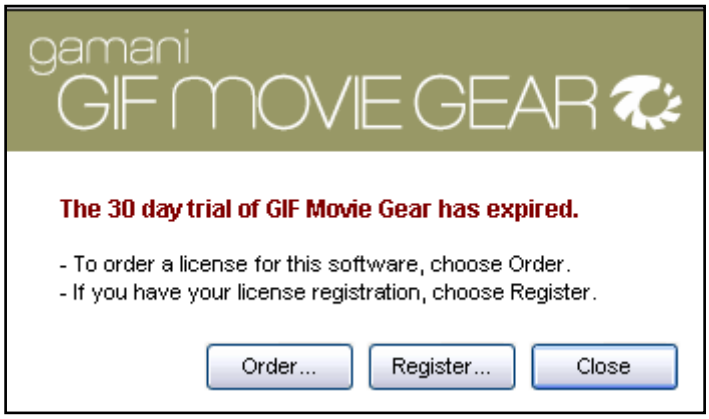
Programa	**GIF MOVIE GEAR**
Download	www.gamani.com
Descripción	Programa para crear gif´s.
Herramientas	OllyDbg 1.10
Dificultad	Newbie
Compilador	Microsoft Visual C ++ v 7.0
Protección	Serial/Name
Objetivos	Parchar y dejarlo full
Cracker(?)…Newbie	**BioHaZarD** Fecha: 23/07/09
Tutorial nº	1

Antes que nada PERDÓN!!! Je! Viendo todos los tutos espectaculares de los listeros me da vergüenza escribir esto…pero bue como lei en un post viejo del grupo de google que Ricardo decía que lo que escribamos siempre le sirve a alguien…hací que bueno ahí voy. Espero esto le sirva a algún Newbie como yo…

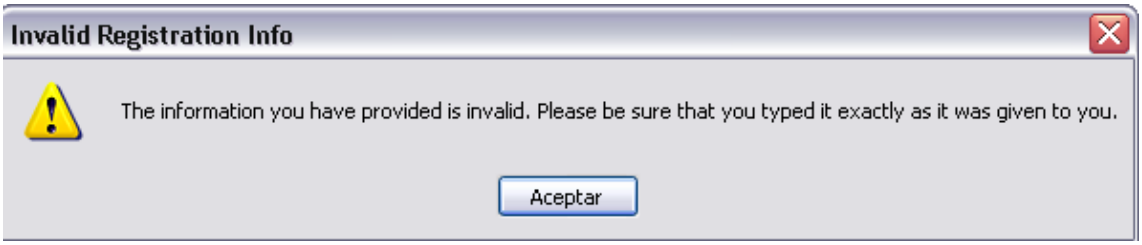
EMPEZANDO:

El programa es full por 30 días, tenemos la opción de registrar buscando en "HELP-REGISTER NOW…" en el período trial pero transcurrido este aparece la siguiente nag:

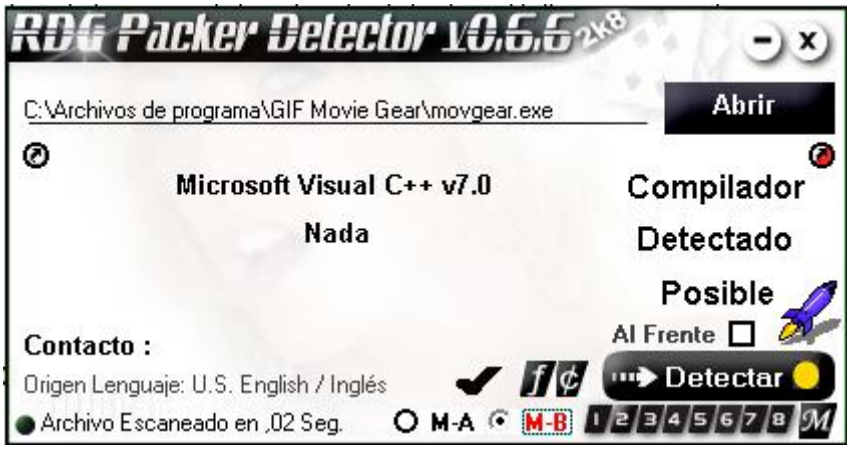


Y luego si pinchamos el botón “Register…” nos aparece una nueva ventana para ingresar un nombre y un serial:

