Programa	SUISkin (Componentes para Delphi)
Versión	4.46
Herramientas	Ollydbg+IDA
Compilador	Delphi
Objetivos	Eliminar la nag
Cracker	Spandau

Si estás perdido en un tunel oscuro y de repente ves una luz al fondo, en el 99% de los casos será la del tren que se te viene encima, este es el relato de una de las pocas ocasiones en que la luz es la salida del tunel.

SuiSkin es un componente para dotar a Delphi de la posibilidad de incluir Skins de forma sencilla, tiene un bug al poner el título de form pero por lo demás están bastante bién. La web del sitio es: http://www.sunisoft.com/suiskin/.

Una vez instaladas, me creo un proyecto en delphi, tiro el componente en el form, pongo un skin en la propiedad skinfile, y pongo a true las propiedas Active y BuiltIn.Guardamos el proyecy, F9 y me regala este cartelito:

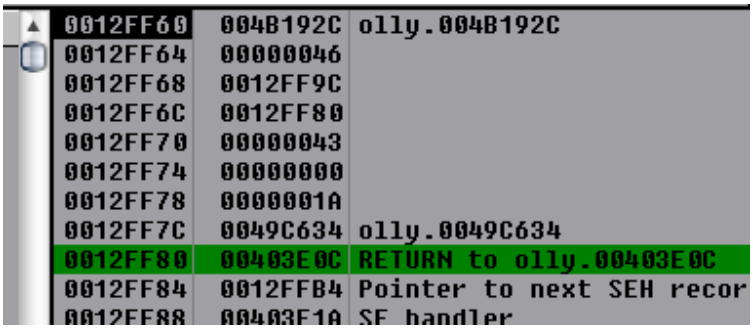
Ya tengo el exe, cierro delphi y lo ejecuto, ahora me regala no una sino dos nag, vamos bién.

Lo cargo en Olly, pongo un bp MessageBoxA, F9 y veo como me tira a la basura al momento, por lo visto me va a dar más guerra de la que pensaba.

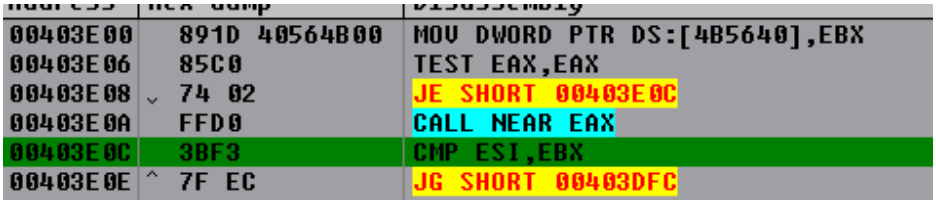
Borro el bp, reinicio F9 y cuando tengo la nag en la pantalla F12 y pauso el programa, ahora execute till user code (Alt+F9) acepto la nag y caigo en medio de la mierda:

Address	Hex dump	Disassembl
0049E3AA	EB	DB EB
0049E3AB	0C	DB 0C
0049E3AC	90	DB 90
0049E3AD	90	DB 90
0049E3AE	90	DB 90
0049E3AF	6A	DB 6A
0049E3B0	00	DB 00
0049E3B1	FF	DB FF
0049E3B2	95	DB 95
0049E3B3	D5	DB D5
0049E3B4	1B	DB 1B
0049E3B5	40	DB 40

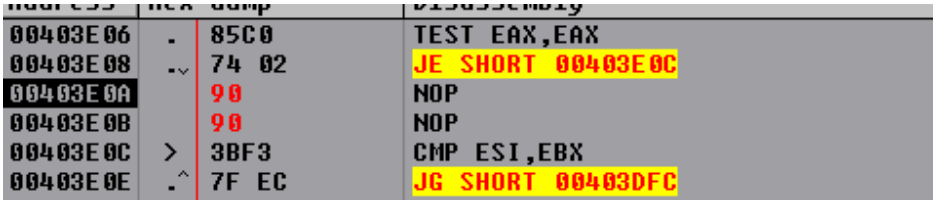
Es evidente que esto ha sido desencriptado hace un momento. Botón derecho Analysis->Remove Analysis from module traceo hasta pasar el retn y busco en la pila y un poco más abajo veo de donde se llamó a la nag



Pincho con el botón derecho, Follow in disassembler y caigo aquí:



Pues justo encima tengo la llamada misteriosa.Ahora si nopeo la call deberíamos quitar la nag, voy a probar. Pongo un bp en 403E0A reinicio y para sin problemas, barra espaciadora nopeo y quito el bp



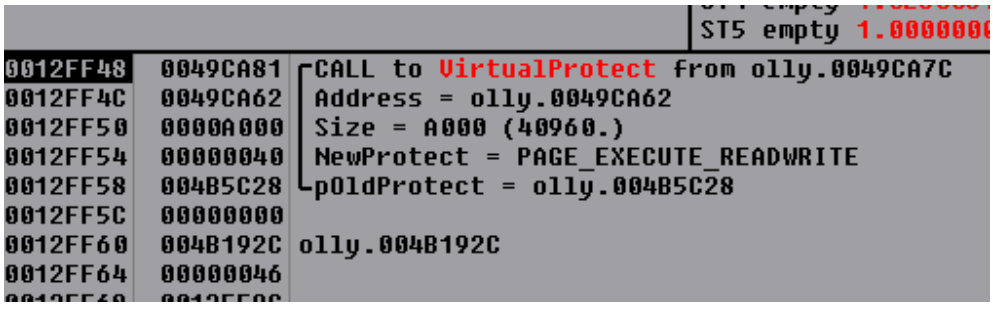
F9 y:

Mierda!!!!!!!!, era demasiado fácil, (olly.exe es el proyect1.exe renombrado).

