.-<Iv!n\$on - Ing. R3v3rS!v0>-. CLS T3aM-No CoMp3t3Nc3

Software: Sys.exe Objetivo: Desempacar. Tools: OllyDG 1.10, ImpRec, LordPE. Fecha: 06/05/2012. Cracker: Ivinson Tutorial Nº: 12 Team: CracksLatinos.

Download: http://www.mediafire.com/?we6bx2br38yz3sb



Introducción

Tenemos otro programa al cual desempacar con fines educativos porque somos una lista que busca e intenta constantemente que los integrantes puedan compartir sus conocimientos y ¿qué mejor manera que dejarlos por escritos para la posteridad? Cuando desempaques, crackees o consigas algo interesante en un software sería muy importante que lo publiques a través de un tutorial para que todos podamos seguir investigando y ampliando ese descubrimiento o novedad.

Reconozco que este arte no es tan fácil como parece, por lo menos cuando empezamos, pero poco a poco vamos desentrañando sus secretos. Esto será una lucha constante entre Programador-Cracker. ¿Quién ganará? Pienso que siempre habrá un equilibrio porque ni ellos dejarán de programar ni nosotros dejaremos de crackear.

¿Quién dejaría de hacer algo que le guste y qué le ponga retos interesantes?

Quiero cerrar esta introducción con la frase: Más sabe el diablo por viejo que por diablo.



Buscando un packer

Nuestro amigo RDGMAX creó un detector de packer muy bueno.

¿Quién no ha utilizado RDG Packer Detector? Revisemos nuestro PE a ver con qué sorpresa nos saldrá.

ASProtect v1.2x (New Strain)

En este PE, RDG no nos dice con qué está compilado. Por ahora, carguémoslo en Olly y veamos su EIP.

00401000	68	01D	PUSH	Sys.0139D001
99491995	E8	919	CALL	Sys.0040100B
0040100A	C 3		RETN	Notice and the second states
0040100B	C 3		RETN	

Detenemos el análisis con la tecla Espacio. 😊



Podemos buscar en memoria "M" a ver si conseguimos algo que nos indique con qué está compilado.





Podemos observar la DLL de Visual Basic "msvbvm60"

733A0000	00001000	msvbvm60		PE header
733A1000	000FC000	msvbvm60	.text	code,import:
73490000	000000000	msvbvm60	ENGINE	code,data

Volvamos al desensamblado de Olly.

WHC/K

El OEP de un programa compilado en Visual Basic tiene esta apariencia:

00403162	EE25 58124000	JMP DWORD PTR DS:[401258] msvbvm60.ThunRTMain
00403168	68 70E44800	PUSH AudioCut.0048E470 ASCII "VB5!6&VB6FR.DLL"
0040316D	E8 FØFFFFFF	CALL AudioCut.00403162 JMP to msvbvm60.ThunRTMain
00403172	0000	ADD BYTE PTR DS:[EAX],AL
00403174	0000	ADD BYTE PTR DS:[EAX],AL

- 1) Un PUSH a la dirección de la DLL "VB5!6&VB6FR.DLL"
- 2) Un CALL al salto indirecto de la DLL "msvbv60.ThunRTMain"

Para encontrar el OEP en nuestra víctima buscaremos la String "ThunRTMain" presionando CTRL+G.

Enter expression to follow		
ThunBTMain		-
VA / API	OK.	Cancel

Caemos aquí:

73383584	55	PUSH EBP
733A35A5	8BEC	MOU EBP,ESP
таанарит	он гг	PUSH -1



Anotemos esta dirección 733A35A4. Reiniciemos Olly con CTRL+F2.

Y vayamos a esa dirección con CTRL+G.

Enter expression to follow		
733A35A4		•
• VA / API	OK	Cancel

Luego, le ponemos un HBP en Execution. Presionamos F9 y caemos en:

73383584	55 PUSH EBP						
73383585 73383587	88EC MOV EBP,ESP 60 FF PUSH -1						
EBP=41F0	75.00						
Address	Hex dump	ASCII	^	0012FFBC	01F9 0280	RETURI	to 01F90
01372000 01372008	80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80	3		0012FFC0 0012FFC4	00459004 7C817067	ASCII RETURN	"VB5!6&*" to kerne

En el Stack vemos algo interesante: "VB5;6&*" el cual está incompleto.

Debería ser así: "VB5!6&VB6FR.DLL". Por los momentos, solo anotemos la dirección interesante que vemos en el Stack 459004.