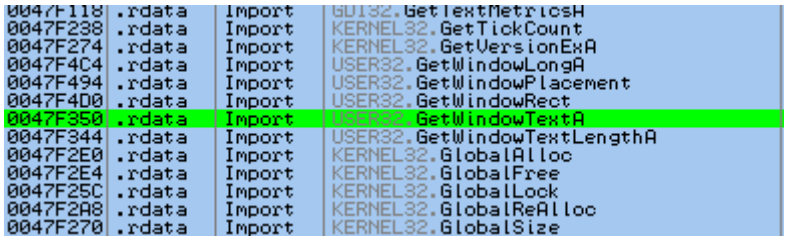
Ingresando un serial cualquiera…



El amigo RDG Packer detector nos da buenas noticia s(por lo menos para mí que soy un Newbie…):

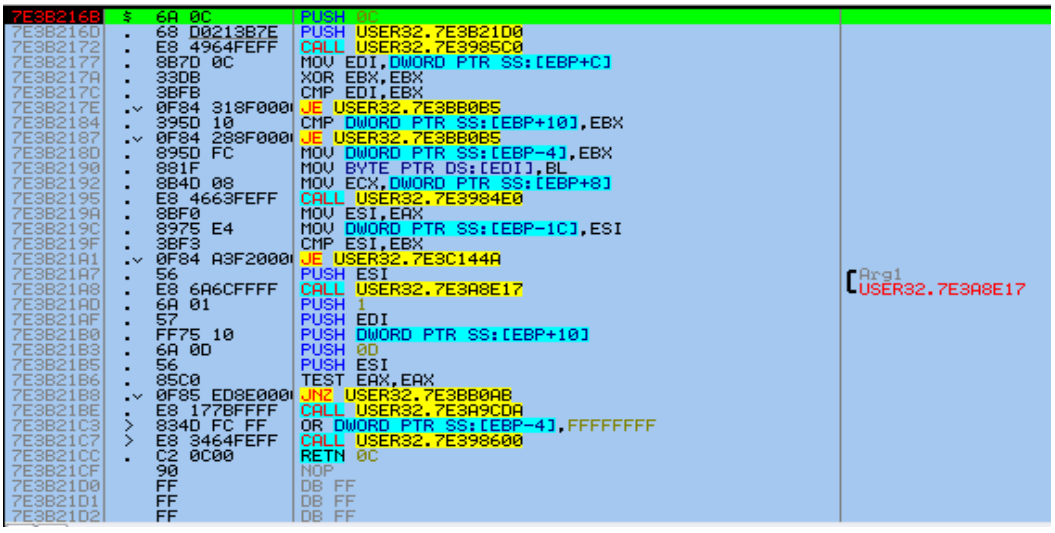


Lo cargamos en Olly y con Ctrl+N (o click con el botón derecho del mouse y…SEARCH FOR/NAME (label) IN CURRENT MODULE) buscamos alguna Api conocida en la lista:

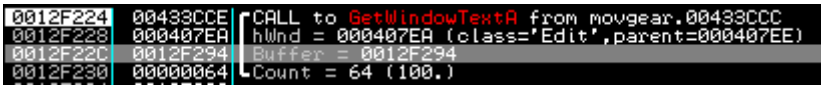


Allí vemos la api GetWindowTextA que utilizada para ingresar texto, le ponemos un Breakpoint (BP) con F2 y con F9 comienza a correr el programa y en la nag inicial pinchamos en “Register…” y en la que sigue ingresamos un nombre y un serial cualquiera:

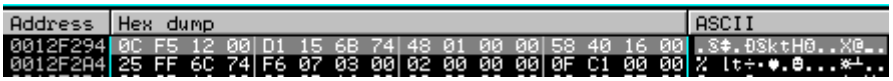
Olly para aquí,dentro de USER32:



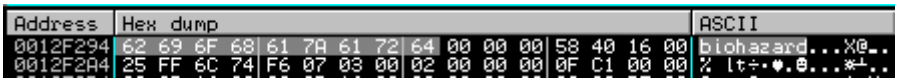
Si miramos el Stack:



