1/6

Manual Unpacking IcedID Write-up

kienmanowar.wordpress.com/2020/08/16/manual-unpacking-icedid-write-up/

August 16, 2020

BytePlot (left) 0xFF 0x00 visible ASCII

Sample hash:

SHA256: 76cd290b236b11bd18d81e75e41682208e4c0a5701ce7834a9e289ea9e06eb7e

Tools:

- PE files static analysis: PortExAnalyzer; PE-bear
- Debugger & plugin: <u>x64dbg</u> + <u>ScyllaHide Anti-Anti-Debug</u>
- Aplib decompress: <u>aplib-ripper</u>

1. Static Analysis

Thow the sample to **PortEx Analyzer**, tool will analyse file with a special focus on malformation. We get the results:

invisible ASCII non-ASCII Entropy (middle) 0.2 (repetition) 0.5 (code) 0.8 (packed) e de l'Arager de le PE Structure (right) MSDOS Header COFF File Header Optional Header Section Table .text .rdata .data COMPARENTS: and an internal of .reloc Resource Table Imports Relocations Debug Info RT_VERSION RT_MANIFEST RT_ICON RT_GROUP_ICON

The section .text has high entropy, so may be the sample is packed:

	1text	2rdata	3data	4rsrc
	~			
Entropy	6.74	4.97	5.29	3.73
Pointer To Raw Data	0x400	0x10c00	0x17c00	0x19e00
Size Of Raw Data	0x10800	0x7000	0x2200	0x8400
Physical End	0x10c00	0x17c00	0x19e00	0x22200
Virtual Address	0x1000	0x12000	0x19000	0x34000
Virtual Size	0x106ab	0x6f4c	0x1a220	0x831c
-> actual virtual size	0×11000	0x7000	0x1b000	0×9000
Pointer To Relocations	0x0	0×0	0x0	0x0
Number Of Relocations	0×0	0x0	0x0	0×0
Pointer To Line Numbers	0×0	0x0	0×0	0×0
Number Of Line Numbers	0x0	0×0	0x0	0x0
Code	x			
Initialized Data		x	x	x
Execute	x			
Read	x	x	x	x
Write			x	

This sample is PE32 with **ASLR enabled** (can quickly disable this feature by using <u>setdllcharacteristics</u>):

Magic Number: PE32, normal executable file Entry Point is in section 1 with name .text DLL Characteristics * DLL can be relocated at load time. * Image is NX compatible. * Terminal Server aware. Subsystem: The Windows graphical user interface (GUI) subsystem

This sample reveals information about the pdb path:

Section Table

Debug Information

Some anomalies were identified by **PortEx**:

Anomalies

* Import function typical for code injection: VirtualProtectEx may set PAGE_EXECUTE flag for memory region

* Import function typical for code injection: CreateThread is used to open and execute a thread in the victim process

2. Dynamic Analysis

Load specimen to **x64dbg**, for unpacking process, we set breakpoints at some common APIs:

- VirtualAlloc
- VirtualProtect
- CreateProcessInternalW
- WriteProcessMemory

CPU	🍨 Graph 🛛 🗋 Log 📄 Notes	🖲 Breakpoints 🗰 Memory Map 🌐 Call Stack 🧠 SE	l 🛛 🙍 Scrá	pt 🎱 Symbols 🔿 Source 🔑 References	😒 Threads 💦
Туре	Address	Hodule/Label/Exception	State	Disassembly	Hits
Software					
	77845ED8	<pre><kernel32.dl1virtualallocstub@16></kernel32.dl1virtualallocstub@16></pre>	Enabled	nov edi, edi	0
	77847078	<pre><kernel32.dl1virtualprotectstub016></kernel32.dl1virtualprotectstub016></pre>	Enabled	nov edi, edi	0
	778648E8	<pre><kernel32.dllcreateprocessinternalwstub@48></kernel32.dllcreateprocessinternalwstub@48></pre>	Enabled	nov edi, edi	0
	77866528	<pre><kernel32.dllwriteprocessmemorystub020></kernel32.dllwriteprocessmemorystub020></pre>	Enabled	nov edi, edi	

After placing the breakpoints like above picture, press **F9** to execute. First hit at **VirtualAlloc**:

			· · · ·	
77845ED8 <_VirtualAllocStub@16>	8BFF	nov	edi, edi	
77845ED2	55		ebp	
77845ED3	8BEC	nov	ebp. esp	
77845ED5	5D	pop	ebp	
7704SED6 <jmp.6_virtualalloc016></jmp.6_virtualalloc016>	- FF25 04130877	jnp	dword ptr ds:<<&_VirtualAlloc@16>	JHP.&_VirtualAlloc@16
778ASEDC	CC	lint3		