Buscando el OEP

Reiniciamos con CTRL+F2. Damos run con F9 y presionemos CTRL+B. Busquemos FF25 que son saltos indirectos.

Enter binary string to search for				
ASCII	×			
UNICODE	[
HEX +00	FF 25			
	9			

Caemos en:

0045845F	00FF ADD	BH , BH	12
00458461	25 BC1 AND	EAX,40118C	
00458466 -	FF25 9 JMP	DWORD PTR DS:[401294]	msvbvm60vbaExceptHandler
0045846C -	FF25 E JMP	DWORD PTR DS:[4012E4]	msvbvm60vbaFPException

Vayamos al final de esos JMP con CTRL+L o como ustedes quieran.

Y nos encontramos el OEP:

00458850 -	FF25 A	JMP DWORD PTR DS:[4013A4]	msvbvm60.ThunRTMain
00458862	0000	ADD BYTE PTR DS:[EAX],AL	
00458864 -	E9 D37	JMP 01F9023C	
00458869	84	MOUS BYTE PTR ES:[EDI],BYTE PTR DS:[ESI	
00458B6A	65:40	INC EAX	Superfluous prefix
00458B6C 🗸	70 78	JL SHORT Sys.00458BE9	
0045886E	C9	LEAVE	
00458B6F	0058 0	ADD BYTE PTR DS:[EAX],BL	
00458872	0000	ADD BYTE PTR DS:[EAX],AL	
00458874	3000	XOR BYTE PTR DS:[EAX],AL	
00458876	0000	ADD BYTE PTR DS:[EAX],AL	
00458B78	50	PUSH EAX	



Vemos que hay Stolen Bytes e instrucciones raras que no se parecen en nada a un OEP de un Visual Basic normal.

Entonces, nuestro OEP temporal será: 458B5C. Reinicimos Olly con CTRL+F2. Luego, CTRL+G y vamos a 458B5C.

0045885C	90 NOP
9945885D	D7 XLAT BYTE PTR DS:[EBX+AL]
0045885E	BA 28E MOU EDX,2EB7E828
00458863	59 POP ECX
00458864	1D F15 SBB EAX,4DDD55F1

¡Que feo! ¿Verdad? Pongámosle un HBP en Execution y damos F9.

00458850 -	FF25 A	JMP DWORD PTR DS:[4013A4]	msvbvm60.ThunRTMain
00458862	0000	ADD BYTE PTR DS:[EAX],AL	
00458864 -	E9 D37	JMP 01F9023C	
00458869	84	MOUS BYTE PTR ES:[EDI], BYTE PTR DS:[ESI	
00458B6A	65:40	INC EAX	Superfluous prefix
00458B6C 🗸	70 78	JL SHORT Sys.00458BE9	
0045886E	C9	LEAVE	
00458B6F	0058 0	ADD BYTE PTR DS:[EAX],BL	
00458872	8888	ADD BYTE PTR DS:[EAX],AL	
00458874	3000	XOR BYTE PTR DS:[EAX],AL	
00458876	0000	ADD BYTE PTR DS:[EAX],AL	
00458878	50	PUSH EAX	

Ahora, sí. ⁽³⁾ Arreglemos los Stolen Bytes de la siguiente manera:

PUSH 459004 ;Dirección de "VB5!6&VB6FR.DLL". CALL 458B5C ;Dirección del salto indirecto a "ThunRTMain".

00458B5C -	FF25 A JMP DWORD PTR DS:[4013A4]	msvbvm60.ThunRTMain
00458862	68 049 PUSH Sys.00459004	ASCII "VB5!6&*"
00458867	E8 FOF CALL Sys.0045885C	JMP to msvbvm60.ThunRTMain

Al arreglar esas instrucciones, debemos poner a cero las 2 siguientes porque están muy raras.

.-<Iv!n\$on - Ing. R3v3rS!v0>-. CLS T3aM-No CoMp3t3Nc3

00458B6C	7C 78 JL SHORT Sys. 004588E9		
00458B6E	C9 LEAVE	Backup	
00458B6F	0058 0 ADD BYTE PTR DS:[EAX],	Сору	
00458B72	0000 ADD BYTE PTR DS:[EAX],	Binary	Edit Ctrl+E
00458874	3000 XOR BYTE PTR DS:[EAX],	Assemble Space	Fill with 00's

Está quedando mejor. Hasta recuperamos un POP EAX:

00458B5C -	- FF25 A	JMP DWORD PTR DS:[4013A4]	msvbvm60.ThunRTMain
00458B62	68 049	PUSH Sys.00459004	ASCII "VB5!6&*"
00458B67	E8 F0F	CALL Sys.00458850	JMP to msvbvm60.ThunRTMain
00458B6C	8888	ADD BYTE PTR DS:[EAX],AL	
00458B6E	8888	ADD BYTE PTR DS:[EAX],AL	
00458870	58	POP EAX	

Arreglemos "VB5;6&*" por esto "VB5!6&VB6FR.DLL"

00458B5C -	FF25 A JMP DWORD PTR DS:[4013A4]	msvbvm60.ThunRTMain	
00458862	68 849 PUSH Sys. 88459884	ASCII."UB5!6&*"	n
00458B67	E8 F0F CALL Sys.0045885C	JMP to Follow in Dump	Selection
00458B6C	8888 ADD BYTE PTR DS:[EAX],AL		Immediate constant

En el Dump, lo editamos así:

Address	He	x dı	JMD						ASCII
00459004	56	42	35	21	36	26	56	42	VB516&VB
0045900C	36	46	52	2E	44	40	40	00	6FR.DLL.

Arreglando la IAT

La IAT de los Visual Basic comienzan en 401000. Por lo tanto, vayamos al Dump a esa dirección y elijamos la vista Long Address.

Address	Value	Comment
88481888	73469A19	msvbvm60.EVENT_SINK_GetIDsOfNames



Vayamos al final:

00401488	734730A7	msvbvm60.rtcR8ValFromBstr
904914AC	000000000	
00401480	00140025	
00401484	0080D868	Sys.00800868

Datos recopilados:

OEP: 458B62

Inicio de IAT: 401000

Fin de IAT: 4014AC

Datos para ImpRec:

OEP: **58B62** (458B62-400000==58B62)

RVA: **1000** (401000-400000==1000)

Size: 4AC (Fin – Inicio==Size. 4014AC-401000==4AC)

Dumpeando

Abramos LordPE y seleccionemos el proceso Sys.exe.

c:\documents and settings\administrador\escritoric c:\windows\system32\notepad.exe	Nsystexe 000005 000000F 2010-000002	dump full dump partial dump region	
Path Ir	LordPE's job IntelliDumo	active dump engine Select!	Þ



Y luego:

с	dump full	critorio\sys.exe
o 🖉	dump partial	

Guardamos el Dumpeado.

Nombre:	dumped.exe	 Guardar
Tipo:	* *	 Cancelar

Arreglando el Dumpeado

Cargamos el original en ImpRec y colocamos los datos recopilados.

Allech Mark Markin Deserve	
Attach to an Active Process	6 102200200 -1
c:\documents and settings\administrador\escritorio\sys.exe (00000504) 🥌 📃	Pick DLL
Imported Functions Found	6
rva:000011F8 mod:msvbvm60.dll ord:016F name:vbaR4Str 🛛 🛛 🔒	Show Invalid
- rva:000011FC mod:msvbvm60.dll ord:0230 name:rtclsNull	
rva:00001200 mod:msvbvm60.dll ord:00CF name:vbaDateR8	Show Suspect
rva:00001204 mod:msvbvm60.dll ord:015F name:vba0bjVar	<
rva:00001208 mod:msvbvm60.dll ord:0164 name:vbaPrintObj	
rva:0000120C mod:msvbvm60.dll ord:0086 name:vbal2l4	Auto Trace
rva:00001210 mod:msvbvm60.dll ord:0231 name:rtclsNumeric	
 rva:UUUU1214 mod:msvbvm60.dll ord:U232 name:rtclsUbject 	Clear Imports
rva:00001218 ptr:01300060	Cieal miports
Log	
IAT read successfully. 📉	
Current innexter	ClearLog
0 (decimal:0) valid module(s)	
12B (decimal:299) imported function(s). (added: +12B (decimal:+299))	
(1 (decimal:1) unresolved pointer(s)) (added: +1 (decimal:+1))	1
	Options
2 IAT Infos needed New Import Infos (IID+ASLII+LUADER)	
UEP 00058862 IAT AutoSearch RVA 00000000 Size 00000028	10000
	About
Add new section	Ewit
	EXI



Vemos que solo tenemos una entrada mala. La de siempre. Le damos doble click y la escribimos a mano.



Ahora, le damos a Fix Dump y buscamos el Dumpeado.



Lo ejecutamos y:

MENSAGEM Não foi localizado o arquivo	Contate nosso suporte
ysMarket\GetDiskSeri para obter o Nº do Serial do Físico do seu Hard Disk.	Unpacked by Ivinson



Gracias por leer.

Contacto: ipadilla63@hotmail.com