

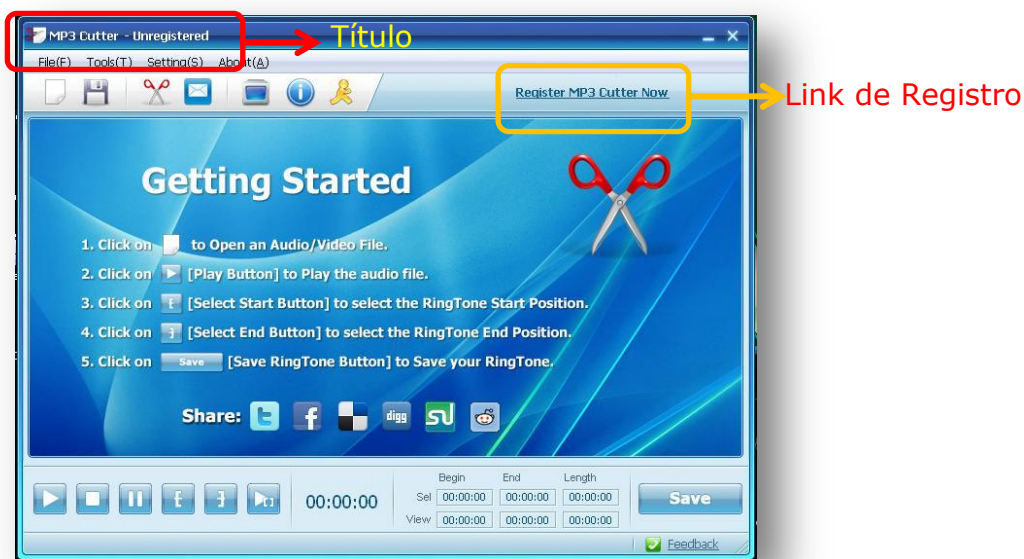
# Serial Fishing

**Programa** : MP3 Cutter v1.1.0  
**Descarga** : <http://www.mp3cutter.org/down/mp3cutter.exe>  
**Objetivo** : Obtener serial válido  
**Nivel** : Fácil  
**Reverser** : Elix

MP3 Cutter es una herramienta que permite cortar audios mp3's, y en este momento me urgía algo así para cortar un audio en partecitas y hacer una especie de dedicatoria personalizada (mentira solo buscando seguir el tutorial de Ivinson <http://ricardonarvaja.info/WEB/CURSO%20NUEVO/TEORIAS%20NUMERADAS/1301-1400/1328-AudioCutter-%20Mi%20primer%20tuto%20para%20CLS-By%20Ivinson.pdf>) por error baje un binario distinto y esto fue lo que surgió.

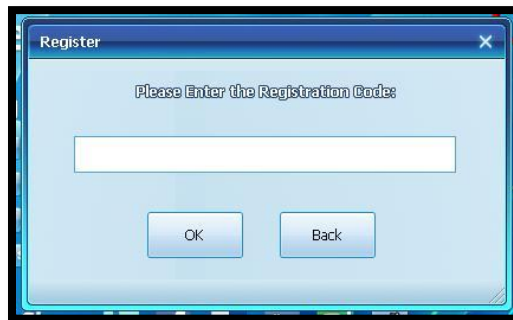
## 1. Instalación y Recojo de Información

La aplicación lo podemos bajar libremente desde la página oficial, lo cual bajamos e instalamos al abrir el programa (MP3 Cutter) en cuestión no muestra NAG (Carteles al inicio de aplicación) ni impedimento alguno para su uso.



*Imagen 01. Ventana principal de MP3 Cutter*

Por el título podemos notar que la aplicación necesita registrarse "MP3 Cutter – Unregistered" y el link que nos lleva a la ventanita del registro es "Register Mp3 Cutter Now" sin mayores perjuicios hagamos clic y veamos a lo que nos enfrentaremos



*Imagen 02. Ventana de Registro*

Junto a la ventana del registro se abre el navegador con una página con un contenido web donde podemos distinguir la opción de compra del programa con descuento y todo a...



*Imagen 03. Costo del programa*

Bueno ya sabemos que había en ese contenido web ahora lo cerramos y comencemos por recoger información de nuestro objetivo, ingresemos un serial falso para ver las strings del chico malo otro dato importante es también el título de la ventana *Register* hasta ahí más que suficiente para laburar



*Imagen 04. Serial falso y chico malo*

Ahora sabemos que la validación lo va hacer en alguna parte de sus entrañas (bien por nosotros). Sigamos con el recojo de información, sometamos a nuestro objetivo al valor de la verdad ;) y hagámosle confesar en que lenguaje le escribieron.... Jejeje no es necesario para esto utilizaremos *RDG Packer Detector* que dicho sea de paso nos dirá también si está empacado o no.