Execute till Return (Ctr1+F9) and Follow in dump the allocated memory (return in EAX register):

🖴 🖸 🖬 🔶 🖬 🔮 🗛 🖷 🖉 🍐 🔮 🖬 🖉	😸 💣 🥜 fx # 🗛 👗 📗					
🔤 CPU 🌻 Graph 📝 Log 🕥 Notes 🔹 Bu	reakpoints - Memory Map	🚺 Call Stack 🛛 🖷 SEH	📃 Script 🛛 🎱 Symbols	O Source 🖉 Referen	ces 😒 Threads	💼 Handles 💡 Tr
C 1P 7457755C 74577595 74577596 74577596 74577596 74577596 74577596 74577596 745775968 745775968 745775968 6	C2 1998 80C8 E8 6A29FEFF - E8 F8 CC CC CC CC CC CC	ret mill nov ecx, eax ecx, eax CaseSetti into into into into into into into	ustMTErrorQ&>		^ ~	Hide FPU EAX EDX FFAMES6 ECX 77671E0 EDX Coortao EBP 00305430 ESP 00305470 Default (stdcall)
.text:7617E95C kernelbase.dll:\$10E95C B1	00555	inducate 9 Struct		actor are	F0010CA27	1: [esp+k] 00000 2: [esp+8] 00000 3: [esp+C] 00003
Index Inc. Inc. Inc. 00C40000 100 00				0130F108 0130F104 0130F108 0130F108 0130F108 0130F108 0130F118 0130F118	00000000 00000000 00000000 00000000 0000	new_iced.00100E5- return to new_ic

Continue run with **F9**, hit the second call to **VirtualAlloc** and observe changes in the allocated memory. We see new bytes value was written to this location and it is likely a shellcode:

ESI 🔹	00C40000	98	nop
	88C48881	E8 0000000	call 0xC40006
	88C48886	58	pop ebx
	00C40007	8D43 31	lea eax. dword ptr ds:[ebx+0x31]
	00040000	BF 04F6027B	nov edi. 0x7802F604
	88C4888F	89 46898888	nou ecx 8x94C
	00040014	89FA	nou edv. edi
	88648816	3108	vor ehv ehv
	00040010	89CE	
	<	0702	
	×		
00C40000			
Sili Dumo 1	Dumo 2 III Dumo 3 III Dumo 4 III	Dump E Match 1 Iv-I	locale () Staust
Att only 1	Rif numb z Rif numb 2 Rif numb 4 Rif	Dump 5 🐨 watch 1 [x=]	
Address Hex	t	ASCII	
00040000 90	E8 00 00 00 00 5B 8D 43 31 BF 04 F	6 02 7B B9 🛃è[.C12	ö.{'
00C40010 4C	09 00 00 89 FA 31 DB 89 CE 83 E6 0	13 75 0A 89 Lú1Ü.Î.a	
00C40020 FB	66 81 DA C1 CA 83 89 D7 38 18 48 C	1 CA 08 E2 ÛF.ÚÁÊ×0.0	AÊ.3
00C40030 E7	E9 BC 04 00 00 90 68 5C 34 2E 89 0	9 00 00 B8 cé%h\4.	
00C40040 1A	00 00 00 1A 00 00 D3 0C 00 00 50 7	C 01 00 1CÓ	1
00040050 00	00 00 3A DA 1C 6D 0E 18 0E 05 03 0	12 88 80 86:Ú.m	
88648868 83	03 00 00 00 04 14 07 08 0F 05 07 0	0 00 00 F8	
88648878 97	A1 00 00 00 00 00 07 00 00 00 00 00	0 0F 00 3C	
00010000 01			

Once again, **Ctrl+F9** and Follow in dump the new allocated memory:

🖬 CPU 🌻 Graph 🗋 Log 🕥 Notes 🔹 Breakpo	xints 🔳 Memory Map 🛛 Call S	tack 🔫 SEM 🗵 Script 🌒 Symbols 🔿 Source	🖉 References 🛛 😒 Threads	📥 Handles	e ⁹ Trace 🔹 NFO 🗜
1P 7617E950	C2 1080	8418		^	Hide FPU
• 7617E957	EB 6429FEFF CIT	C_BaseSetLastHTError940			EAX DESIDE
• 7617E966	- EB FO	@x7617E958			EBX 0053F000
7617E968 7617E969	CC int3				ECK Provilec
7617E968	CC int3				EBP 0830F130
7617E968 7617E968	CC int3				ESP 0830F108
■ 7617E960	cc int3				ES1 00C40200
7617E96E	CC int3				01 00040070
761712461	1413		No. of the local district in the local distribution of the local distribution of the local distribution of the	~	C10 34170000
C				,	Default (stdcall)
					1: [esp+k] eacacaca
text TATOR beauthers diliting the second					3: [esp+C] 00003000
TERCITOTIENSE REPORTBASE BITTSTURYSE RINNESS					
💱 Dump 1 💱 Dump 2 💱 Dump 3 💱 Dump 4 💱	Dump 5 💮 Watch 1 🔤 Locals	2 Struct	88335F185 08888888	return to de	CAGE From TTT
Address Hex	ASCII		0000F108 000001C		
			0033F10C 02083208		
acc5ac2a ca ao ca ao ca ao ca ao ca ao ca ao	00 00 00 00		0030F110 0000060 0030F115 00CA0830		
00C50830 00 00 00 00 00 00 00 00 00 00 00 00	08 00 08 00		0000F118 080FF92F	return to ne	u_iced.000FF92F fro

Let's continue execute and hit the third call to **VirtualAlloc**, some bytes were written to the new allocated memory. They do not look like shellcode but could be some data that malicious code uses:

😐 CPU	🌳 Graph 🛛 🍃	Log 🛐 Notes	• Breakpoints 🛛 📟	Memory Map 📋	Call Stack	🗠 SEH	o Script	Symbols	Sou
EDI	00050008 00050002 00050008 00050008 00050009 00050000 00050000 00050013 00050013 00050013 00050014 00050014 00050014 00050014		23FC 38 06 34 0 0 11 07 EF 0 06 7 52 63 07 0 100 0 0000 0 0005 6990 0 0000 0 0000 0 0000	1 05 06113271 14 0000000 00000	and ed cnp by xor al add dw add by push ed arpl wo add by add by inul ed add by add by add by	i, esp te ptr ds: , 0x1 ord ptr ds , eax te ptr ds: x rd ptr ds: te ptr ds: te ptr ds: te ptr ss: x, dword p te ptr ds: te ptr ds: te ptr ds: te ptr ds: te ptr ds:	<pre>[esi], al s:[edi], ea [edi+0x5], [edi+0x713 s:[eax], ea [eax], al [ebp+0x14] btr ds:[eax], al [eax], al [eax], al</pre>	x ah 21106], sp x , ah], 0x0	
edi=00C50 esp=003DF	1000 104								
00050000									
Ump 1	🚛 Dump 2	🚛 Dump 3 🛛 🚛 Du	mp 4 🗱 Dump 5	😸 Watch 1 🛛 💷	Locals 🤰 🤌	Struct			
Address	Hex			ASCII					
00C50000 00C50010 00C50020 00C50030	23 FC 38 96 3 11 32 71 91 9	34 01 01 07 EF 0 80 80 80 80 65 1 80 80 80 80 80 80 8 80 80 80 80 80 80 8 80 80 80 80 80 80 8	8 67 05 52 63 A7 4 69 98 88 88 88 8 88 88 88 88 8 88 88 88 88 8	96 1 08.4ï.g 98 .2qe.i 98 98	.Rc§.				

Continuing to execute the call to the **VirtuallAlloc** function, we have a newly allocated memory:

🔲 CPU 🌻 Graph [] Log [] Notes 🔹 Breakpoin	nts 💻 Memory Hap 👩 Call Stack	📲 SEH 🙍 Script 🔮 Symbols 🛛 Source	₽ References	😒 Threads 🛛 📩 Handles	g ³ Trace 🚳 NFD 📑
EEP 7617E950	C2 1698 ret #	6.50		^	Hide FPU
26172951	88C8 E8.6429FFFF (311) (CX, CAX BaseSetLastNTErrorBb)			105
■ 7617E966	- EB FB ing B	x7617E958			EBX 0053F080
7617E968	CC int3				ECX 7787110C
76175969	CC int3				EDX CONTINUE
7617E968	cc int3				ESP 8830F184
7617E96C	CC int3				EST 80C48038
76175965	CC int3				ED1 80C58080
Chammer and Chammer an	- The second sec			· · · · · · · · · · · · · · · · · · ·	
					Default (stdcall)
					1: [esp+k] 00000000
					3: [esp+C] 00003000
.text:/61/LYSC kernelbase.dll:510LYSC #100DSC					
Bung 1 Bung 2 Dung 3 Bung 4 Bung 1	Dump 5 👹 Watch 1 🔅 Locals 🕻	Struct	000007400	BOCA8668 return to B	ACABASS From 177
Address Nex	ASCII		0830F18C	0001183	
000000000000000000000000000000000000000	00 00 00 1		0830F118	00003000	

Press F9, we break at VirtualProtect. The newly allocated device has been filled with bytes. I spotted a PE file that has been compressed using *aPlib* because the PE magic bytes MZ become M8Z.

