. : [CrackSLatinoS] : .

[[The Art of Reverse Engineering]]

Software	Arkanoid Space Ball
Web	http://www.wegroup.org/games/arkanoid-games/arkanoid-space-ball.html
Tools	Ollydbg 1 y 2
Fecha	25/05/10

[Introducion]

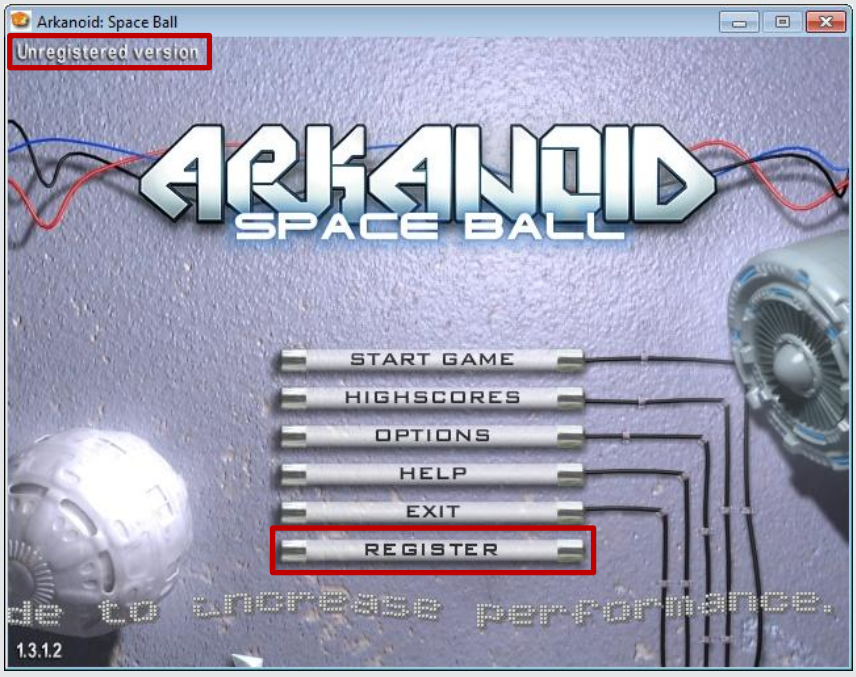
En este mi primer tutorial, veremos el análisis y crackeo de un videojuego, para ello he buscado un juego comercial, *Arkanoid Space Ball* , no tan famoso y claro sin muchas protecciones que generalmente son los desarrollado por Grupos Indie, pero nos servirá para coger bases para el crackeo de otros juegos con protecciones mas complicadas.

[Analisis PE]

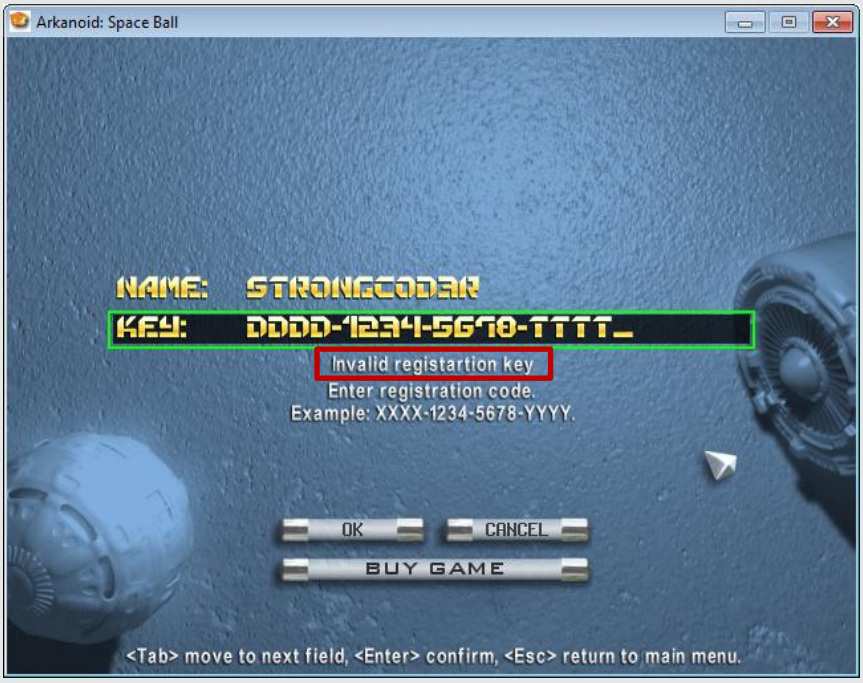
PeiD.95	Nothing found *
Detect iT Easy	Microsoft Visual C++ x.x
ExeInfo PE	Dev-C++ Compiler v4.9.9.? (MINGW 32 vx.x.x)
FastScanner 3	MingWin32 vx.x
RGD Packer Detector 0.6.6	MingWin32 GCC 3.x
	MingWin32 Compiler

[Primer Contacto]

Instalamos el juego y lo ejecutamos, y observamos esto:



Vemos que en la parte superior izquierda nos muestra el mensaje " **Unregistered versión ",** damos clic en **"REGISTER"** y probamos a meter el nombre y el serial y damos ok:



Como vemos nos tira un mensaje "**Invalid registartion key**" diciendo que la clave no es válida, bueno ya tenemos dos pistas por donde buscar:

- **Unregistered versión**
- **Invalid registartion key**

NOTA: Al introducir el fake serial el juego solo detecta teclados en Ingles, bug del juego, así que tuve que cambiar la configuración de mi teclado de Spanish a English en panel de control del S.O.

Me puse a jugar un poco para ver si el juego nos daba algunas sorpresitas, cuando empezamos a jugar vemos esto:

Como vemos hay muchas NAG´s, el autor del juego busca de todas las maneras de que compremos el juego, bueno con esto ya estamos listos para crackearlo.

[Manos a la Obra]

Abrimos el "**arkanoidsb.exe**" con Olly hacemos lo típico, clic derecho Search for → All Referenced strings y ponemos un Bp(*BreakPoint*) en las string que vimos atrás

Olly al correr el juego observaran el fullscreen del juego y se bloquea el juego a causa de los Bp puestos y no hay como minimizar ni nada ya que el juego está en primer plano, probando Ctrl + Alt + Supr para mostrar Administrador de tareas e intentar finalizar el juego, yo uso Win7 y a diferencia de WinXP crea una capa entre la aplicaciones y un menú para invocar al Administrador de tareas sin que el juego interfiera y poder finalizarlo.

Y ahora que hacemos para poder debugearlo si el juego se queda bloqueado y no nos deja hacer nada, una solución sería ir en el juego Menu→ Options poner al "Fullscreen Mode = NO" y problema solucionado:

Pero eso sería lo más fácil a hacer y pensando que no todos los juegos tienen esa opción, así que vamos hacer este tutorial más interesante y solucionar el problema desde Olly.

La pregunta es cómo hacerlo, así que pongámonos a pensar desde lo más básico, que se necesita para programar videojuegos:

- Librería Grafica
- Librería Sonido
- Librería Física
- etc

Creo que la más importante es la librería Grafica ya que controla todo lo visual en el juego como primitivas, imágenes, fonts, Modos de Video etc. Así que cargamos de nuevo el juego en Olly y nos dirigimos a View → Log (Alt + L):

```
          OllyDbg v2.00 (intermediate version - under development!)

          File 'C:\Program Files\Arkanoid Space Ball\arkanoidsb.exe'
          New process (ID 00001EC8) created
004012A0  Main thread (ID 00001200) created
00400000  Module C:\Program Files\Arkanoid Space Ball\arkanoidsb.exe
002C0000  Module C:\Program Files\Arkanoid Space Ball\SDL_image.dll
002E0000  Module C:\Program Files\Arkanoid Space Ball\libpng13.dll
00300000  Module C:\Program Files\Arkanoid Space Ball\zlib1.dll
00320000  Module C:\Program Files\Arkanoid Space Ball\SDL_mixer.dll
10000000  Module C:\Program Files\Arkanoid Space Ball\SDL.dll
64F40000  Module C:\Program Files\Arkanoid Space Ball\jpeg.dll
73310000  Module C:\Windows\system32\WINMM.dll
```