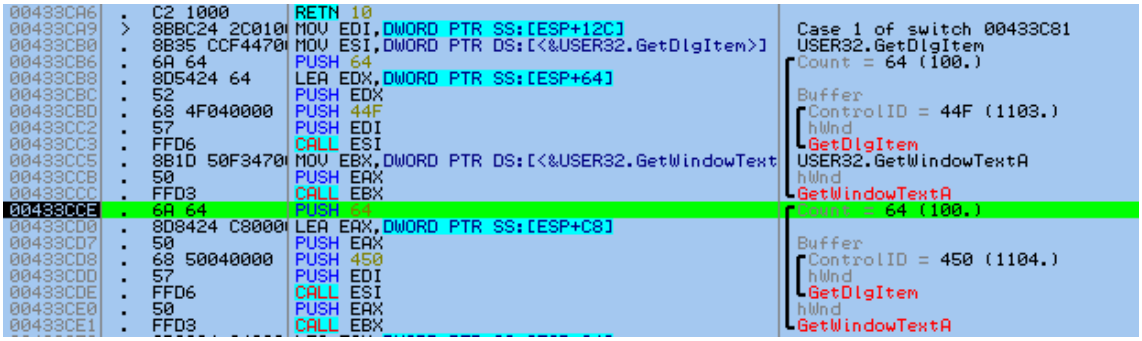
Hacemos click con el botón derecho en el Buffer y FOLLOW IN DUMP que por ahora esta haci:



Con un EXECUTE TILL RETURN(Ctrl+F9) ejecutamos el codigo hasta el "Retn" que nos va a devolver al programa, si miramos el Stack en la misma posición que recién,vemos que ahora ingreso allí el nombre que pusimos en la ventana de registro:

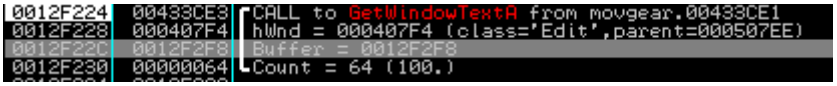


Parados en "Retn" con F8 o F7 volvemos a la siguiente instrucción a ejecutarse del módulo "movgear":

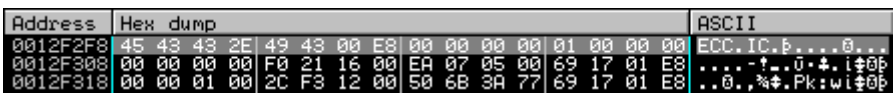


Parados aquí veamos algo: nosotros volvemos en ese PUSH 64 y mas abajo hay dos CALL, la primera llama a GetDlgItem y la segunda es una llamada a la api GetWindowTextA y si miramos arriba de, PUSH 64 vemos que hay otro CALL a dicha api; es en esa en la que paro el Breakpoint que pusimos nosotros,y en ella se ingreso el nombre trucho en el Dump, entonces en esta segunda llamada a WindowGetTextA lo que va a hacer es ingresar nuestro serial trucho. Si entramos con F7 a "CALL EBX" volvemos a tener un buffer en el Stack y procediendo de igual manera que con el nombre trucho:

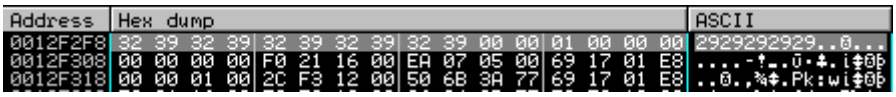
En el Stack:



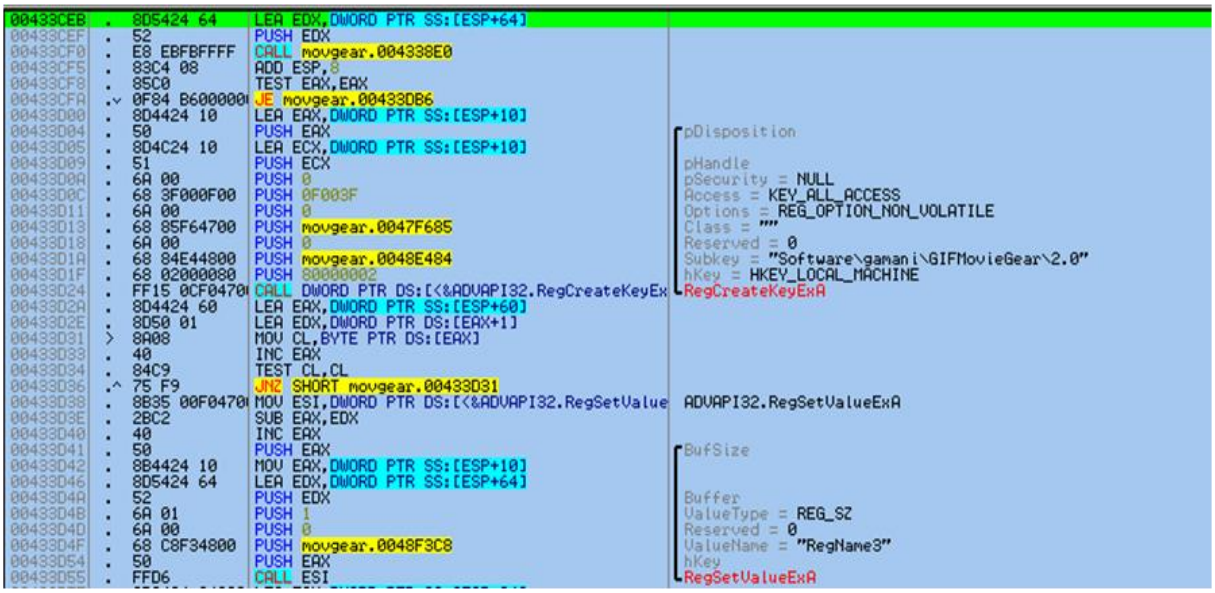
En el Dump:



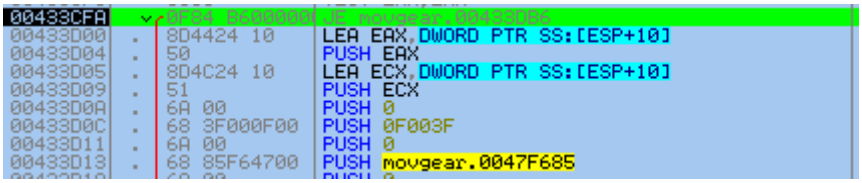
Luego de hacer un EXECUTE TILL RETURN (Ctrl+F9):



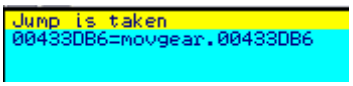
Con F8 volvemos al modulo"movgear" aqui:



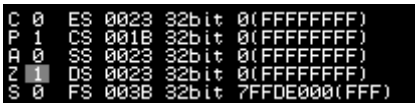
Nosotros estamos en LEA EDX, DWORD PTR SS:[ESP+64] y más abajo vemos llamadas a apis que trabajan con el Registro de Windows, Si miramos la que está más abajo en la imagen vemos que crea un valor con el nombre "RegName3" y un poco más abajo tenemos algo similar pero hay un "RegCode". Al principio de la imagen hay JE y más arriba una CALL., dentro de la cual "hace algo" de lo que después va a depender ese salto JE, si no salta guardara nuestros datos en el registro y supongo que los guardara solo si el serial ingresado es correcto. Podríamos probar si este es el salto clave; el JE verificara el estado del flag Z y si este vale 1 saltará y si vale 0 no lo hará.



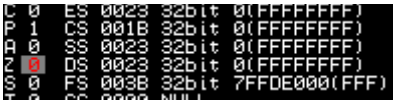
En la imagen de arriba estamos ya sobre el salto y si miramos bien vemos una línea roja que va hacia abajo; el color de la línea nos indica que el salto se tomará, igual Olly nos da esta información:



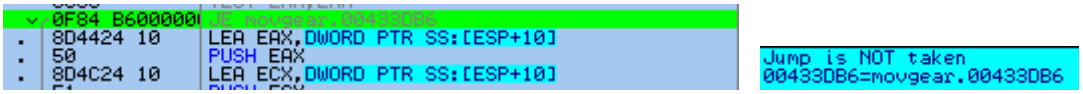
El estado del flag Z es el siguiente:



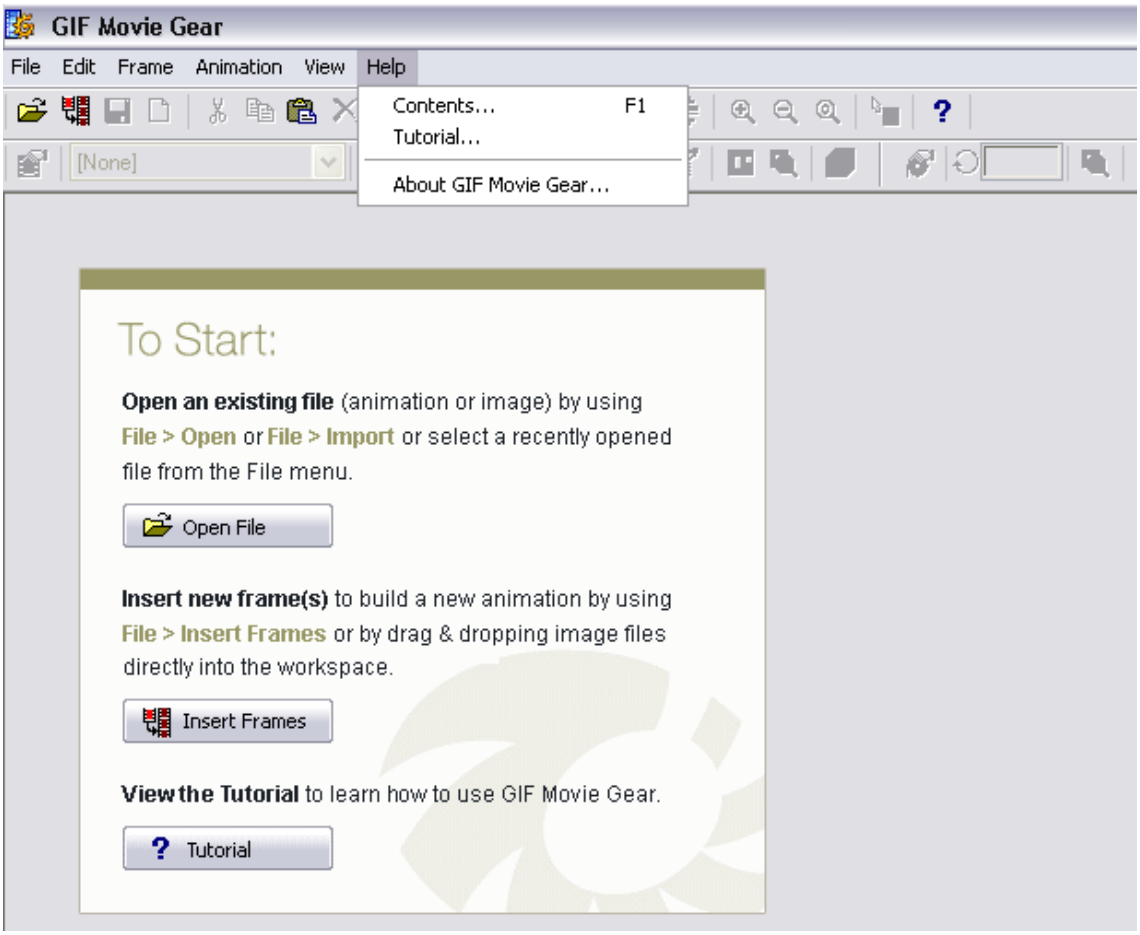
Con un doble click sobre el 1:



Ahí vemos que cambia el flag a 0 y ahora nuestro salto se ve haci:

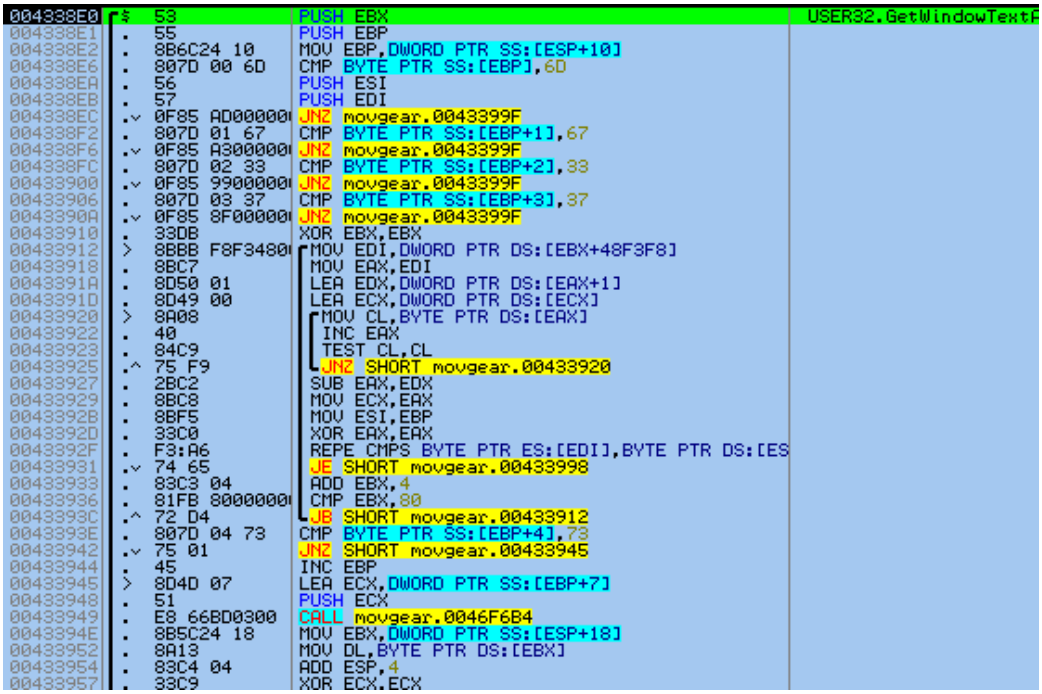


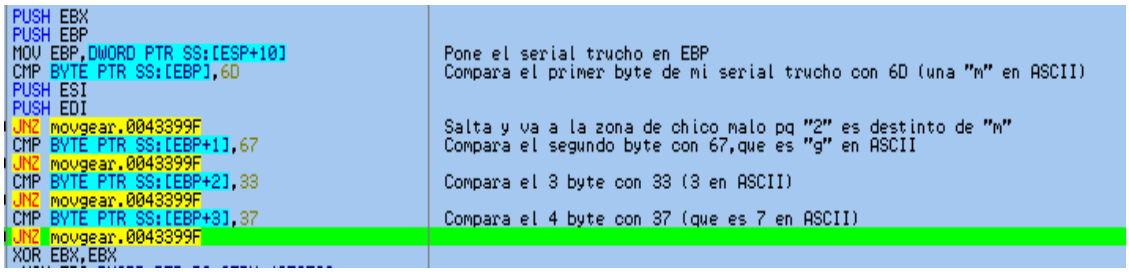
Este cambio no es permanente, se realiza en memoria, es para ver si este salto es el correcto. Pongamos un Breakpoint (F2) ya que si es nuestro salto deberemos volver aquí. Demos Run (F9) y veamos que pasa:



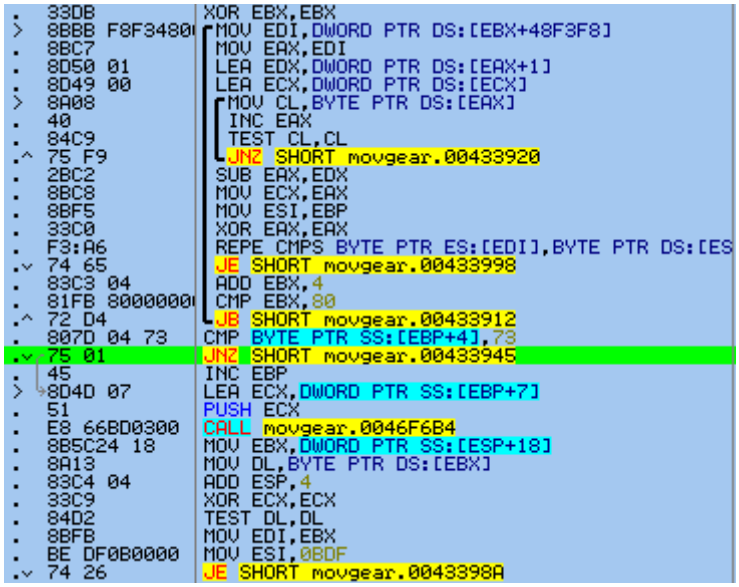
EL programa corre y en HELP ya no aparece la opción REGISTER NOW…
Sin duda este es nuestro salto! Carguemos de nuevo el programa (Alt + F2).
Bien ahora pensemos como parcharlo. Podriamos cambiar ese JE por un JNE para que cuando el flag Z vale 1 NO salte. Inicialmente pensé que esto funcionaría y de hecho funciona, pero en este caso tiene 2 contras: hay que poner el número de registro (jeje una re vagancia) y al final sale una nag diciendo "THIS TRIAL SOFTWARE EXPIRES IN -1 DAYS".

