PC Guard
Herramientas	Olly, LordPE, Pupe2002 y Hex WorkShop
Objetivo	Desempacar

Introducción

Hola a todos,

Este tutorial va dedicado a Xilefare, el cual me pidió un poco de ayuda con unas dlls (mejor dicho bpl) que estaban empacado con PC Guard.

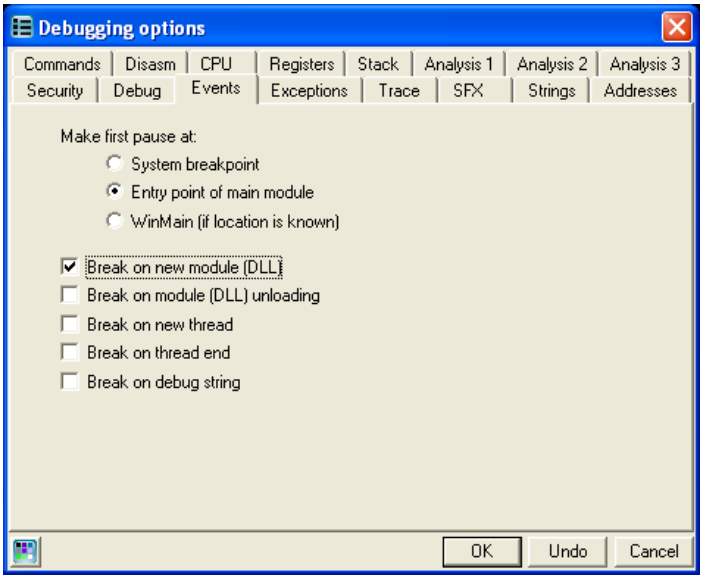
Para el que no lo sepa un bpl es un componente de Borland, y si mal no me equivoco es de C++Builder. Estos componentes no son más que dlls con la extensión cambiada.

La protección de PC Guard es muy sencilla, casi ronda el UPX, pero al estar en una dll veremos que tiene una peculiaridad de más que hay que tener en cuenta.

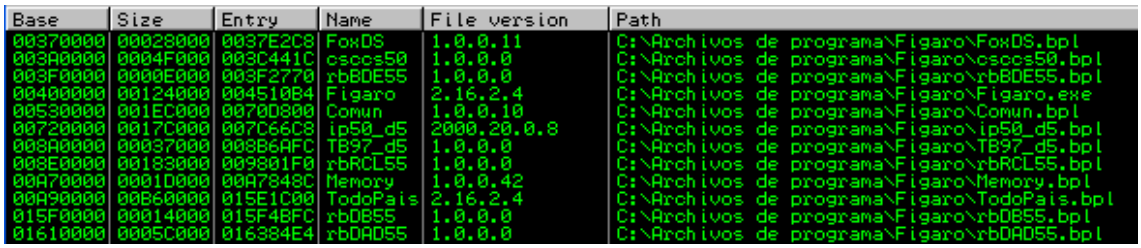
Buscando el OEP

El programa que tiene estas bpl es un tal Figaro. Aparte de esto dispone de una protección con dongle y no se que más. El objetivo de este tutorial no es desproteger el programa, es una simple escusa para estudiar la protección de PC Guard. El que quiera desproteger el programa es ya cosa suya.

Abrimos Olly y lo configuramos para que pare en la carga de las dlls.