Vemos que Olly ha cargado algunas librerías dinámicas, ósea las que el juego va a utilizar en ejecución y entre ellas vemos las de nombre SDL:

- SDL.dll ←Librería Grafica principal
- SDL_image.dll ←Extensión de SDL para soporte de formato imágenes
- SDL_mixer.dll ←Extensión de SDL para soporte de formatos de sonido

La Librería SDL(*Simple DirectMedia Layer*) es una librería muy conocida para el desarrollo de juegos 2D, hay que tener en cuenta que no es la única, hay muchas más, aquí un listado de las conocidas en el desarrollo de videojuegos:

- DirectX,
- OpenGL,
- Allegro,
- IrrLich,
- Ogre,
- IndieLib,
- ClanLib,
- XNA.Net

Seguimos. En todas las anteriores librerías antes comentadas hay una API que tienen en común todas ellas, es la de establecer el modo de video en el juego. Viendo la documentación de la librería de SDL (Link) encontramos el API que se encarga de establecer el modo de video del juego, aquí el prototipo de la API:

```
// Para mas flags ver documentacion
//Flag: SDL_FULLSCREEN = Pantalla completa

SDL_Surface* SDL_SetVideoMode(int width,        //Ancho de la ventana
                              int height,       //Alto de la ventana
                              int bitsperpixel, //Profundidad de color
                              Uint32 flags);    //Flags
```

Aquí un ejemplos de cómo usarla:

(1)

```
// Juego en FullScreen(pantalla completa)
SDL_SetVideoMode( 1024, 768, 32, SDL_FULLSCREEN);
```

(2)

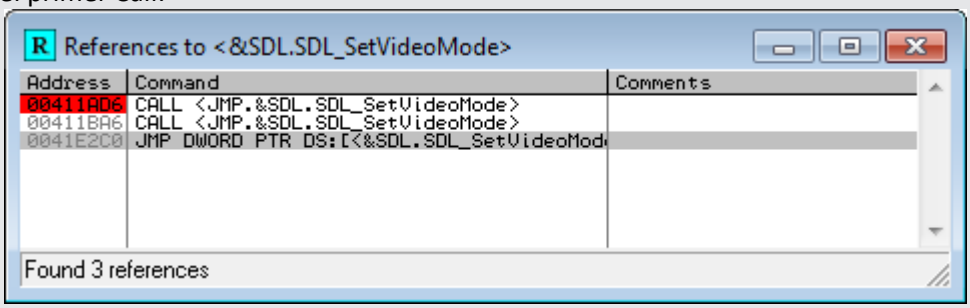
```
// Juego en WindowMode(modo ventana)
SDL_SetVideoMode( 800, 600, 32, SDL_SWSURFACE);
```

Como vemos actualmente el juego se ejecutaría como el ejemplo 1, lo que necesitamos es que se ejecute como el ejemplo 2, así que cuando Olly llame a esas API le cambiamos los parámetros para que se ejecute en pantalla normal, pero como sabemos qué valor tiene cada flag, la respuesta a eso podríamos tenerla si hiciéramos un pequeño ejemplo en C/C++ con SDL con eso parámetros y debugearlo con Olly, pero sería muy lioso, ya que tendríamos que bajar la librería SDL, configurar las librerias con el compilador etc, ese método es recomendado para cuando no tenga el código fuente de las librerías como DirectX, también podríamos intentar meterle algún numero cualquiera por probar, pero como quiero hacer las cosas bien vamos a lo seguro, así que lo que aremos es leer el código fuente de SDL(libre distribución) y ponernos analizar donde tiene definidos esos flags y ver el valor, entonces vamos alla.

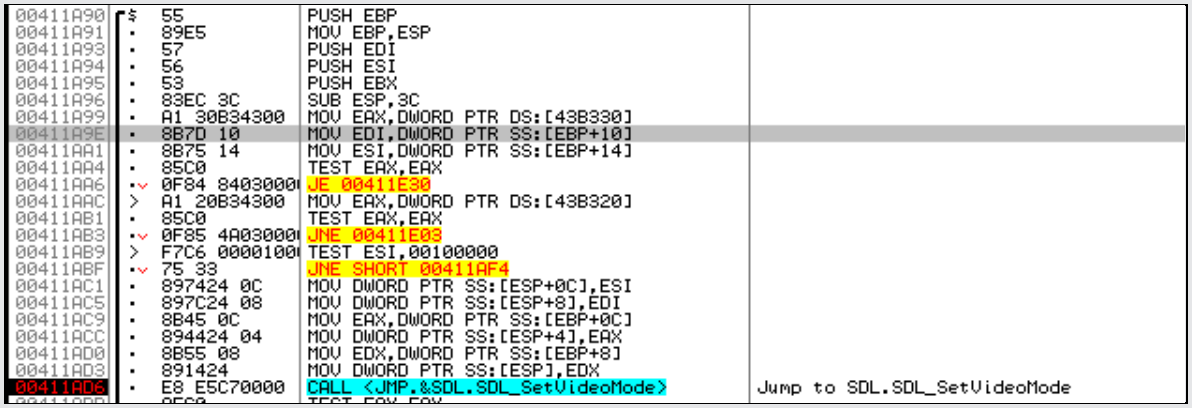
Después de bajarme la librería, la descomprimimos y vemos una tonelada de carpetas, pero dónde buscar?, normalmente estos flag en C/C++ se saben definir el las header .h, no siempre, así que vamos a la carpeta **include** y abrimos el archivo **SDL_video.h** y vemos en las líneas **131-155** definidas las constantes con sus valores y vemos que nuestro flag tiene de valor:

```
SDL_FULLSCREEN = 0x80000000
SDL_SWSURFACE = 0x00000000      // Vale 0, tanta búsqueda para esto ☺
```

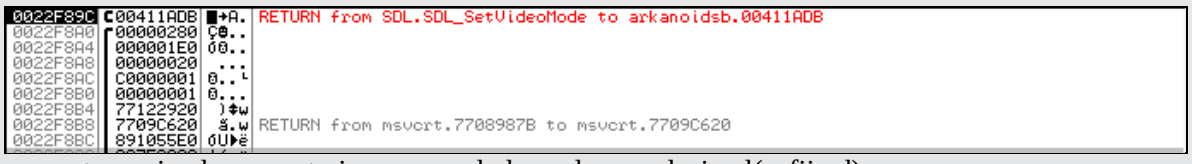
Como podemos ver nuestro flag vale 0, ahora si ya estamos listos, cargamos el juego en Olly y damos clic derecho Search For→Name y ponemos damos clic derecho en la Api **SDL.SDL_SetVideoMode→Find References** y vemos 2 Call, ponemos un Bp en el primer Call:



El segundo Call lo usa el juego cuando cambiamos la configuración en Menu→Options dentro del juego y aplica los cambios, seguimos, damos al botón Run y para aquí:



Vemos a la API SeTVideoMode que se encuentra en el librería SDL.dll, la api va hacer llamada con el Call, tecleamos F7 para entrar en el Call, luego miramos al Stack y vemos algo como esto:



Intepretamos esto poniendo comentarios y pasando los valores a decimal(sufijo d):

```
Address     Value      Comments
0022F89C  00411ADB   ; RETURN from SDL.SDL_SetVideoMode to arkanoidsb.00411ADB
0022F8A0  00000280   ; 640d  Ancho
0022F8A4  000001E0   ; 480d  Alto
0022F8A8  00000020   ; 32 bits  Profundidad de color
0022F8AC  C0000001   ; Valor con la suma de Flags – FullScreen
```

Como vemos nos haría cambiar el valor de la address de los flags poniéndolo a "0" para que se inicie en WindowMode asi que lo modificamos, para ello damos doble click en Value y podemos modificar:

Address	Value	Comments
0022F89C	00411ADB	; RETURN from SDL.SDL_SetVideoMode to arkanoidsb.00411ADB
0022F8A0	00000280	; 640**d** Ancho
0022F8A4	000001E0	; 480**d** Alto
0022F8A8	00000020	; 32 bits Profundidad de color
0022F8AC	00000000	; Valor con la suma de Flags - WindowsMode

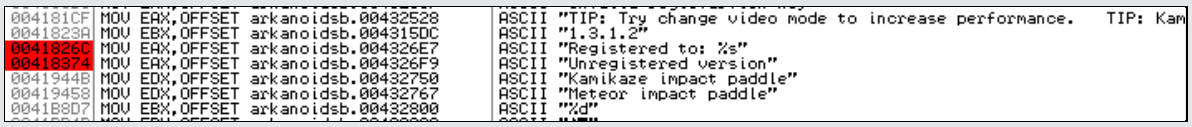
Listo ahora si damos F9 y vemos que inicia en ventana normal podemos poner un Bp y cualquier lado del juego y vemos que ya no tendremos los problemas del bloqueo.