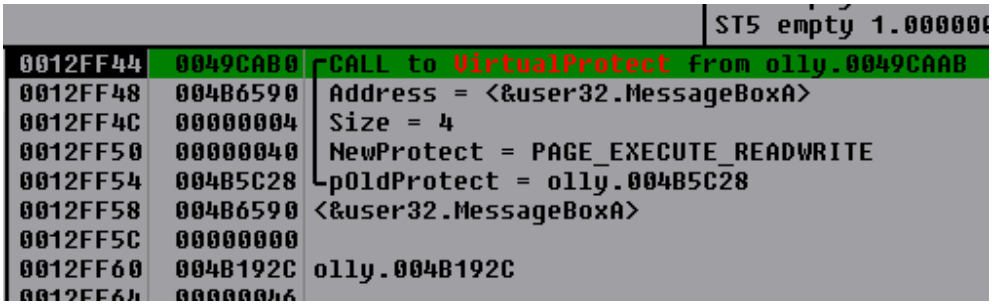
Reinicio y vuelvo a poner el bp en 403E0A, ejecuto y para, paso con F7, F9 y f7de nuevo y puedo seguir así hasta hacer viejo, para en distintas funciones y no saco nada en claro.

Vemos: tengo una función empaquetada que se carga en ejecución, debería dumpearla y luego anular las llamadas a empacar y desempacar, claro que esto no vale para nada si no soy capaz de pasarlo al DCU correspondiente. Pues debo reconocer que no tengo ni idea de cómo hacerlo, es hora de dejar todo esto de cracking y pirarme al plantar cebollinos, pero antes voy a enredar un poco más a ver que pasa.

El exe no tiene permisos de escritura en la seccion CODE, así que en algún sitio debe darselos, quito los bp que tengo puestos y añado uno en VirtualProtect a ver que pasa, reinicio y veo que da permiso de escritura a 49CA62