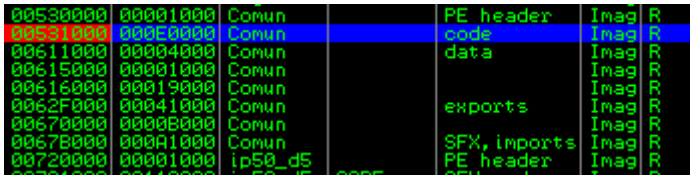


Arrancamos el programa figaro.exe y nos para aquí



Base	Size	Entry	Name	File version	Path
00370000	00028000	0037E2C8	FoxDS	1.0.0.11	C:\Archivos de programa\Figaro\FoxDS.bpl
003A0000	0004F000	003C441C	csccs50	1.0.0.0	C:\Archivos de programa\Figaro\csccs50.bpl
003F0000	0000E000	003F2770	rbBDE55	1.0.0.0	C:\Archivos de programa\Figaro\rbBDE55.bpl
00400000	00124000	004510B4	Figaro	2.16.2.4	C:\Archivos de programa\Figaro\Figaro.exe
00530000	001EC000	0070D800	Comun	1.0.0.10	C:\Archivos de programa\Figaro\Comun.bpl
00720000	0017C000	007C66C8	ip50_d5	2000.20.0.8	C:\Archivos de programa\Figaro\ip50_d5.bpl
008A0000	00037000	008B6AFC	TB97_d5	1.0.0.0	C:\Archivos de programa\Figaro\TB97_d5.bpl
008E0000	00183000	009801F0	rbRCL55	1.0.0.0	C:\Archivos de programa\Figaro\rbRCL55.bpl
00A70000	0001D000	00A7848C	Memory	1.0.0.42	C:\Archivos de programa\Figaro\Memory.bpl
00A90000	00B60000	015E1C00	TodoPais	2.16.2.4	C:\Archivos de programa\Figaro\TodoPais.bpl
015F0000	00014000	015F4BFC	rbDB55	1.0.0.0	C:\Archivos de programa\Figaro\rbDB55.bpl
01610000	0005C000	016384E4	rbDAD55	1.0.0.0	C:\Archivos de programa\Figaro\rbDAD55.bpl

Nos vamos a la ventana Memory y buscamos la sección code de Comun.bpl (la primera victima) y le ponemos un BMP



00530000	00001000	Comun		PE header	Imag	R
00531000	000E0000	Comun		code	Imag	R
00611000	00004000	Comun		data	Imag	R
00615000	00001000	Comun			Imag	R
00616000	00019000	Comun			Imag	R
0062F000	00041000	Comun		exports	Imag	R
00670000	0000B000	Comun			Imag	R
0067B000	000A1000	Comun		SFX, imports	Imag	R
00720000	00001000	ip50_d5		PE header	Imag	R

Damos a run y nos para aquí

```
00719981   281F                SUB BYTE PTR DS:[EDI],BL
00719983   50                  PUSH EAX
00719984   8B85 2D4B4100       MOV EAX,DWORD PTR SS:[EBP+414B2D]
0071998A   3207                XOR AL,BYTE PTR DS:[EDI]
0071998C   D1C0                ROL EAX,1
0071998E   8985 2D4B4100       MOV DWORD PTR SS:[EBP+414B2D],EAX
00719994   58                  POP EAX
00719995   47                  INC EDI
00719996   59                  POP ECX
00719997 ^ E2 AC               LOOPD SHORT Comun.00719945
00719999   C3                  RETN
```

Estamos en un bucle, ponemos un BP en el ret, damos a F9 y cuando pare quitamos el BP volvemos a poner el BMP en la sección code.

Nos vuelve a parar en un bucle

```
00719CE5   01140F              ADD DWORD PTR DS:[EDI+ECX],EDX
00719CE8 v EB 00               JMP SHORT Comun.00719CEA
00719CEA   83C6 02             ADD ESI,2
00719CED   59                  POP ECX
00719CEE ^ E2 C8               LOOPD SHORT Comun.00719CB8
00719CF0 ^ EB AE               JMP SHORT Comun.00719CA0
00719CF2   C3                  RETN
```

Lo mismo BP en el ret F9, cuando pare quitamos el BP y otra vez el BMP en la sección code

Despues de que olly nos pare en unas cargas de dlls, terminamos llegando aquí

```
005CE280 ^ E9 D333F6FF         JMP Comun.00531658
005CE285   8D40 00             LEA EAX,DWORD PTR DS:[EAX]
005CE288   0000                ADD BYTE PTR DS:[EAX],AL
```