Seguimos, con mi amigo REFoRCE y yo nos hemos propuesto un reto que consiste en buscar distintas soluciones para el crackeo del juego, a continuación mi solución.

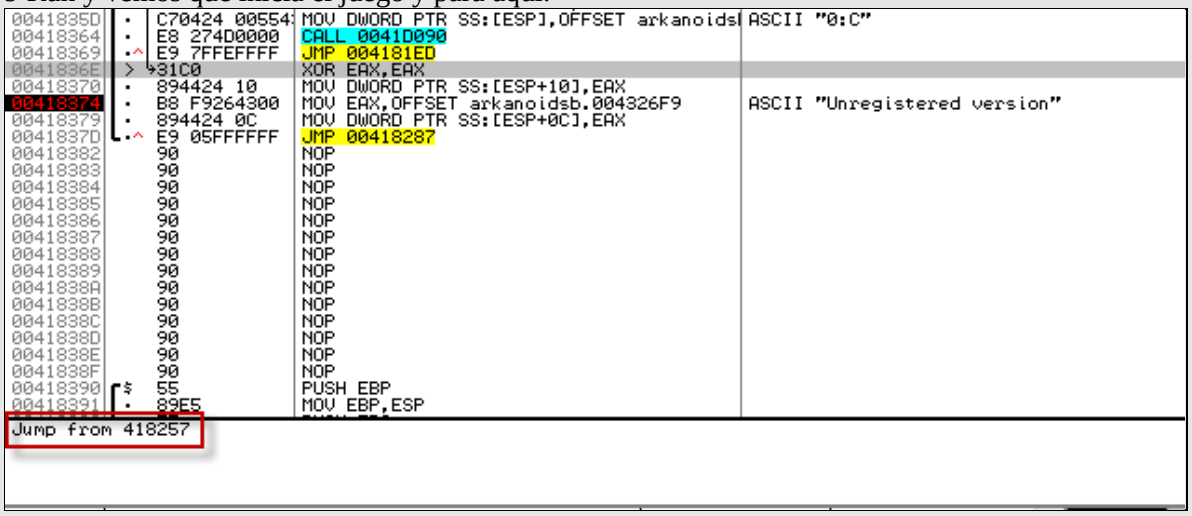
[Solución 1]

Método : Por Parcheo
Autor: StrongCod3r

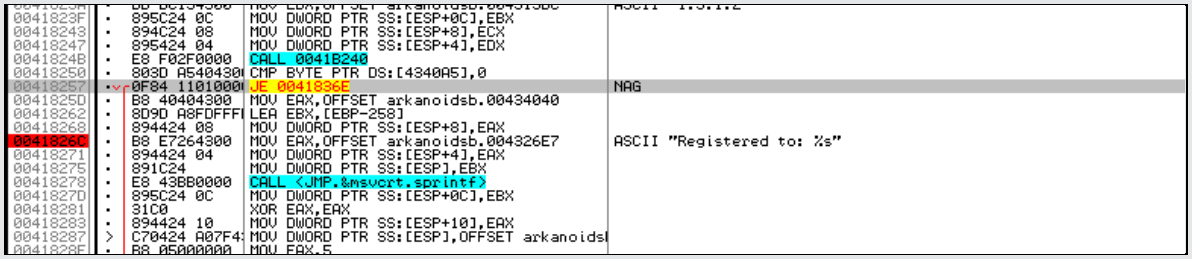
Bueno se que esta solución es la más fácil y rápida y menos elegante, pero igual es una solución :D.
Cargamos el juego vamos a Search for → All Referenced Strings y ponemos un Bp en:



Damos F9 Run y vemos que inicia el juego y para aquí:



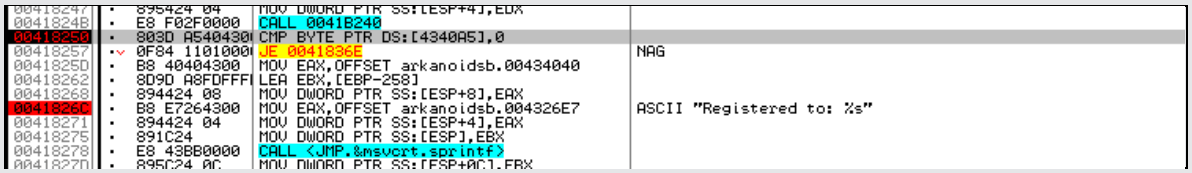
Vemos que hemos saltado desde **418257 ,** asi retrocedemos a esa offset, aquí:



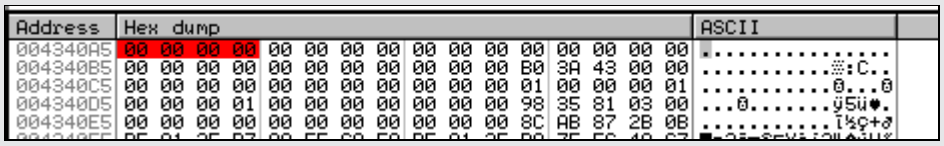
Como vemos caemos en **JE 0041836E** el que nos lanza a la NAG Bad y vemos que más debajo de JE vemos la NAG Good, así que la cuestión de que el juego no muestre la NAG buena esta en ese JE, lo parcharíamos allí pero seria un grave error ya que si parcheamos solo arreglaríamos la NAG mala por la buena y el juego todavía estaría sin registrar, ya que el juego no solo se fija en la NAG sino en muchos lados más para comprobar si de verdad estamos registrados, así que eso de parchar allí no es la solución así vamos a tomar otro camino, arriba del JE vemos :

00418250 |. 803D A5404300 CMP BYTE PTR DS:[4340A5],0

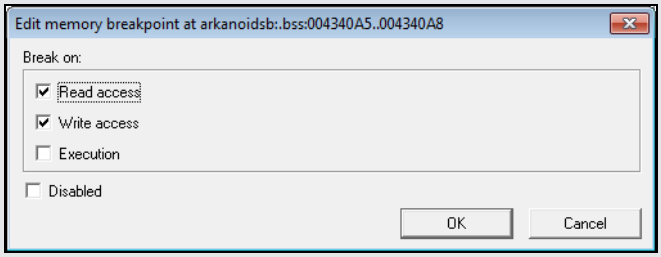
Vemos que compara esta address 4340A5 con 0 si son iguales se activaría el Flag **Z** y JE se cumpliría y saltaría a la NAG Mala, así que para que no se cumpla debe haber algún valor en 4340A5, seguimos, así que ponemos un Bp en 00418250 y reiniciamos el juego y damos run de nuevo y caemos aquí:



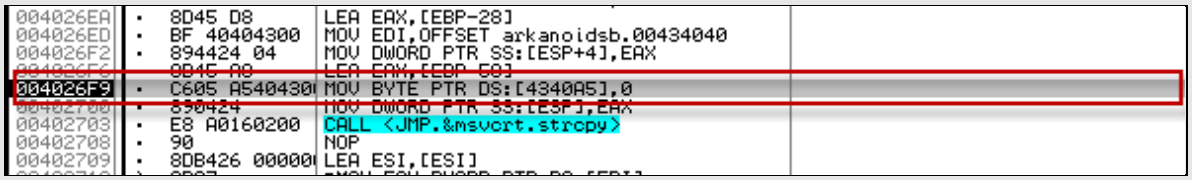
Y vemos en el Dump la que tiene la address 4340A6:



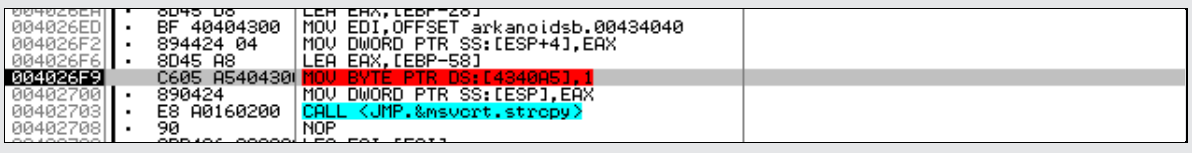
Y vemos que no está a cero osea el CMP se cumpliría pero si estuviera algún valor allí el CMP no se cumpliría y JE no saltaría a la Nag Mala sino que pasaría de largo a la Nag buena. Jojo…ósea la solución está en esa Address, vamos a poner un Bpm(BreakPoint Memory) en 4340A5, seleccionamos el valor de la address clic derecho Breakpoint→Memory:



Marcamos esas Read y Write y damos ok, listo ahora vamos a reiniciar el juego y damos run de nuevo y parara aquí:



Y vemos que está moviendo 0 a nuestra dirección y por consecuencia de eso saltaríamos más después a la NAG mala, la solución es cambiar ese 0 por un 1 y quedaría así:



Una vez que hemos hecho eso más adelante veremos que el JE ya no saltara a la NAG mala sino que pasara de largo, así que quitaremos todos los Bp y Bpm y damos Run y veremos esto:

Vemos que ya nos sale registrado y el botón de "register" ha desaparecido, para comprobar si el juego es bien crackeado me puse a jugar un rato y ni rastro de las Nag´s ni limite de tiempo de jugar, ahora si podemos guardar una copia del exe parchado. Como hemos visto hemos cambiado 1 byte y se a registrado el juego jejej, aquí acaba mi solución vamos a dar paso a la siguiente solución.

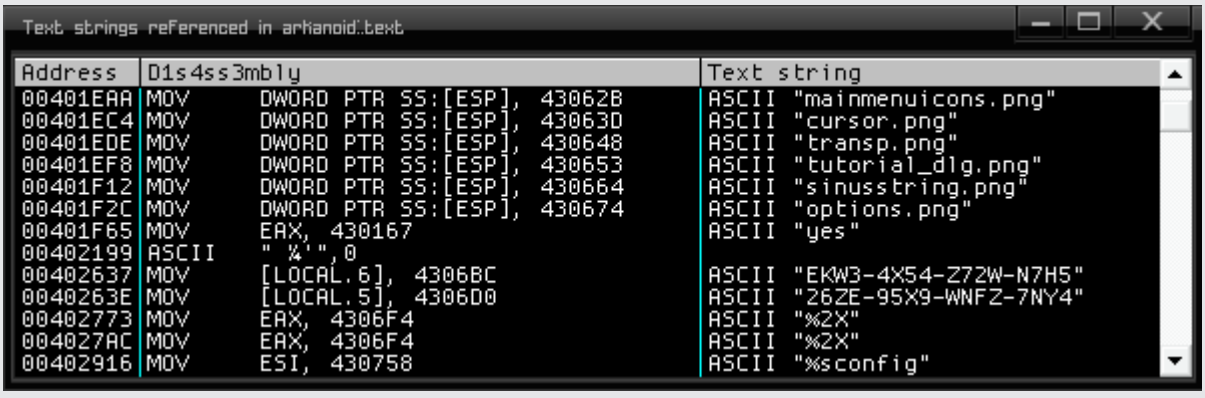
[Solucion 2]

Método: Serial
Autor: REFoRCE

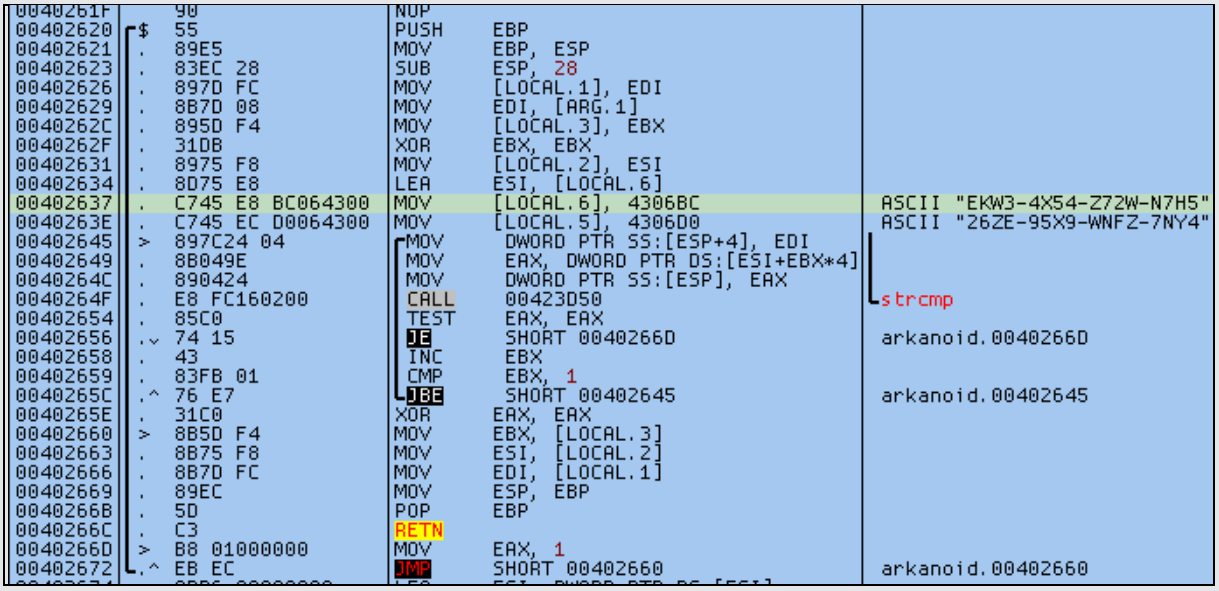
Bueno esta solución es la elegante como dicen lo maestros, así que allí va el va el escrito REFoRCE.

Hola! .Pues me toca la continuación del tute de StrongCod3r acerca de Arkanoid v-1.3.1.2,.La forma en que StrongCod3r registra el programa es una buena opción, demuestra lo simple que puede hacer registrar el juego y es la más fácil sin modificar tanto la aplicación. Bueno la siguiente parte es de cómo registrar el juego atreves de la búsqueda de su serial.
Bueno pues manos a la obra. >)

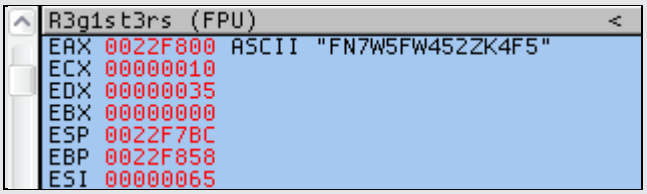
Como ya se vio en el tutorial de StrongCod3r las screenshot del menú principal con el botón:
"Register" y el texto "Unregistered versión" se sabe también que debemos agregar un "name" y un "serial" o "key" tipo: XXXX-XXXX-XXXX-XXXX. Esta vez no agregare ni clave ni name solo lo cargo en ollydbg y empiezo a buscar por pistas.
Bueno entonces lo cargo en ollydbg. Doy Search for-All Referenced text string y veo:



Bueno al parecer son 2 claves en 00402637 y 0040263E así que doy doble clic en alguno de ellas para que me retorne al desensamblado justo aquí:



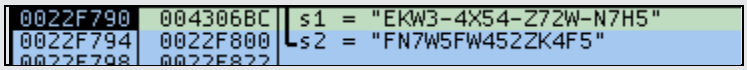
OK. Todo bien pongo un breakpoint or F2 en 00402620 PUSH para no perderme de algunos movimientos que se hagan antes de las claves, doy Run o F9. :) y el breakpoint se activa.
En los registros aparece un valor específicamente en EAX.