A ver si recordamos encima del salto había un CALL. Una llamada arriba de un salto clave me dice que en esa CALL se debe generar el serial y en base a ello el JE se produce o no. Situemonos sobre "CALL movgear.004338E0" y entremos con F7 a ver qué pasa. Una vez dentro vemos algo hací:

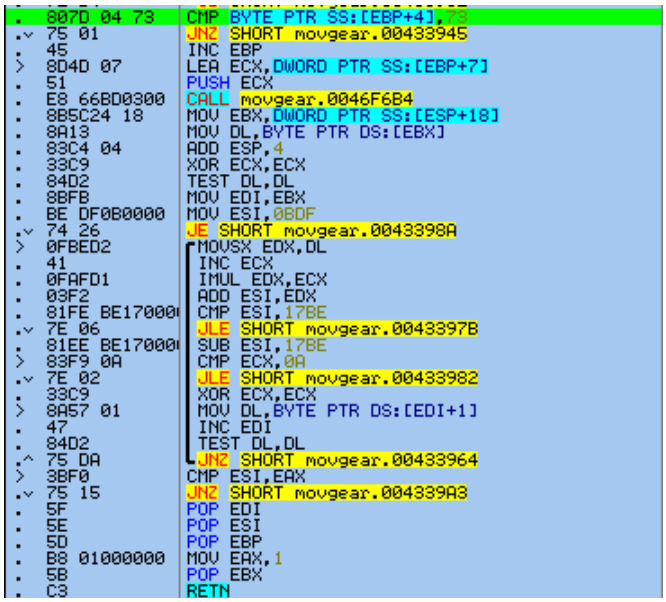




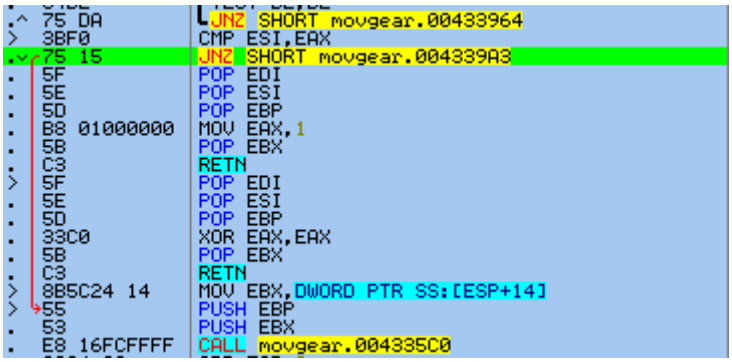
Puse algunos comentarios para que vean lo que hace. Todos esos JNZ me llevan a la zona de chico malo (zona donde se genera el MessageBoxA que me dice que es un serial inválido).Si quisiéramos buscar un serial ya tendríamos como dato que los primeros caracteres tendrían que ser **"mg37";** pero como todavía soy un poco Newbie y ando flojo en buscar seriales lo voy a parchar. Podríamos invertir todos los saltos con JZ pero preferiría "nopear", no sea cosa que a alguien se le ocurra poner m337 como primeros números de su serial trucho…jeje…obviamente saltaría a chico malo. Por ahora anótense estos saltos MALOS en algún lado o pongan BP en ellos para no olvidarse.
Continuemos y veamos si hay más saltos que nos envíen a la zona de serial inválido:



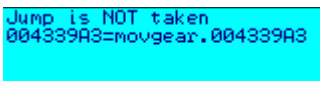
Aquí hay un bucle….trabajará con el serial en ellos, veo unos saltos pero ninguno me lleva a chico malo. Pongo un BP en el CMP a la salida del bucle y doy RUN.