Imagen 05. Información de RDG Packer Detector

Nuestra aplicación está escrita en Visual Basic 6.0 Código Nativo y no cuenta con protección alguna eso nos dice la herramienta.

## 2. Pescando el Serial

Ahora se viene lo bueno XD, abrimos el OllyDbg y ponemos nuestro programita para verle las entrañas y con algo de suerte pescamos el serial, lo cual es objetivo de este tutorial. En la imagen a continuación nos muestra justo después de cargar el MP3Cutter en Olly cuyo título nos indica de posible código comprimido pero en pregunta, y dos opciones si queremos que olly continúe con el análisis sí o no... nosotros le daremos que no

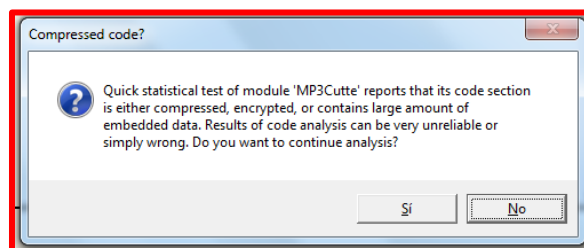


Imagen 05. Posible código comprimido

En la ventana del disassembler del OllyDbg vemos el Entry Point (EP) siendo este la primera instrucción a ejecutarse

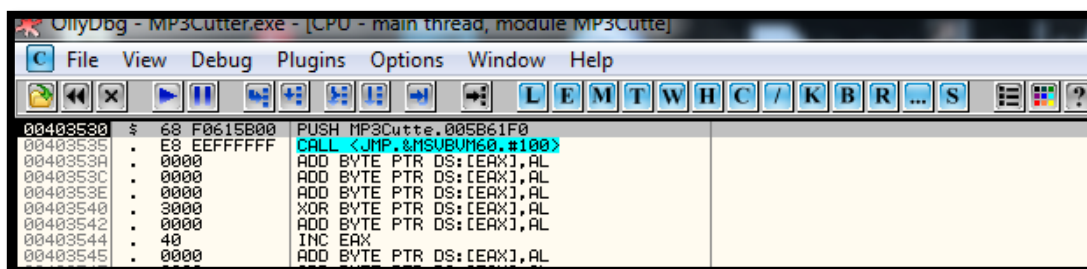


Imagen 06. Entry Point de MP3 Cutter

Para el laburo en esta oportunidad tengo las excepciones del Olly seteado (*Debugging Options > Exceptions*) de este modo

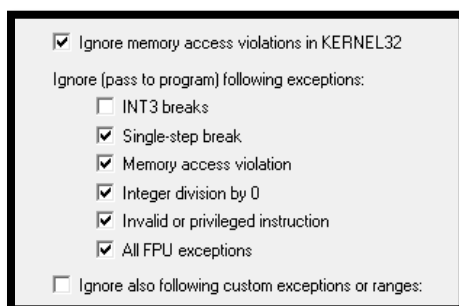


Imagen 07. Configuración de la ventana de Excepciones

Con eso marcharía de perlas :P para clarificar más el panorama vayamos a ver las API que importa (utiliza) el programa clic derecho *search for > Name (label) in current module* o [CTRL]+N, como dato primordial sabemos que el objetivo no posee protección alguna lo sabemos por el RDG Packer Detector (ver Imagen 4)