Ok anoto ese valor “FN7W5FW452ZK4F5” en el block de notas o en algún editor de textos, y analizo la pieza de código de arriba a partir del push, mientras voy traceando voy haciendo una traducción más o menos de los valores que se toman y de lo que se hace.

00402620	PUSH EBP	
00402621	MOV EBP, ESP	
00402623	SUB ESP, 28	
00402626	MOV [LOCAL 1] , EDI	
00402629	MOV EDI, [ARG 1]	EL valor “FN7W5FW452ZK4F5” a EDI
0040262C	MOV [LOCAL 3] EBX	
0040262F	XOR EBX, EBX	EBX = 0
00402631	MOV [LOCAL 2] , ESI	
00402634	LEA ESI, [LOCAL 6]	
00402637	MOV [LOCAL 6] 004306BC	ASCII “EKW3-4X54-Z72W-N7H5”
0040263E	MOV [LOCAL 5] 004306D0	ASCII “ 26ZE-95X9-WNFZ-7YN4”
00402645	MOV [ESP + 4] EDI	offset de “FN7W5FW452ZK4F5” a ESP +4 = 0022f794
00402649	MOV EAX [ESI + EBX * 4]	EKW3-4X54-Z72W-N7H5 a EAX
0040264C	MOV [ESP] EAX	
0040264F	CALL 00423D50	{ strcmp ➔ step into F7 :)
00402654	TEST EAX , EAX	
00402656	JE SHORT 0040266D	
00402658	INC EBX	
00402659	CMP EBX, 1	
0040265C	JBE SHORT 00402645	
0040265E	XOR EAX, EAX	
00402660	MOV EBX [LOCAL 3]	
00402663	MOV ESI [LOCAL 2]	
00402666	MOV EDI [LOCAL 1]	
00402669	MOV ESP, EBP	
0040266B	POP EBP	
0040266C	RETN	
0040266D	MOV EAX, 1	
00402672	JMP 00402660	

Step into→ 0040264F CALL strcmp lo que hará aquí será comparar:

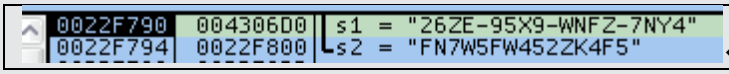


Está claro que no son iguales, dentro del call se sé hace la comparación del primer byte del valor: "EKW3-4X54-Z72W-N7H5" con el primer byte de "FN7W5FW452ZK4F5" como no son iguales F y E , EAX hace una SBB quedando en FFFFFFFF y después un SHL ,1 quedando en FFFFFFFE después se Incrementa quedando en -1. Osea [FFFFFFFF]. .

Saliendo del CALL		
00402654	TEST EAX, EAX	EAX = -1 [FFFFFFFF]
00402656	JE SHORT 0040266D	sin tomarse.
00402658	INC EBX	EBX a 1
00402659	CMP EBX, 1	Cmp 1, 1
0040265C	JBE SHORT 00402645	Salto tomado : 0
0040265E	XOR EAX, EAX	
00402660	MOV EBX [LOCAL 3]	
00402663	MOV ESI [LOCAL 2]	
00402666	MOV EDI [LOCAL 1]	
00402669	MOV ESP, EBP	
0040266B	POP EBP	
0040266C	RETN	
0040266D	MOV EAX, 1	
00402672	JMP 00402660	

Ok. Como el JBE fue tomado al tracear con F8 me retorna a 00402645 y esta vez tomara la segunda clave: "26ZE-95X9-WNFZ-7YN4" y lo comparara con "FN7W5FW452ZK4F5" lo puedo checar en el stack al llegar traceando con ***step over*** sobre :

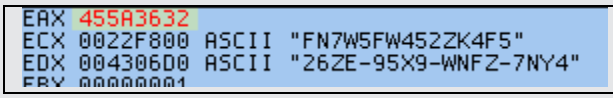
0040264F	CALL 00423D50	strcmp	← :)



← Stack

Entro con ***step into*** or **F7** in strcmp veo que hace lo mismo:

```
77C17730 MOV     EDX, DWORD PTR SS:[ESP+4]
77C17734 MOV     ECX, DWORD PTR SS:[ESP+8]
77C17738 TEST    EDX, 3
77C1773E JNZ     SHORT 77C1777C
77C17740 MOV     EAX, DWORD PTR DS:[EDX]
77C17742 CMP     AL, BYTE PTR DS:[ECX]      ← Comparación de los siguientes
77C17744 JNZ     SHORT 77C17774               valores al tracear hasta aquí con F7
                                              Se pueden ver las comparaciones en los registros.
```



← REGISTROS.

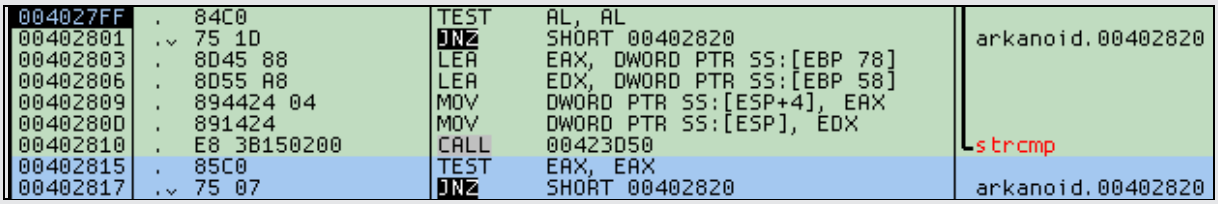
Más claro que el agua ! Las 2 claves fueron comparadas con el valor "FN7W5FW452ZK4F5" probablemente hará lo mismo con mi clave "key" falsa haciendo lo mismo en strcmp. Ok traceo hasta salir del CALL por lo tanto doy: ***Execute till user code*** or **Alt+ F9**.

Para llegar aquí:

00402654	TEST EAX, EAX	Igual EAX = FFFFFFFF (-1)
00402656	JE SHORT 0040266D	el salto no es tomado y se incrementa EBX
00402658	INC EBX	que anteriormente era 1 al incrementarse será 2
00402659	CMP EBX, 1	se compara con 1 por lo tanto el JBE tampoco
0040265C	JBE SHORT 00402645	es tomado permitiéndome tracear hasta el RETN.
0040265E	XOR EAX, EAX	Ok. Traceo hasta el RETN con ctrl+F9
00402660	MOV EBX [LOCAL 3]	
00402663	MOV ESI [LOCAL 2]	

00402666	MOV EDI [LOCAL 1]
00402669	MOV ESP, EBP
0040266B	POP EBP
0040266C	RETN
0040266D	MOV EAX, 1
00402672	JMP 00402660

Llegando hasta el RETN en 0040266C doy ***step in*** or **F7** para llegar aquí:



Sip otro strcmp,…….:

Ok. Todo bien, es tiempo de ejecutar por completo doy F9 y pongo mi "key" falsa:

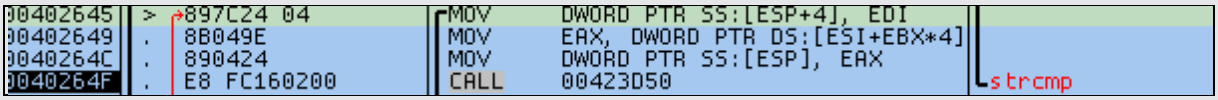
ABCD-1234-5678-ABCD

Y hago lo mismo ingreso mi nickname:

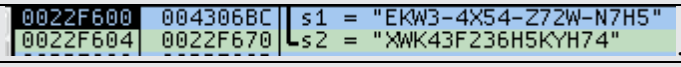
REFORCE

Presiono el botoncillo para aceptar los valores y veo que mi Bp en el PUSH se activo

00402620	PUSH EBP	← BP F2 Activado y traceo hasta llegar a:



Sip en strcmp de nuevo. Ahora veo el stack y un nuevo valor comparando la clave:
"EKW3-4X54-Z72W-N7H5"

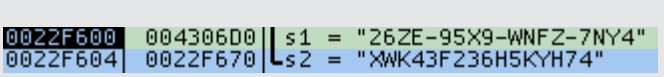


← Un nuevo valor! :).