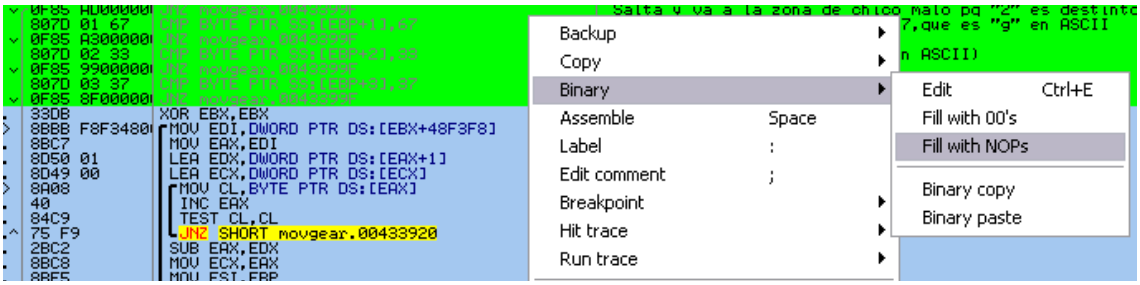
Ahora vemos un salto y una llamada, las pasamos con F8 y no pasa nada. Pongamos un BP en ese CMP ESI, EAX a la salida de este otro bucle:



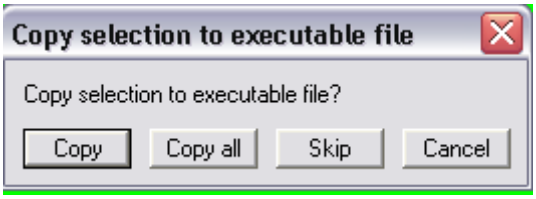
Ese JNZ es el último y si lo pasamos unas instrucciones más abajo viene un RETN, que nos devolverá un poquito más arriba del JE que parchamos al principio y no funcionó. Una vez en el cambiemos el estado del flag Z y veamos si pasando este salto ya estaría todo bien. Recuerdan como cambiar el flag, no? Esta más arriba.
Vemos que ahora el salto no se tomará:



Anoten este salto por las dudas y apretemos F9 a ver si ya esta!!!
El programa corre y en el HELP no aparece más el “REGISTER NOW…”.
Jeje funcionó.Ahora modifiquemos y guardemos los cambios al ejecutable.Primero busquemos todos esos JNZ que estaban juntos y los seleccionamos y hacemos click con el botón derecho y en el menú contextual que se abre elegimos: BINARY/FILL WITH NOPs. :

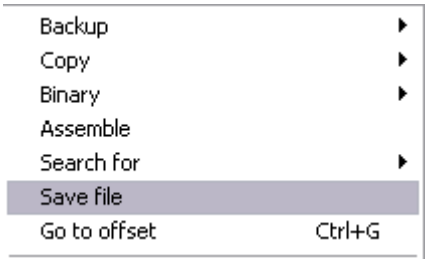


Ahora vamos a el último salto que nos llevaba a chico malo y también lo nopeamos de la misma manera. Una vez hecho esto clickeamos con el botón derecho y en el menú contextual buscamos lo siguiente:



Acá le damos a “Copy all”

Se nos abre una nueva ventana y vovlemos a hacer click con el botón derecho y ahora elegimos SAVE FILE.



Guardamos el archivo con otro nombre en el mismo directorio o en otro lado. Con este no hay problema en guardarlo en otro lado pero otros deben estar si o si en el directorio de instalación del programa. Bueno buscando y ejecutando ven que ya esta registrado y la molesta nag del final ya no aparece. DERROTADO.
Bueno este tute está orientado a muy Newbies y espero que les sirva.
De a poquito y mientras más vaya aprendiendo voy a tratar de devolver a la lista todo lo que me va dando haciendo algunos tutes. Saludos a Ricardo, no te conozco pero te estoy agradecido por el curso de “Cracking con Olly DBG desde 0”.
.:. BhZd .:.