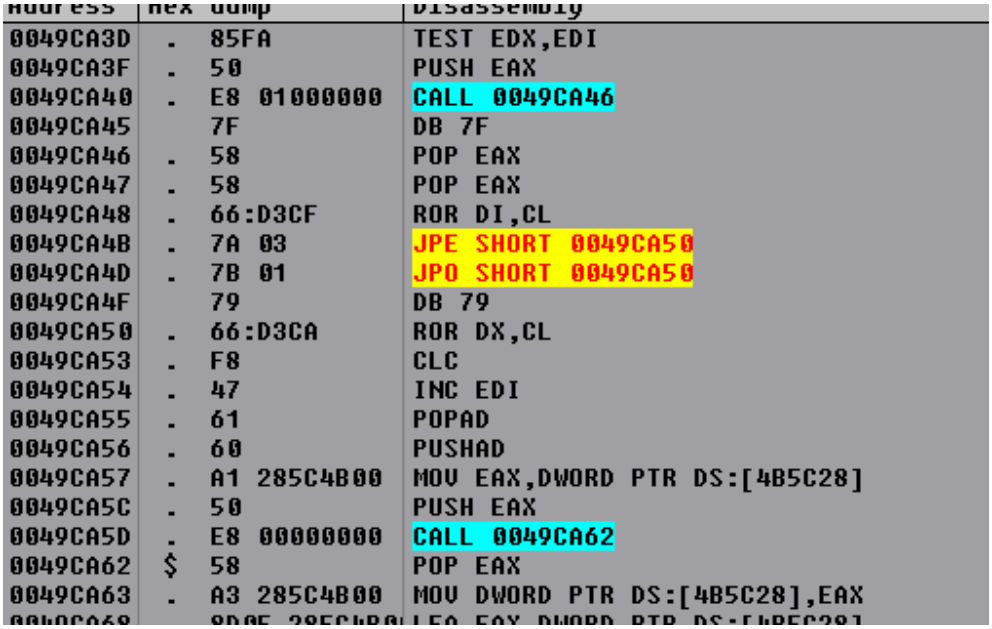


Por curiosidad pulso F9 a ver si para más veces

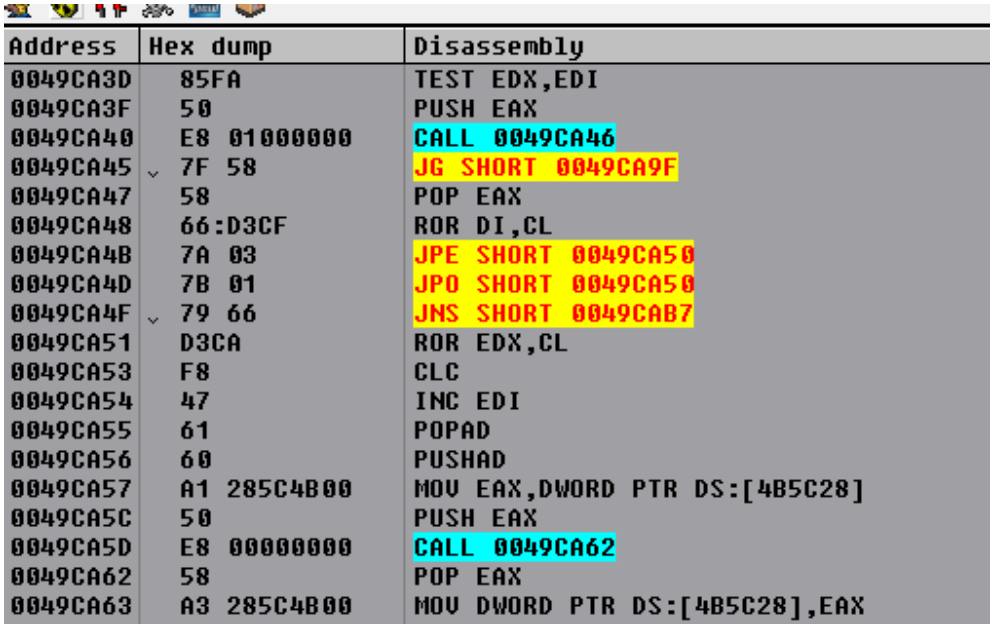


¡Vaya! También da permisos de escritura a MessageBoxA, por eso nos tiraba a la basura al poner un bp alli (si os parais a mirar en realidad no cambia nada de MsgBox, pero debe jugar con los permisos para ver si hemos toqueteado en ella).

Bueno, reinicio y parado en la primera llamada a VirtualProtect veo que se llama desde 49CA7C, asi que me voy alli. Un poco más abajo está la otra llamada a VirtualProtect, así que cre que voy por buén camino, subo un poco y:



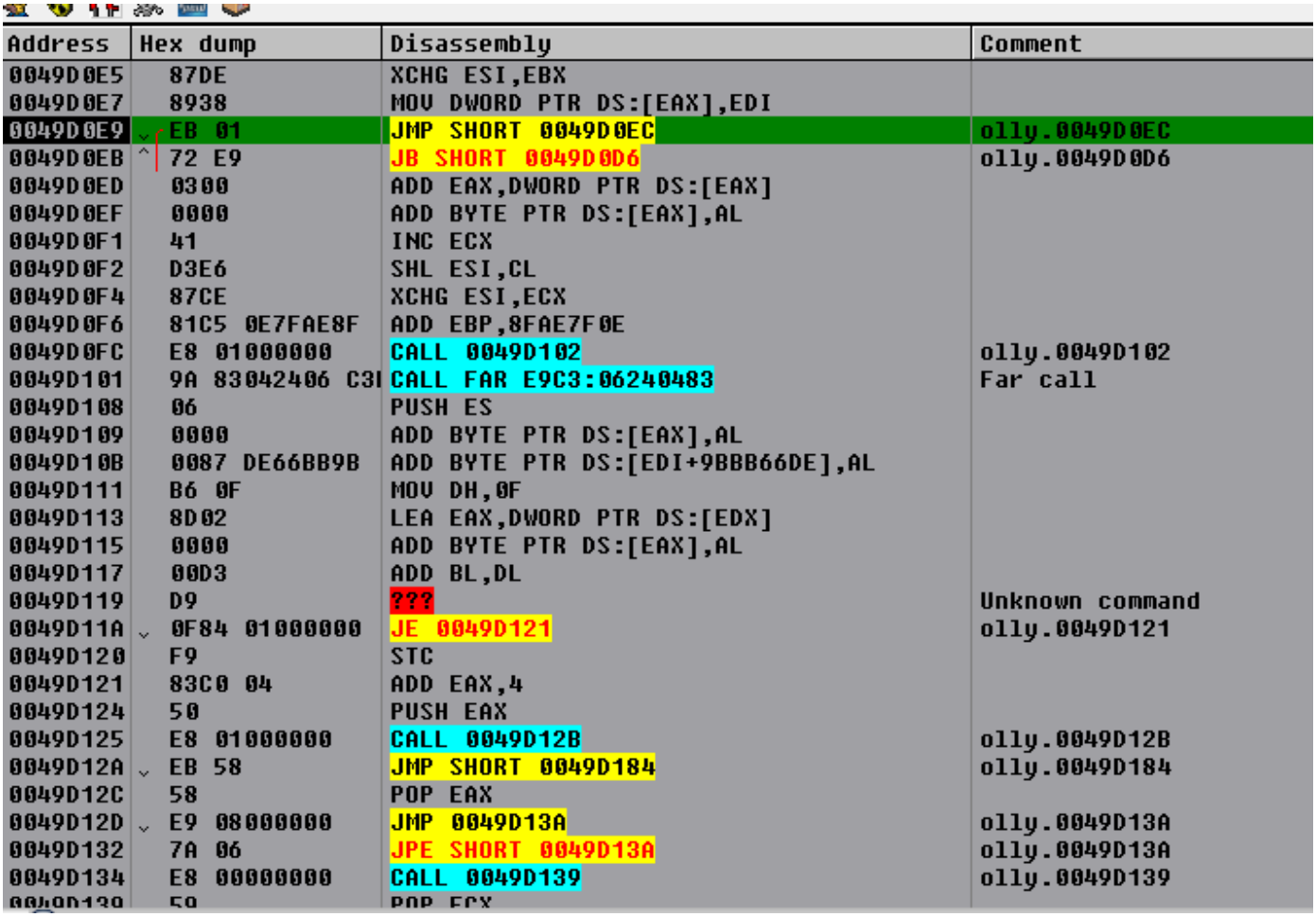
Esto se ve bastante feo, asi que quito el analisis y