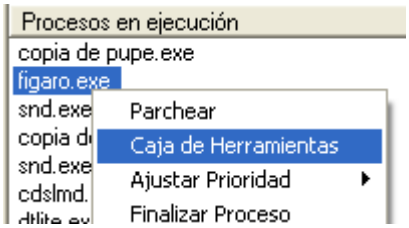
Seguimos el jmp y llegamos al OEP

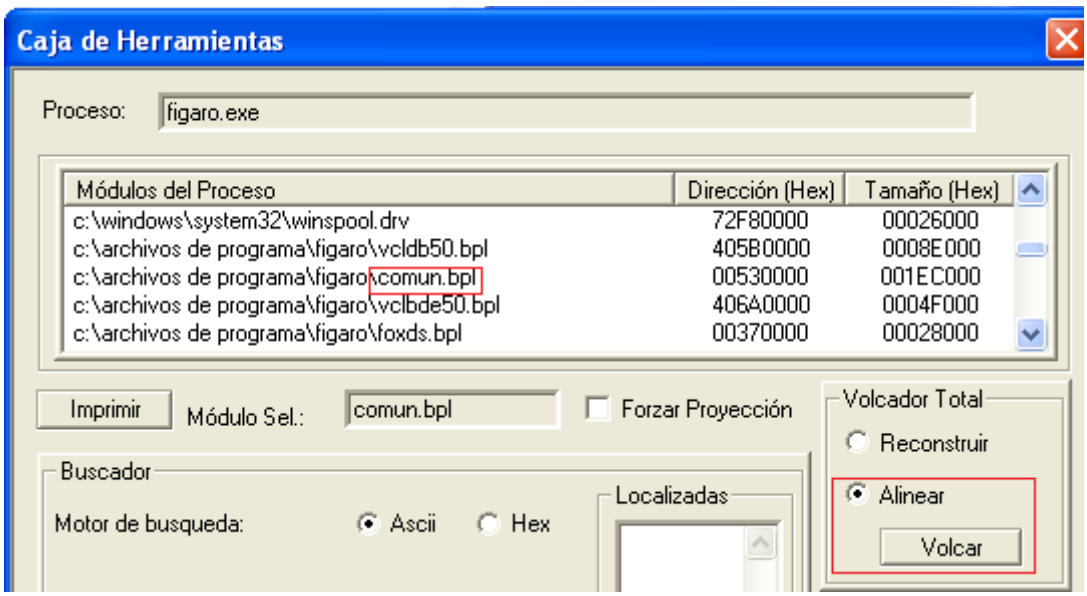
```
00531658   55                  PUSH EBP
00531659   8BEC                MOV EBP,ESP
0053165B   53                  PUSH EBX
0053165C   56                  PUSH ESI
0053165D   8B5D 0C             MOV EBX,DWORD PTR SS:[EBP+C]
00531660   8B45 08             MOV EAX,DWORD PTR SS:[EBP+8]
```

Dumpeando

Si intentamos hacer el dump con Olly este intenta pillar el .exe y no la dll. Alternativas a esto es LordPE, PUPE2002 y SirPE entre otros. Como estoy en la VM de 32bits esta vez no podremos usar el SirPE.

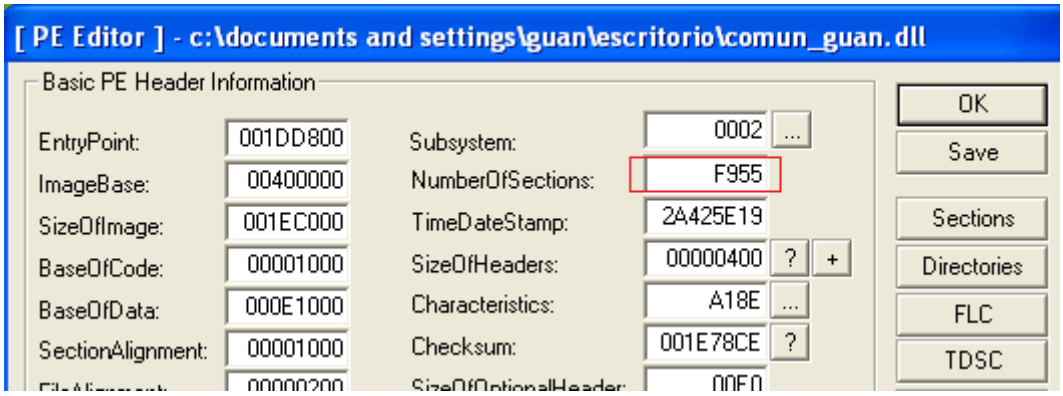
Esta vez LordPE me falló, luego veremos porqué así que heché mano del PuPe2002