00401108 .text	Import	MSUBUM60 .vbaAryConstruct2	00401104 .text	Import	MSUBUM60 .vbaNew2
00401258 .text	Import	MSUBUM60 .vbaAryCopy	004010E0 .text	Import	MSUBUM60 .vbaNextEachCollObj
00401084 .text	Import	MSUBUM60 .vbaAryDestruct	004010A4 .text	Import	MSUBUM60 .vbaObjSet
0040107C .text	Import	MSUBUM60 .vbaAryVar	00401080 .text	Import	MSUBUM60 .vbaObjSetAddr
00401254 .text	Import	MSUBUM60 .vbaCastObj	00401114 .text	Import	MSUBUM60 .vbaObjVar
004010E8 .text	Import	MSUBUM60 .vbaChkstk	004010A0 .text	Import	MSUBUM60 .vbaOnError
004011F4 .text	Import	MSUBUM60 .vbaDerefAry1	00401168 .text	Import	MSUBUM60 .vbaPrintFile
00401034 .text	Import	MSUBUM60 .vbaEnd	00401108 .text	Import	MSUBUM60 .vbaRstr
004011C8 .text	Import	MSUBUM60 .vbaErrorOverflow	00401040 .text	Import	MSUBUM60 .vbaRaiseEvent
0040115C .text	Import	MSUBUM60 .vbaExceptHandler	00401050 .text	Import	MSUBUM60 .vbaRecAnsiToUni
0040108C .text	Import	MSUBUM60 .vbaExitProc	00401280 .text	Import	MSUBUM60 .vbaRecAssign
004010F0 .text	Import	MSUBUM60 .vbaFileClose	0040106C .text	Import	MSUBUM60 .vbaRecDestruct
004011CC .text	Import	MSUBUM60 .vbaFileOpen	0040123C .text	Import	MSUBUM60 .vbaRecDestructAnsi
00401130 .text	Import	MSUBUM60 .vbaFixstrConstruct	00401140 .text	Import	MSUBUM60 .vbaRecUniToAnsi
00401094 .text	Import	MSUBUM60 .vbaForEachCollObj	00401070 .text	Import	MSUBUM60 .vbaSetSystemError
0040118C .text	Import	MSUBUM60 .vbaFreeException	00401060 .text	Import	MSUBUM60 .vbaStrCat
004010C8 .text	Import	MSUBUM60 .vbaFreeObj	00401100 .text	Import	MSUBUM60 .vbaStrCmp
0040122C .text	Import	MSUBUM60 .vbaFreeObjList	004011E8 .text	Import	MSUBUM60 .vbaStrCopy
00401234 .text	Import	MSUBUM60 .vbaFreeStrList	004010C0 .text	Import	MSUBUM60 .vbaStrFixstr
00401274 .text	Import	MSUBUM60 .vbaFreeVar	00401008 .text	Import	MSUBUM60 .vbaStrI2
004010BC .text	Import	MSUBUM60 .vbaFreeVarList	00401018 .text	Import	MSUBUM60 .vbaStrI4
004010CC .text	Import	MSUBUM60 .vbaGenerateBoundsError	00401250 .text	Import	MSUBUM60 .vbaStrMove
00401044 .text	Import	MSUBUM60 .vbaInitStr	00401128 .text	Import	MSUBUM60 .vbaStrR4
00401288 .text	Import	MSUBUM60 .vbaLateMemCall	0040113C .text	Import	MSUBUM60 .vbaStrR8
004011F0 .text	Import	MSUBUM60 .vbaLateMemCallLd	00401220 .text	Import	MSUBUM60 .vbaStrToAnsi
00401020 .text	Import	MSUBUM60 .vbaLateMemCallLd	0040116C .text	Import	MSUBUM60 .vbaStrToUnicode
004010F8 .text	Import	MSUBUM60 .vbaLateMemSt	0040125C .text	Import	MSUBUM60 .vbaStrVarCopy
00401074 .text	Import	MSUBUM60 .vbaLateMemSt	00401024 .text	Import	MSUBUM60 .vbaStrVarMove
0040105C .text	Import	MSUBUM60 .vbaLateMemSt	004011A4 .text	Import	MSUBUM60 .vbaStrVarVal
00401118 .text	Import	MSUBUM60 .vbaLateMemSt	00401140 .text	Import	MSUBUM60 .vbaStrVarVal
00401210 .text	Import	MSUBUM60 .vbaLateMemSt	0040114C .text	Import	MSUBUM60 .vbaStrVarVal
00401170 .text	Import	MSUBUM60 .vbaLateMemSt	00401218 .text	Import	MSUBUM60 .vbaStrVarAdd
00401100 .text	Import	MSUBUM60 .vbaLateMemSt	004011A8 .text	Import	MSUBUM60 .vbaStrVarCat
00401198 .text	Import	MSUBUM60 .vbaLateMemSt	00401238 .text	Import	MSUBUM60 .vbaStrVarCopy
00401134 .text	Import	MSUBUM60 .vbaLateMemSt	00401180 .text	Import	MSUBUM60 .vbaStrVarDiv
00401214 .text	Import	MSUBUM60 .vbaLateMemSt	00401224 .text	Import	MSUBUM60 .vbaStrVarDup
00401244 .text	Import	MSUBUM60 .vbaLateMemSt	00401090 .text	Import	MSUBUM60 .vbaStrVarInit
00401088 .text	Import	MSUBUM60 .vbaLateMemSt	00401278 .text	Import	MSUBUM60 .vbaStrVarNext
0040102C .text	Import	MSUBUM60 .vbaLateMemSt	00401144 .text	Import	MSUBUM60 .vbaStrVarNextSt
00401068 .text	Import	MSUBUM60 .vbaLateMemSt	00401014 .text	Import	MSUBUM60 .vbaStrVarMove
00401148 .text	Import	MSUBUM60 .vbaLateMemSt	00401158 .text	Import	MSUBUM60 .vbaStrVarMul
			00401120 .text	Import	MSUBUM60 .vbaStrVarOr
			00401004 .text	Import	MSUBUM60 .vbaStrVarSub
			0040110C .text	Import	MSUBUM60 .vbaStrVarTstEq
			0040120C .text	Import	MSUBUM60 .vbaStrVarTstNe
			0040101C .text	Import	MSUBUM60 .vbaStrVarTstGt
			00401064 .text	Import	MSUBUM60 .vbaStrVarTstLe
			00401080 .text	Import	MSUBUM60 .vbaStrVarTstLt
			00401138 .text	Import	MSUBUM60 .vbaStrVarTstNeq