Address	Hex dump	Disassembly
0049CA3D	85FA	TEST EDX,EDI
0049CA3F	50	PUSH EAX
0049CA40	E8 01000000	CALL 0049CA46
0049CA45	7F 58	JG SHORT 0049CA9F
0049CA47	58	POP EAX
0049CA48	66:D3CF	ROR DI,CL
0049CA4B	7A 03	JPE SHORT 0049CA50
0049CA4D	7B 01	JPO SHORT 0049CA50
0049CA4F	79 66	JNS SHORT 0049CAB7
0049CA51	D3CA	ROR EDX,CL
0049CA53	F8	CLC
0049CA54	47	INC EDI
0049CA55	61	POPAD
0049CA56	60	PUSHAD
0049CA57	A1 285C4B00	MOV EAX,DWORD PTR DS:[4B5C28]
0049CA5C	50	PUSH EAX
0049CA5D	E8 00000000	CALL 0049CA62
0049CA62	58	POP EAX
0049CA63	A3 285C4B00	MOV DWORD PTR DS:[4B5C28],EAX

Más feo todavía, vamos de mal en peor. El codigo está ofuscado, esto nos dice que esta es una zona muy caliente, pero despues de tres cajas de aspinas no veo nada en claro.

Si ponemos en bp en el retn 10 de MessageBoxA, vemos que no nos tira (la primera vez me tiró por no hacer caso a Ricardo Narvaja, los bp en las APIS, nunca hay que ponerlos en la primera instrucción), y quitamos todos los demás, al reiniciar cuando para F7 y caemos en pleno territorio comanche, podeis poner un hbp de escritura en cualquier sitio de esa zona y ver donde desempaqueta, pero es codigo está super ofuscado y da dolor de cabeza aquí os pongo una imagen:

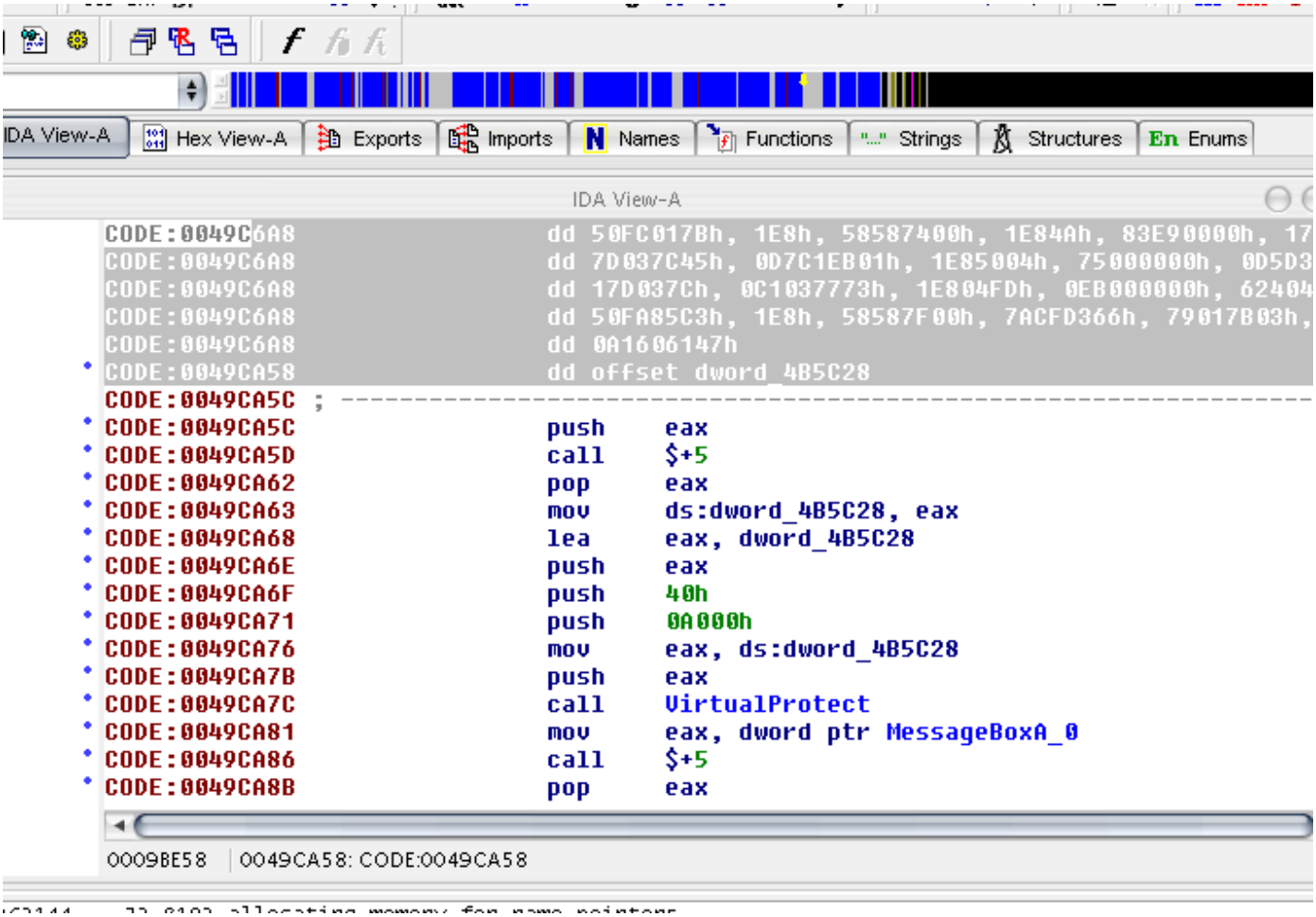


Address	Hex dump	Disassembly	Comment
0049D0E5	87DE	XCHG ESI,EBX	
0049D0E7	8938	MOV DWORD PTR DS:[EAX],EDI	
0049D0E9	EB 01	JMP SHORT 0049D0EC	olly.0049D0EC
0049D0EB	72 E9	JB SHORT 0049D0D6	olly.0049D0D6
0049D0ED	0300	ADD EAX,DWORD PTR DS:[EAX]	
0049D0EF	0000	ADD BYTE PTR DS:[EAX],AL	
0049D0F1	41	INC ECX	
0049D0F2	D3E6	SHL ESI,CL	
0049D0F4	87CE	XCHG ESI,ECX	
0049D0F6	81C5 0E7FAE8F	ADD EBP,8FAE7F0E	
0049D0FC	E8 01000000	CALL 0049D102	olly.0049D102
0049D101	9A 83042406 C3I	CALL FAR E9C3:06240483	Far call
0049D108	06	PUSH ES	
0049D109	0000	ADD BYTE PTR DS:[EAX],AL	
0049D10B	0087 DE66BB9B	ADD BYTE PTR DS:[EDI+9BBB66DE],AL	
0049D111	B6 0F	MOV DH,0F	
0049D113	8D02	LEA EAX,DWORD PTR DS:[EDX]	
0049D115	0000	ADD BYTE PTR DS:[EAX],AL	
0049D117	00D3	ADD BL,DL	
0049D119	D9	???	Unknown command
0049D11A	0F84 01000000	JE 0049D121	olly.0049D121
0049D120	F9	STC	
0049D121	83C0 04	ADD EAX,4	
0049D124	50	PUSH EAX	
0049D125	E8 01000000	CALL 0049D12B	olly.0049D12B
0049D12A	EB 58	JMP SHORT 0049D184	olly.0049D184
0049D12C	58	POP EAX	
0049D12D	E9 08000000	JMP 0049D13A	olly.0049D13A
0049D132	7A 06	JPE SHORT 0049D13A	olly.0049D13A
0049D134	E8 00000000	CALL 0049D139	olly.0049D139
0049D139	59	POP ECX	