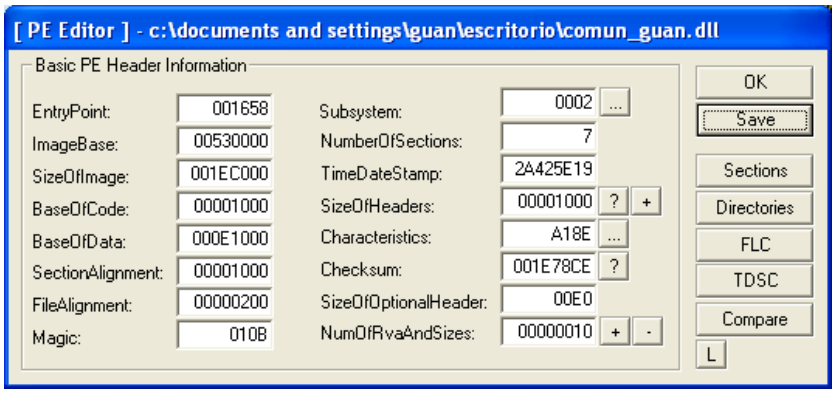


Ya luego seleccionamos la bpl, ponemos alinear y pulsamos Volcar.

Le damos un nombre por ejemplo Comun_guan.dll y lo cargamos en el LordPE

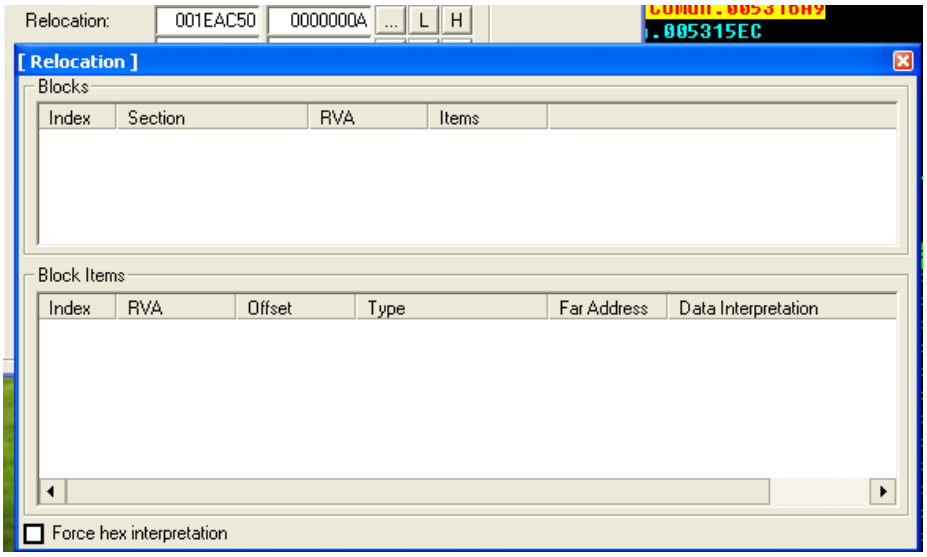


El número de secciones es el problema de que LordPe no pudiera dumpear, en este caso según he contado en Olly son 7. Reparamos esta parte de la cabecera así



Como nota importate es el cambio de la imagen base a la que tiene la dll en el momento del dump.

Di miramos el resto de la cabecera lo más raro es esto:

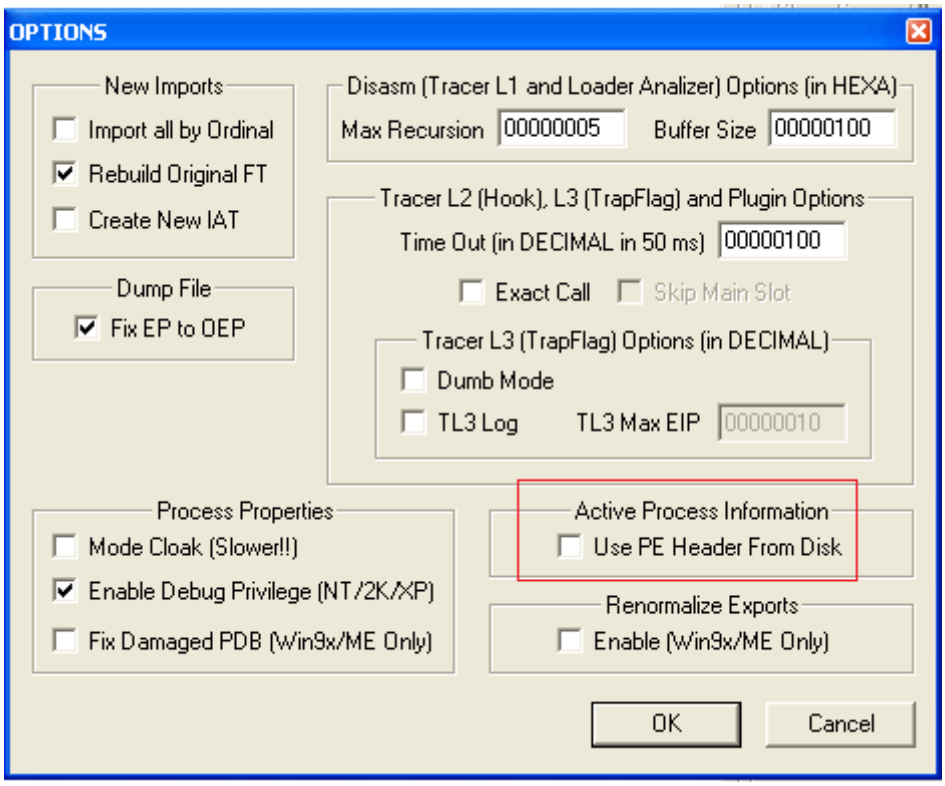


La sección Reloc se encuentra mal apuntada y antes del dump estaba igual.
Por precaucion podemos quitar esto, pero dejaro como está es igual.

Este punto es complejo para una dll, ya que si no se puede cargar en memoria en la dirección apuntada por la Imagen Base se hace uso de esta sección para repar la dll por parte del SO para que todo apunte a donde debe.

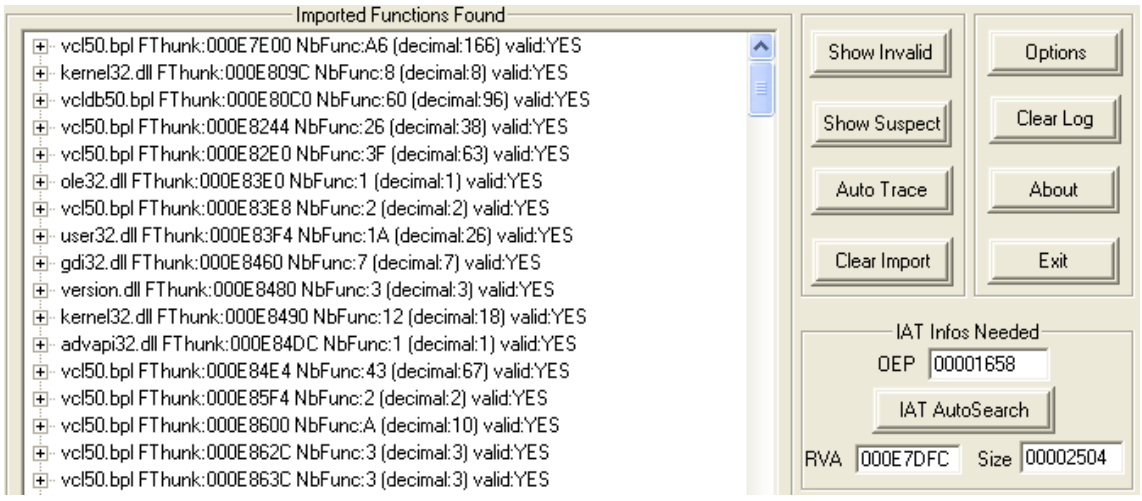
Reparando la IAT

Bueno en este PC Guard la IAT está perfecta solo hay que hacer una puntualización a la hora de usar el ImporRect y es en su apartado de opciones.



Es importante que esa casilla esté desmarcada

Por lo demás es lo desiempre, seleccionamos figaro.exe damos a Pick Dll y buscamos el comunes.



Ya con esto hacemos una copia de seguridad del comun.bpl, y a nuestro dump le cambiamos el nombre y lo ponemos en su lugar.

Si ejecutamos el programa todo parece ir correcto.

Lo siguientes es repetir los mismos pasos con TodoPais.bpl son exactamente los mismo así que os lo dejo como ejercicio.

Reparando la sección Reloc

Una vez ya con los 2 bpls arrancamos el programa y este no va ni en pintura, error de acceso tipo C00000005.

Lo gracioso es que si dejo un bpl unpacked y el otro original todo va bien, es cuando ponemos los 2 cuando se fastidia el invento.