Imagen 08. Listado principal de las API importadas

En la imagen de arriba vemos el listado de las API y varias candidatas a BreakPoint (BPX) para no hacerle muy larga y aburrida al tutorial pongamos un BPX en `__vbaStrCmp` pidiéndole de ese modo a Olly que se detenga (por el BPX) cuando trate de comparar String recordar que nuestro serial falso (ver Imagen 02) también es una String y tal vez la comparación se detenga cuando trate de comparar nuestro serial (falso) con otro string si fuese así nos ahorraríamos labores más tediosas... veamos

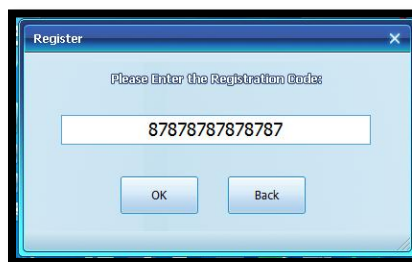
00401070	.text	Import	MSUBUM60. __vbaStrCmp
00401070	.text	Import	MSUBUM60. __vbaSetSystemError
00401060	.text	Import	MSUBUM60. __vbaStrCat
00401100	.text	Import	MSUBUM60. __vbaStrCmp
00401100	.text	Import	MSUBUM60. __vbaStrCopy
004010C0	.text	Import	MSUBUM60. __vbaStrFixstr
00401008	.text	Import	MSUBUM60. __vbaStrI2
00401018	.text	Import	MSUBUM60. __vbaStrI4

*Imagen 09. Breakpoint en API `__vbaStrCmp`*

BPX en `__vbaStrCmp` hecho esto ponemos al programa en ejecución desde el Olly con [F9]... después de varios [F9] se abrió el MP3 Cutter, ahora vayamos al link de registro y después volvamos a colocar nuestro serial falso

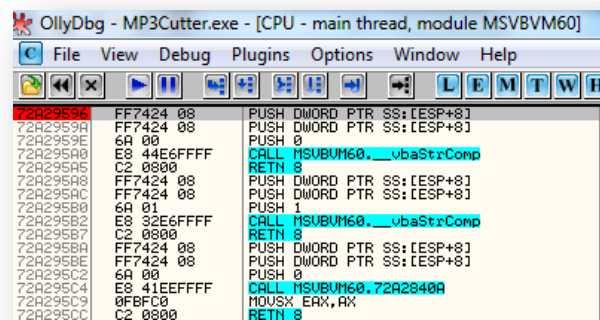


*Imagen 10. Link a la Ventana del registro*



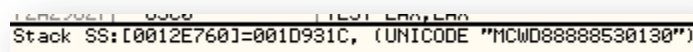
*Imagen 11. Registro y serial falso*

En la ventana (imagen anterior) damos al botón OK y... caemos acá



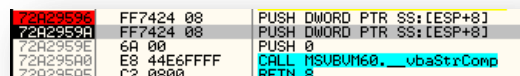
*Imagen 12. Metiendo las Strings a la pila para comparación*

La cosa se puso buena... vemos tres PUSH seguidas seguramente es cuando se forman los argumentos de la API \_\_vbaStrCmp veamos lo que está tratando de poner en la pila con ese primer PUSH



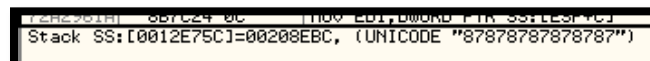
*Imagen 13. Posible serial en la ventana de ayuda de Olly*

Ahora veamos el segundo PUSH



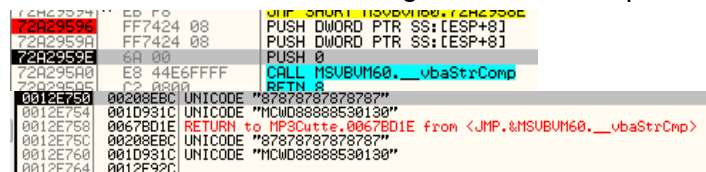
*Imagen 14. Segundo PUSH*

Y en la ventana de ayuda...