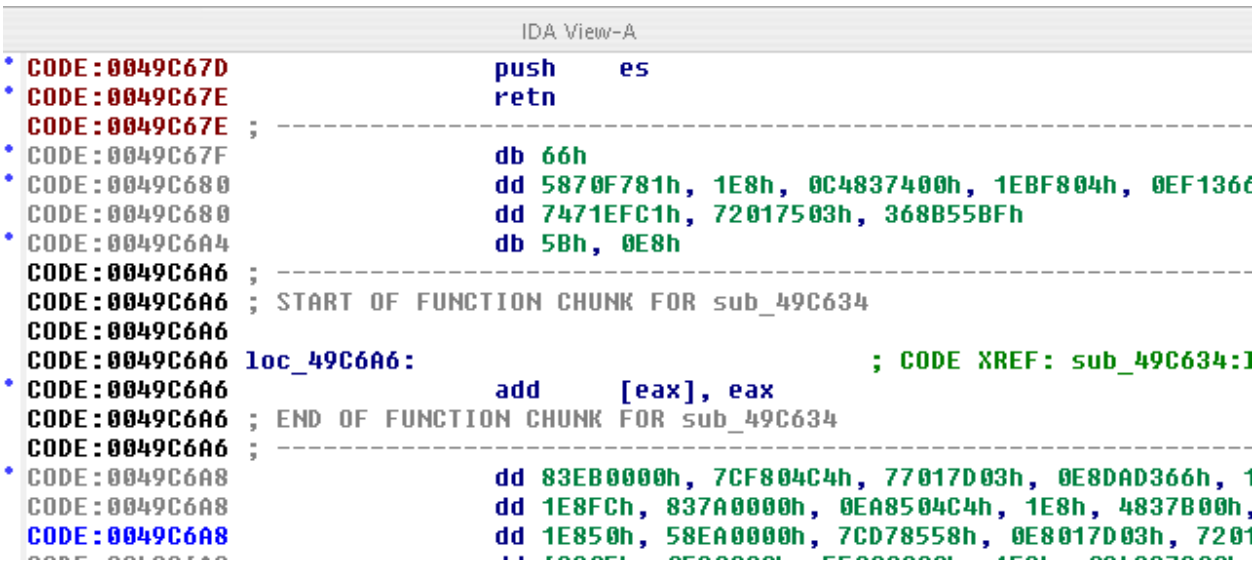
Si os fijais casi todos los call y saltos van a la mitad de otra instrucción, bueno pues yo me rindo aquí, pero antes le voy a echar una mira con el IDA, porque no lo controlo y estoy a ver si voy soltandome con el.

Hago dos copias del progri, a una la llamo olly.exe y a otra IDA.exe

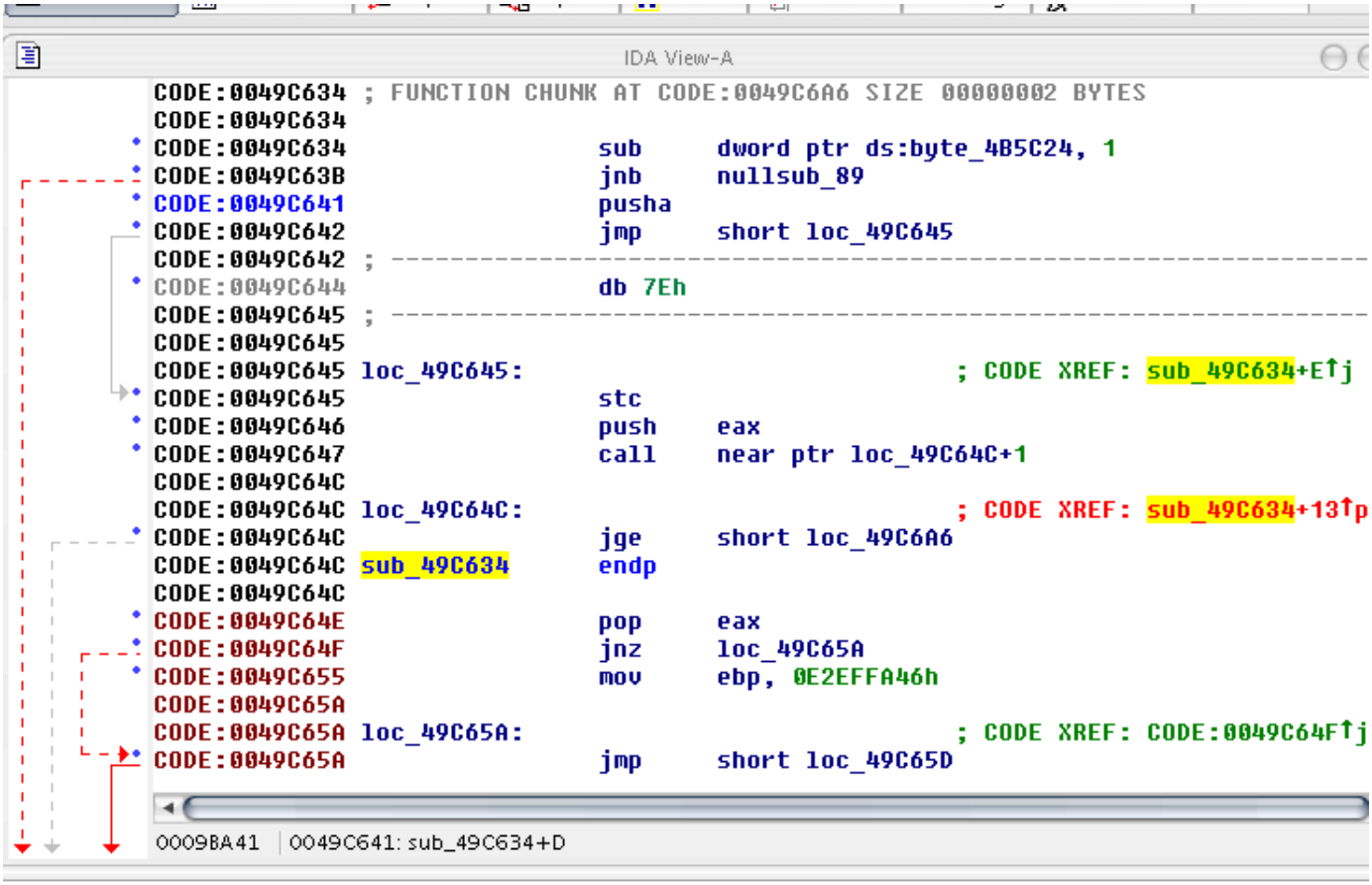
Abro IDA.EXE en el ida y pulso G para saltar a 49CA7C que era la llamada a VirtualProtect