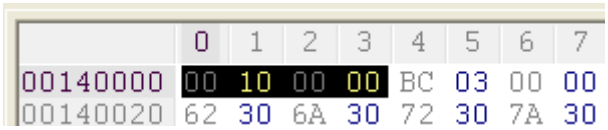
Si recordamos dijimos que faltaba la sección reloc, vamos a buscarla.

Lo que vamos a hacer es ver cómo está compuesta una sección de estas de un bpl correcto que tenga por aquí.

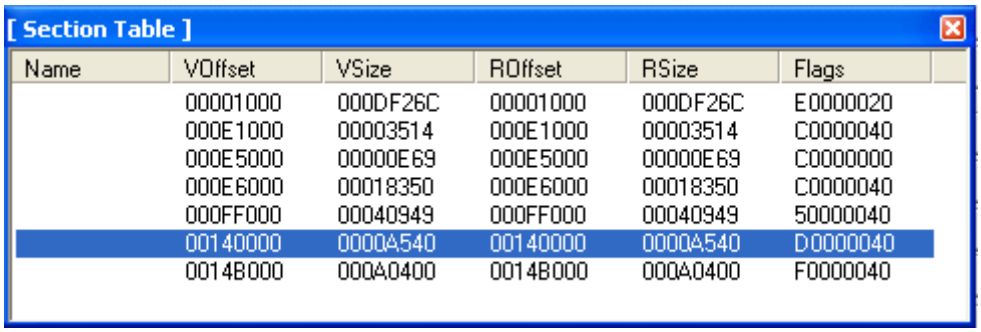
```
00006400:  00 10 00 00 0C 02 00 00
00006410:  EF 36 F5 36 0E 37 1A 37
00006420:  08 37 2C 37 3B 37 46 37
```

Los primeros bytes indica el comienzo de la sección, en este caso la sección code. Nuestra sección code también empieza ahí y sería muy raro que no se tuviera que modificar.

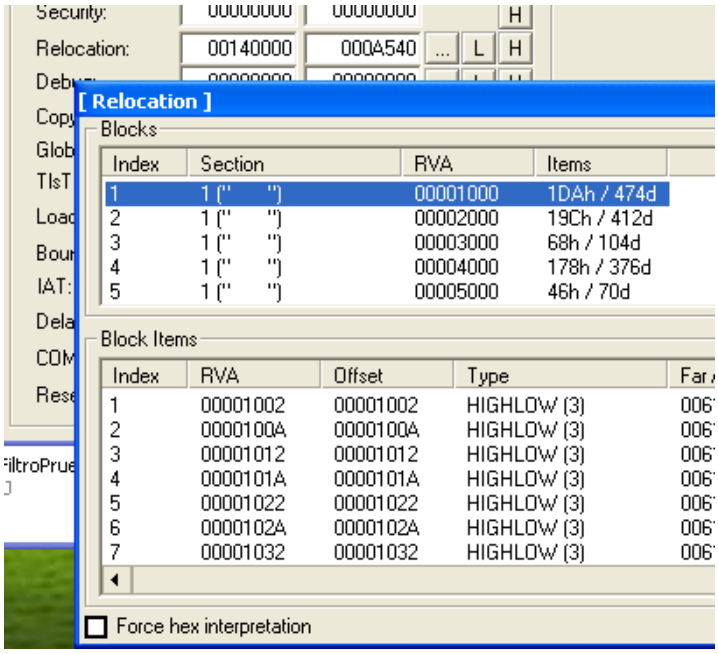
Metemos el comun_guan_.dll en el Hew Work Shop y buscamos la secuenca 00100000. Debe de ser el comienzo del bluque.



Esta es la dirección que más me gusta, sobre todo porque coincide con el comienzo de sección, probemos.



Y la modificación



Si señor aquí tenemos la sección Reloc corregida.

Al parecer en los Borland suele ser la sección anterior a la de recursos. El tamaño como no estamos seguros le damos toda la sección.

Haciendo lo mismo con la otra bpl, le volvemos a cambiar los nombre y ya si ahora todo corre como la seda por lo que damos por concluido este tutial.

Agradecimientos

Como no a toda la pandilla de CLS, Ricardo, Solid, Tena, Shaddy, Aboslom1, NCR, y un largo etc, y sobre todo a ti por a ver llegado hasta aquí.

Hasta la próxima.