Wtf! Okay se supone que compara también: "26ZE-95X9-WNFZ-7YN4" con el nuevo valorcito:
"XWK43F236H5KYH74" y para asegurarme pongo un Breakpoint en:

00402645 MOV DWORD PTR SS: [ESP + 4], EDI

Un poco arriba de strcmp en 0040264F. Y doy run o F9.
OK al hacer esto mi Breakpoint se activa y traceo con F8 de nuevo hasta strcmp en 0040264F.
Por lo tanto puedo ver el stack y checar mi suposición que es correcta.



← Igual con el valor nuevo
Se realizó la comparación.

Ok. Traceo hasta llegar al RETN que se encuentra un poco más debajo de strcmp.

0040266C	RETN	← aqui doy step into.

Entonces llego de Nuevo aquí y traceo hasta strcmp:

```
004027FF   .  84C0          TEST    AL, AL
00402801   .v 75 1D          JNZ     SHORT 00402820                  arkanoid.00402820
00402803   .  8D45 88        LEA     EAX, DWORD PTR SS:[EBP 78]
00402806   .  8D55 A8        LEA     EDX, DWORD PTR SS:[EBP 58]
00402809   .  894424 04      MOV     DWORD PTR SS:[ESP+4], EAX
0040280D   .  891424        MOV     DWORD PTR SS:[ESP], EDX
00402810   .  E8 3B150200    CALL    00423D50                        strcmp
00402815   .  85C0          TEST    EAX, EAX
00402817   .v 75 07          JNZ     SHORT 00402820                  arkanoid.00402820
```

Traceando hasta strcmp veo en el stack mi “key” falsa comparada con: XWK43F236H5KYH74

```
0022F630  0022F670|ASCII "XWK43F236H5KYH74"
0022F634  0022F650|ASCII "ABCD12345678ABCD"
```

Interesante!.
Hasta aquí se sabe que hay 2 valores con los cuales las claves son comparadas

1.- XWK43F236H5KYH74

2.- FN7W5FW452ZK4F5

Bueno es tiempo de tomar un riesgo. Testear los dos valores, mientras tanto reinicio. Ctrl-F2.
Hasta aquí se que mi “key” fue comparada con “XWK43F236H5KYH74” okay.
Empiezo con esa quito todos los Bp. Doy Run F9. Presiono el botón de “Register” eh ingreso la clave de esta manera:

XWK4-3F23-6H5K-YH74

Y también mi name: REFORCE jeje. Clic para aceptar los valores y:

Yep, >) Registered!

Testeando: FN7W5FW452ZK4F5 quedaría así la clave FN7W-5FW4-52ZK-4F5
Esta clave funciona sin agregar un nombre y:

Ok. Se ha registrado el juego.

Bueno espero haberme entendido ☺ en este tutorial, no soy buen escritor y creo hice mi mejor esfuerzo jeje, Saludos a todos los que se conectan los fines de semana en el IRC-Hispano en el canal **# CrackslatinoS** aunque no pertenezco a la lista de CLS ahí me encuentro generalmente en el canal.

StrongCod3r : Bueno ya tenemos las 2 soluciones pero además de crackear vamos a jugar un rato a ser dios en el juego, seguid leyendo.

[GameHack con Olly]

Viendo en las teorías tutoriales en lo que se refiere al GameHack(trainers y formas de hacer cheats) hacían uso de programas externos para hacer dicha tarea, ahora vamos a ver cómo hacerlo solo con olly + plugin sin ningún programa, otro método más de hacer gamehack.

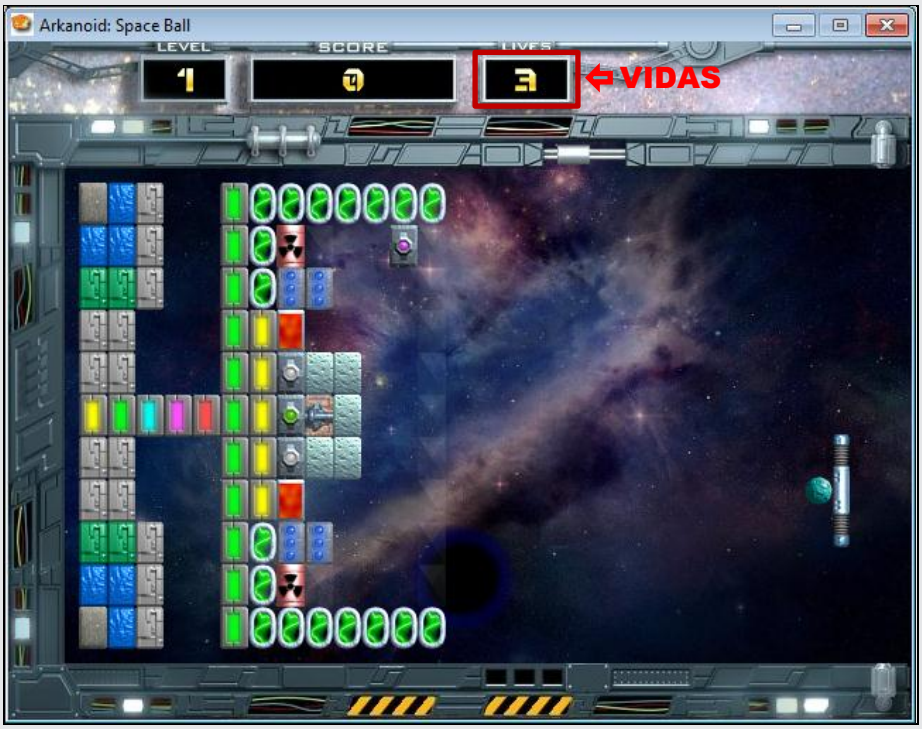
Para dicha tarea vamos hacer uso de unos plugins que encontrado para olly que nos va ayudar a encontrar los valores en la memoria.

Bueno los plugins son 2 :

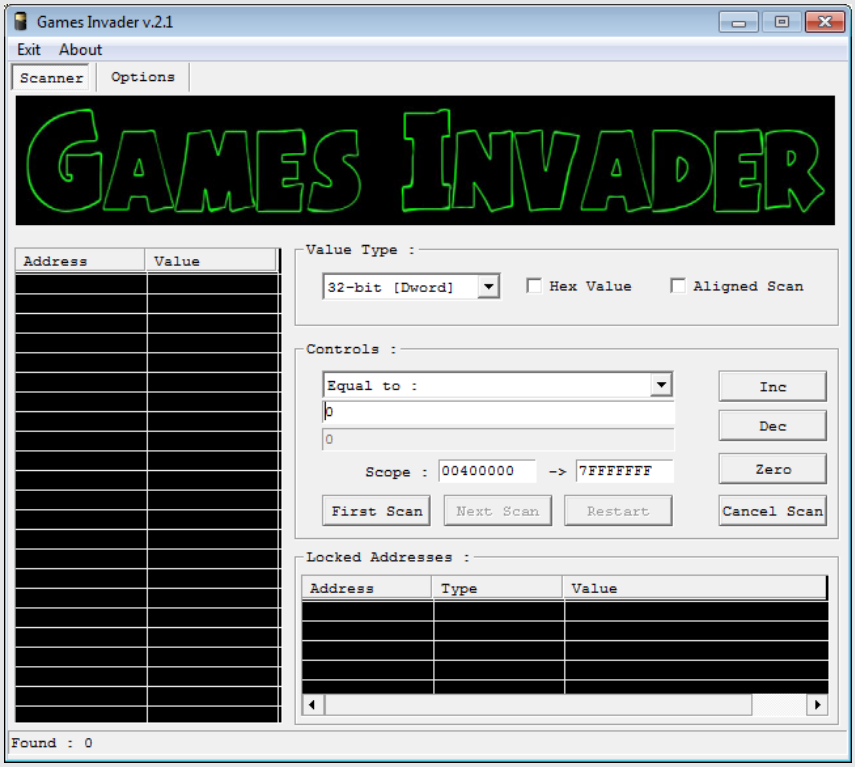
Games Invader.dll
CheatUtility.dll

Los dos son muy bueno , pertenecen al mismo autor *GamingMasteR* , pero personalmente uso el GameInvader.dl porque tiene más opciones, se las adjuntado con el tutorial, cópienlo en la carpeta de plugins de olly.