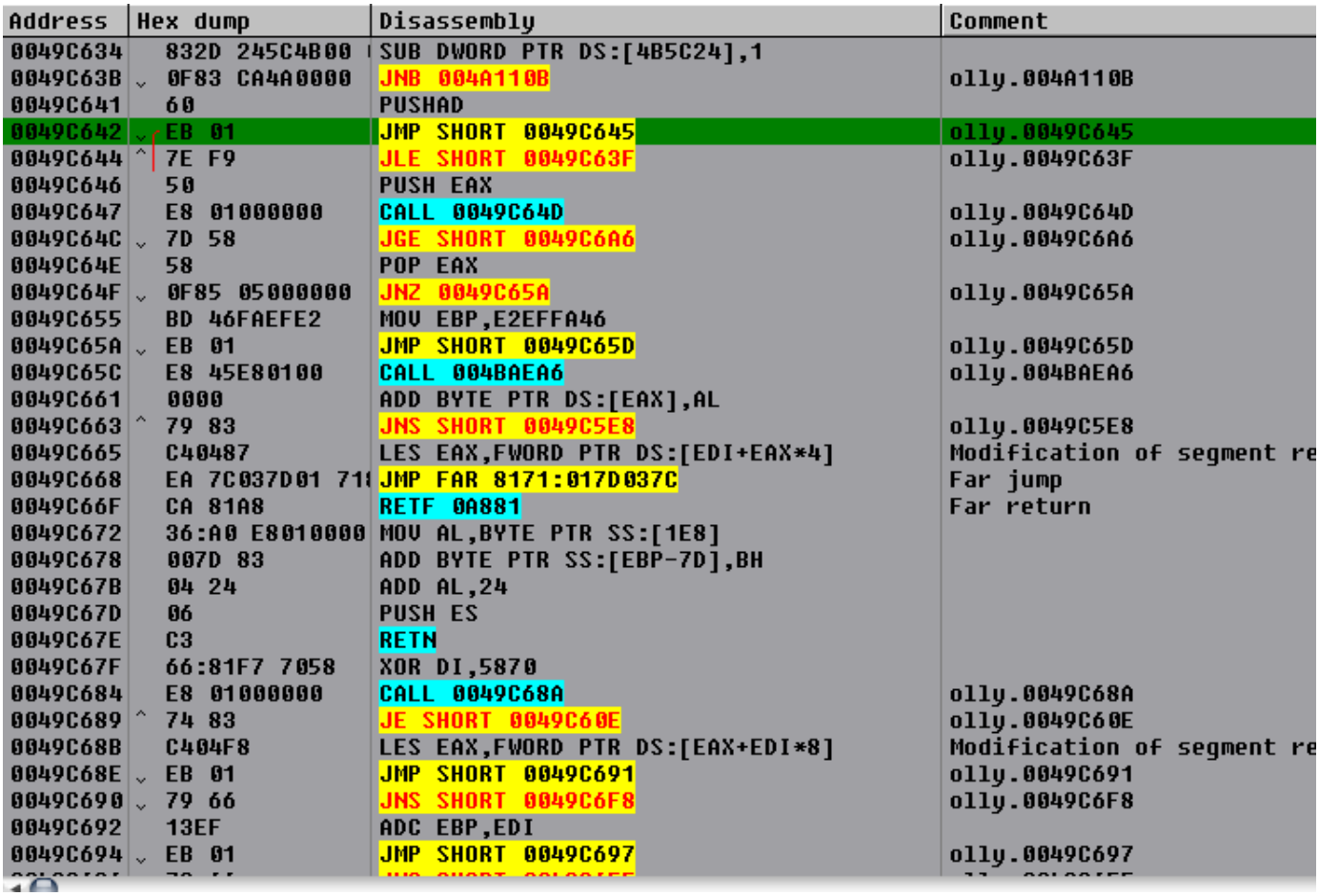
Si nos fijamos algo mas arriba hay una zona (la he marcado en gris) llena de basura, que es la parte ofuscada que vimos en olly, pues sigo subiendo y me encuentro con



Me pegunto que es lo que pasará en 49C634 para que IDA le preste tanta atención, asi que voy alli a ver que pasa (doble click en sub_49C634) y caemos aquí



En olly:



Address	Hex dump	Disassembly	Comment
0049C634	832D 245C4B00	SUB DWORD PTR DS:[4B5C24],1	
0049C63B	0F83 CA4A0000	JNB 004A110B	olly.004A110B
0049C641	60	PUSHAD	
0049C642	EB 01	JMP SHORT 0049C645	olly.0049C645
0049C644	7E F9	JLE SHORT 0049C63F	olly.0049C63F
0049C646	50	PUSH EAX	
0049C647	E8 01000000	CALL 0049C64D	olly.0049C64D
0049C64C	7D 58	JGE SHORT 0049C6A6	olly.0049C6A6
0049C64E	58	POP EAX	
0049C64F	0F85 05000000	JNZ 0049C65A	olly.0049C65A
0049C655	BD 46FAEFE2	MOV EBP,E2EFFA46	
0049C65A	EB 01	JMP SHORT 0049C65D	olly.0049C65D
0049C65C	E8 45E80100	CALL 004BAEA6	olly.004BAEA6
0049C661	0000	ADD BYTE PTR DS:[EAX],AL	
0049C663	79 83	JNS SHORT 0049C5E8	olly.0049C5E8
0049C665	C40487	LES EAX,FWORD PTR DS:[EDI+EAX*4]	Modification of segment re
0049C668	EA 7C037D01 71	JMP FAR 8171:017D037C	Far jump
0049C66F	CA 81A8	RETF 0A881	Far return
0049C672	36:A0 E8010000	MOV AL,BYTE PTR SS:[1E8]	
0049C678	007D 83	ADD BYTE PTR SS:[EBP-7D],BH	
0049C67B	04 24	ADD AL,24	
0049C67D	06	PUSH ES	
0049C67E	C3	RETN	
0049C67F	66:81F7 7058	XOR DI,5870	
0049C684	E8 01000000	CALL 0049C68A	olly.0049C68A
0049C689	74 83	JE SHORT 0049C60E	olly.0049C60E
0049C68B	C404F8	LES EAX,FWORD PTR DS:[EAX+EDI*8]	Modification of segment re
0049C68E	EB 01	JMP SHORT 0049C691	olly.0049C691
0049C690	79 66	JNS SHORT 0049C6F8	olly.0049C6F8
0049C692	13EF	ADC EBP,EDI	
0049C694	EB 01	JMP SHORT 0049C697	olly.0049C697

Como vemos está todo bastante ofuscado, pero en IDA se ve algo mejor (al menos las primeras lineas)

0049C632 MOV EAX,EAX

0049C634 SUB DWORD PTR DS:[4B5C24],1

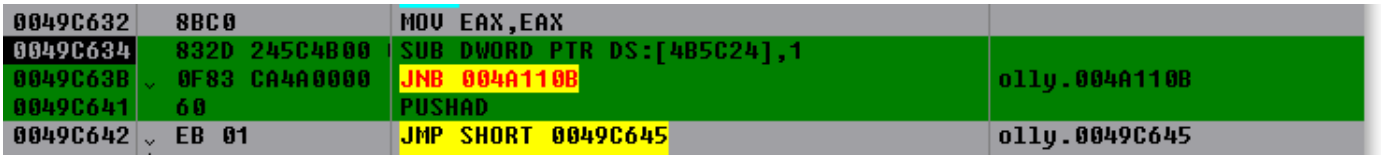
0049C63B JNB 004A110B ; olly.004A110B

0049C641 PUSHAD

Vemos que el salto 0049C63B nos saca de la zona de guerra (el pushad nos dice que empiezan tareas delicadas). Después de tanto encriptamiento y ofuscación no parece probable que este simple salto resuelva nada, pero vamos o intentarlo, en olly ponemos un bp en 49C63B reiniciamos y cuando para pulsamos la barra espaciadora y cambiamos el JNB por un JMP (seguro que hay

que quitar el analisis para ver algo), probamos y que ya no sale ninguna nag que rula todo bién, bingo.

Volvemos a reiniciar y paramos de nuevo en 0049C63B, ahora vamos a marcar tres instrucciones:



las usaremos como firma, botón derecho, bynari copy y lo pegamos en el bloc de notas

83 2D 24 5C 4B 00 01 0F 83 CA 4A 00 00 60

os dígitos marcados en rojo corresponden a la dirección de memoria de la variable por lo tanto en el DCU, estarán a cero, la firma sería entonces:

83 2D 00 00 00 00 01 0F 83 CA 4A 00 00 60

Ahora solo tenemos que abrir en un editor hexadecimal yo utilizo el pspad, en él abrimos todos los dcu (en mi caso están en e:\suiSkin\source\delphi7) y buscamos la firma en todos los archivos, la encuentra en suiskinusing.dcu, ahora solo ue cambiar el salto y listo

0F 83 CA 4A 00 00 por E9 CB 4A 00 00, pero como el salto nos lleva directos a un retn también podemos cambiar el pushad (60) por retn (C3), asi que en el editor cambiamos en el offset 02b9 el 60 por un C3, salvamos y ya está, si abrimos el proyecto en delphi y lo corremos ya no salta la nag.

Como curiosidad si abrimos delphi en olly si aparece la nag, de alguna forma lo detecta (he visto llamadas a rdtsc,).

Dedicado a Arapumk que me enseñó a crackear componentes delphi, gracias por tu proyecto Delphi y sobre todo por tu paciencia.

Recuerdos a todos los CrackSLatinos, y en especial a Ricardo para que se mejore de su gripe.