*Imagen 15. Nuestro serial falso a la vista*

Vemos nuestro serial falso, lleguemos hasta el push 0 y veamos cómo quedó el stack



*Imagen 16. Como queda el stack hasta el PUSH 0*

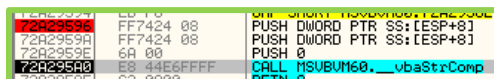


En la ventanita de ayuda de Olly podemos ver una string medio sospechosa y si vemos la ventana del stack justo arriba de esta string vemos nuestro serial falso

[IMAGEN QUITADO A PROPOSITO]

*Imagen 17. String a compararse*

¿Será la string sospechosa nuestro serial verdadero? La emoción nos llevaría a sacar conclusiones precipitadas y decir que sí, pero vayamos con calma y tracemos hasta pasar al call de la API



*Imagen 18. Llamada a la API*

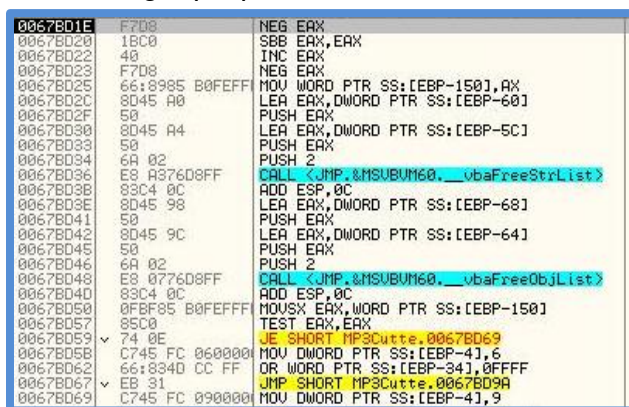
Veamos ahora como quedó el stack



*Imagen 19. Parámetros de \_\_vbaStrCmp en el stack*

Ohhh mai.. cambió las string en ASCII por Arg1, Arg2 y Arg3 que son los argumentos de la API \_\_vbaStrCmp donde los argumentos Arg2 y Arg3 son las que contienen las String que serán sometidas a comparación.

Si llegamos al RETN 4 que es la siguiente línea de código después del CALL de la API y regresamos vemos mucho código que por ahora da fiaca reversearlo



*Imagen 20. Mucho más código*

Solo tocaría probar si lo ingresado es correcto o no demos F9 para correr la aplicación... y para nuevamente en la misma línea de código de comprobación de String (ver Imagen 12). Pero ahora tenemos otra string en la ayuda de Olly será esto los ¿seriales verdaderos?... Y así va parando unas cuantas veces yo fui apuntando todos las string por si fueran los seriales correctos ☺ (MP3Cutter-serial.txt) después que termina de

comprobar nuestro serial falso con algunas string sospechosas nos muestra el siguiente cartelito



*Imagen 21. Una vez más chico malo*

Pero si reiniciamos la aplicación y probamos una string de la lista por ejemplo esto MCWD88888530130 que es la primera que apareció y ponemos nuevamente un BPX en \_\_vbaStrCmp volvemos a parar unas cuantas veces hasta llegar nuevamente a la parte que queremos si vemos el STACK ambos son iguales

[IMAGEN QUITADO A PROPOSITO]

*Imagen 21. Serial pescado en comparación*

Si damos F9 nuevamente vamos viendo las mismas string comparándose sabemos qué hace esto por la API y...



*Imagen 22. Eureka*

Eureka!! Eureka!! Esos eran los seriales veamos ahora la ventana del MP3 Cutter





*Imagen 23. Ventana de MP3 Cutter registrado*

Ahora ya no está el Unregistered en el título en cambio tenemos la versión de programa.

### 3. Despedida

Hasta acá hicimos una pesca de seriales exitosa, quizás obvie algunas cosillas pero igual la pesca fue un éxito, esta vez tuvimos suerte al toparnos con una aplicación sencillita nos ahorramos **19.95 USD**. Quiero saludar a todos los que llegaron hasta acá y a la lista completa de CracksLatinoS.

Elix

Happy Cracking!!