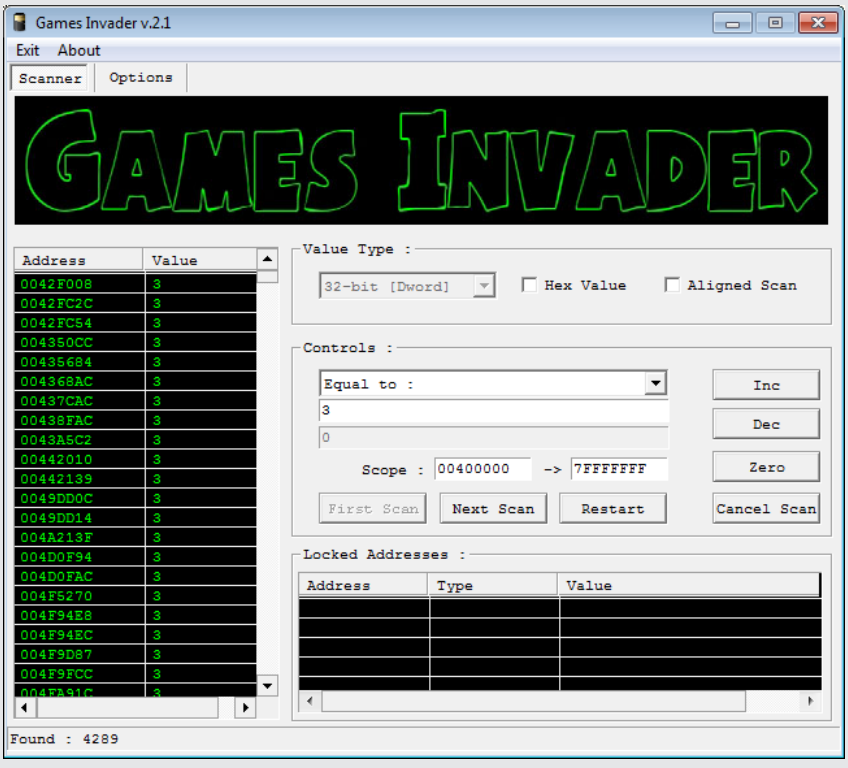
Bueno como estaba cansado de morir y gastarme esas pocas 3 vidas que me dan y que me tire el GameOver, vamos arreglar eso. Cargamos el juego y jugamos una partida y mostrara esto:



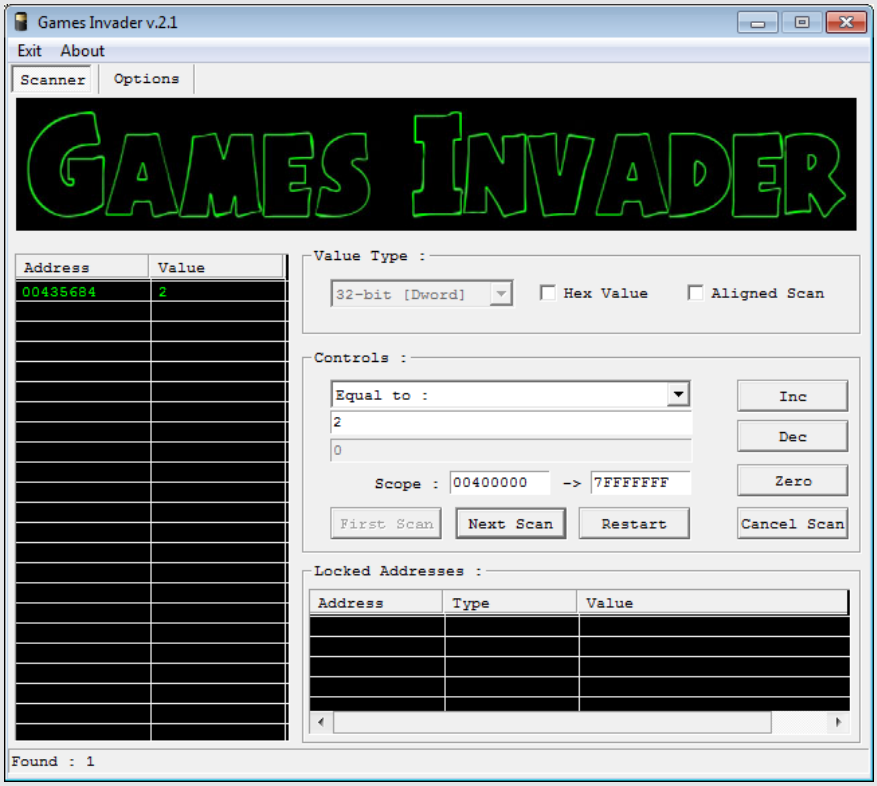
Hay vemos las 3 vidas que nos dan al empezar el juego, bueno vamos al olly en el "Menu→Plugins→ GamesInvader→ BeginScan", y nos aparecerá la siguiente ventana:



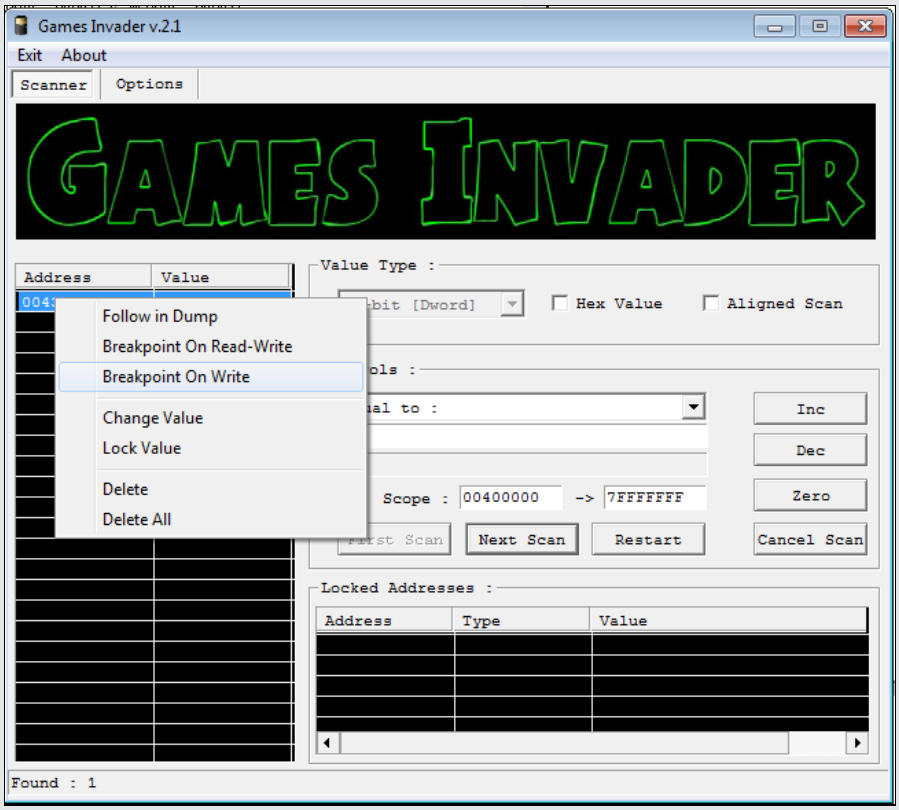
En el 0 ponemos el número de vidas que tenemos actualmente, ósea 3 y damos clic en **First Scan** y vemos esto:



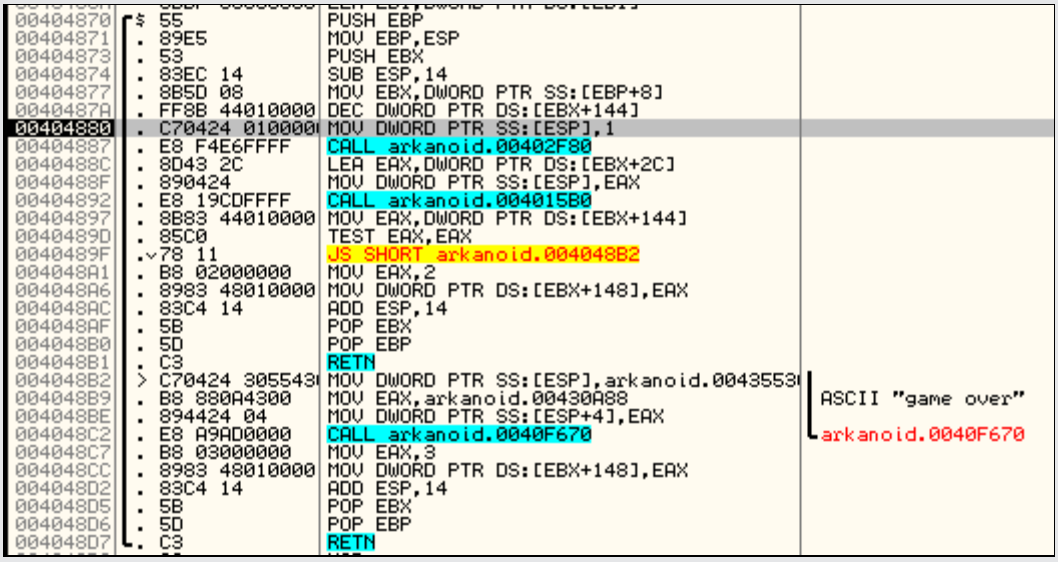
Ha buscado en toda la memoria el numero 3 y vemos que hay bastantes Address con ese valor, sin cerrar la ventana del GamesInvader regresamos al juego y seguimos jugando y perdemos intencionalmente una vida para que nos quede 2 vidas, volvemos al olly sin cerrar el juego y en la ventana GamesInvaders , en el cuadro de texto donde estaba 3 lo ponemos a 2 y damos al Boton **Next Scan** y vemos esto:



Vemos que ha descartado las demás Address y solo ha quedado una (en su pc puede ser otra dirección), en esa address damos clic derecho → Breakpoint On Write, asi:

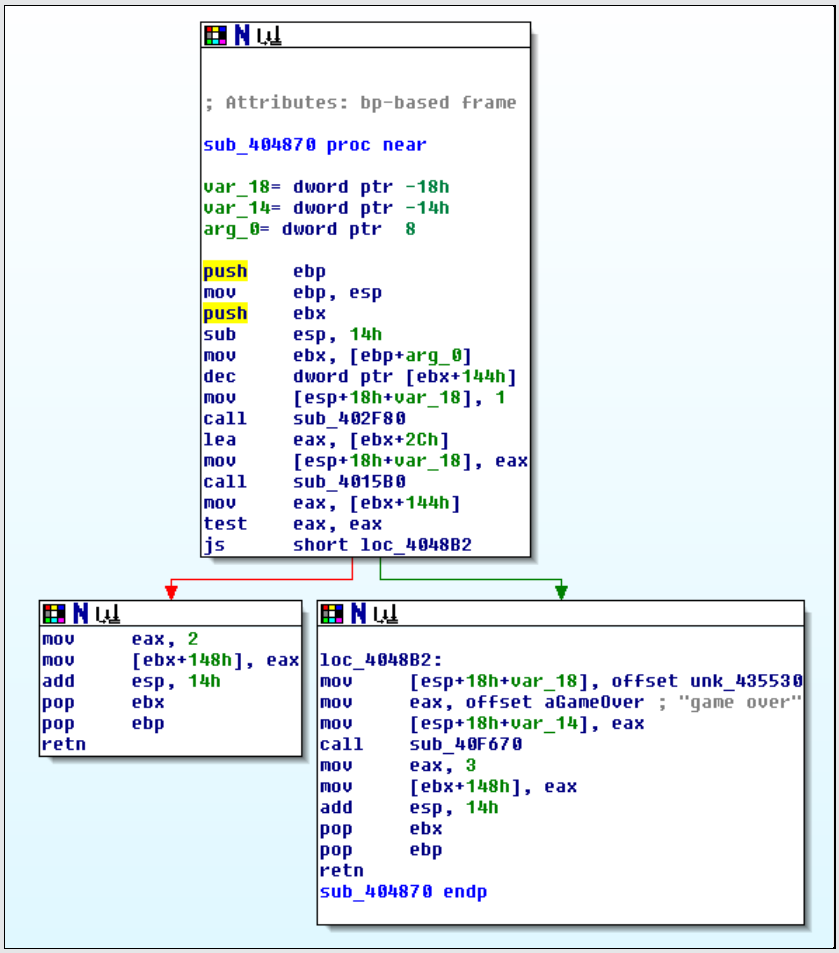


Listo ahora ya podemos cerrar el GameInvaders y seguimos jugando el juego y otras ves perdemos intencionalmente otra vida y parara en el Bp que pusimos antes, aquí:

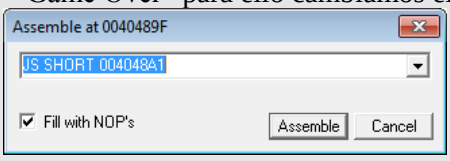


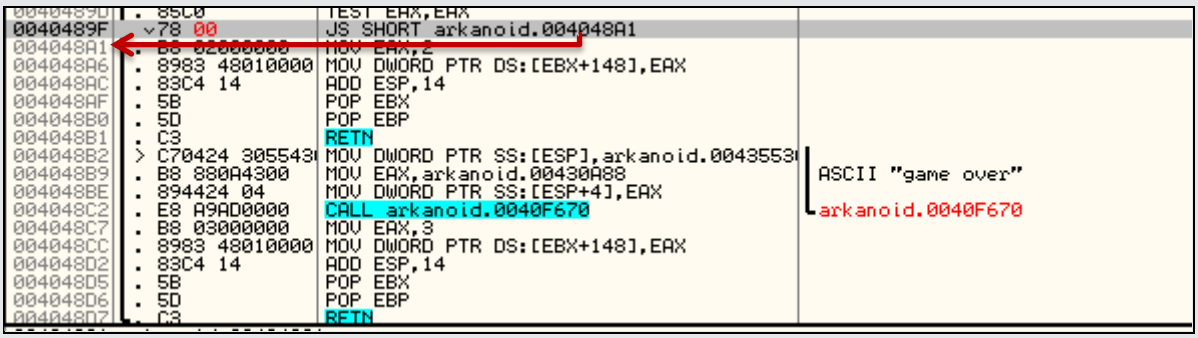
Yo creo que no hace falta explicar nada aquí lo que hace desde la offset **00404870** a **004048D**7, bueno porsiacaso algún despistado no lo ve todavía vamos a dar algunas pistas de lo que hace.

Hay vemos al DEC es el que se encarga de restar las vidas, más abajo vemos un **test eax, eax** si estuviera a cero eax el JS saltaría al "Game Over" :D, para que lo vean más claro yo recomiendo que cuando lleguen a uno de estos puntos y halla mucho código(que no es nuestro caso pero lo vamos a ver, es más útil en juegos de mayor tamaño) , aunque se puede hacer todo con olly sin ningún problema, veamos como se ve e IDA en modo Graph(grafico):



Hay vemos el flujo del juego cuando pasa por el JS el que decide si seguimos jugado o nos manda al Game Over, vamos hacer algo como lo hizo Marciano en su tute del NES, pero nosotros no vamos a restar o sumar las vidas, vamos hacer que el juego nunca nos tire "Game Over" para ello cambiamos el JS con esto :





Vemos que cambiamos la el Offset del JS que nos tiraba al “GameOver” por la siguiente, ósea continuaría el flujo juego normalmente aunque ya no tengamos vidas seguiremos jugando y no perderemos nunca.

Bueno al final me he divertido más crackeandolo que jugando propio juego :D, con esto damos por terminado este MegaHiperNewbie-Tute, pido disculpas desde ya por los errores en el tute y la escritura del tute que seguro los hay.

Saludos los que siempre están en el IRC Zapper, KrozmiCCLS, Asphyxia, jackgris y a toda la lista y en especial a Ricardo por estar tantos años al pie del cañón.

By StrongCod3r - REFoRCE