Address	Hex	<															ASCII
000660000	4D	38	5A	90	38	03	66	02	04	09	71	FF	81	B8	C2	91	M8Z.8.fqÿ.,Â.
00C60010	01	40	C2	15	Có	C8	09	1C	ØE	1F	BA	F8	00	84	09	CD	.@Â.ÆÈºø.′.Í
00C60020	21	B8	01	4C	CØ	ØA	54	68	69	73	20	ØE	70	72	бF	67	!,.LÀ.This .prog
00C60030	67	61	6D	87	63	47	бE	1F	4F	74	E7	62	65	AF	CF	75	gam.cGn.Otçbe Ïu
00C60040	5F	98	69	06	44	4F	7E	53	03	6D	бF	64	65	2E	ØD	89	i.DO~S.mode
00C60050	ØA	24	4C	44	89	01	9B	D8	84	CD	FA	Bó	D7	58	64	BE	.\$LDØ.Íú¶×X.¾
00060060	ØA	98	B7	D6	CØ	ØC	BC	7C	60	EE	11	2B	9E	BE	D6	43	öà.¼ `î.+.¾ÖC
00C60070	C8	3C	84	22	CC	ØA	52	69	63	68	28	21	8C	50	50	45	È<'"Ì.Rich(!.PPE
00060080	80	4C	01	ΑØ	C6	53	74	2B	9C	5D	14	1C	EØ	07	02	01	.L. ÆSt+.]à
00C60090	ØB	23	ØE	ØC	83	ØÅ	76	1 B	A4	14	33	3D	16	ØB	10	2B	.#v.×.3=+
00C600A0	09	20	Еó	ΑØ	ØC	40	02	05	ΕØ	01	D 0	41	08	Aó	A2	ΑE	.æ.@à.ÐA.¦¢®
00C600B0	15	88	1F	40	80	D Ø	53	2C	91	08	DA	ØF	1E	80	20	0C	@.ÐS,Ú
00060000	21	49	78	2D	E9	9C	D7	8C	2B	01	56	89	A8	94	5A	1F	!Ix-é.×.+.V.".Z.
00C600D0	C1	2E	74	65	78	CE	22	32	09	B9	91	ØÅ	4E	B8	42	43	Á.texî"2.'N _. BC
00C600E0	CØ	60	2E	72	64	61	72	74	80	68	64	ΑE	FC	65	06	09	À`.rdart.h.®üe
0.002 8.002 0	0E	20	0.2	79	E 2	25	97	1.0	CC	62	60	AD	9.0	he	AE	1.6	

Follow this section in the **Memory Map** and dump it to file:

61											
0(File name: new_iced_00C60000.bin										
0(6(Save as	type:	Binary files (*	.bin)							
0(
0(01	∧ Hide Folders	5									
0 L											
99	C40000 0	0001	999	,							
00	C60000 0	0005	000								
66	CA0000 0	0003	999								

3. Decompress dumped file

From the command line, simply need to pass dumped file to aprip.py . The tool will do its job and each extracted file will be written to a file "**dump0.bin**", "**dump1.bin**", …

入 Cmder → Downloads	
APLIB RIPPER 1.2	
Ripping PE files, this may take	some time

Check dump0.bin (21dd005162c62af26f3f59e2ebcb345c) with PE-bear: AddressOfEntryPoint = 0x0000163D

8C

 Sections Sections text 		A8D A9D	ABD FF 33 51 FF 15 88 20 40 00 88 20 68 20 40 00 88 9 3 0 9 8										
EP = A	3D	Disasm: .text		General DOS Hdr Rich H		ich Hdr	File Hdr C	ptional Hdr	Imports	orts 🖿 BaseReloc.			
, reloc		163D E087FEFFFF ▲					CALL OX4014F9						
		1644 164A		FF15C0204000 V			CALL DWORD			(KERNEL32.dll).ExitProcess			
+ 2		1648		032010			SUB ESP, 0	C1C					
Name	Raw Addr.	Raw size	Virtua	l Addr.	Virtual Siz	e Cł	aracteristics	Ptr to F	Reloc. Nu	m. of Reloc.	Num. of Linenum.		
> .text	400	A00	1000		932	60	000020	0	0		0		
> .rdata	E00	600	2000		468	40	000040	0	0		0		
> .data	1400	400	3000		250	CO	000040	0	0		0		

Valid IATs:

> .reloc

Offset	Name	Func. Count	Bound?	OriginalFirstThun	TimeDateStamp	Forwarder	NameRVA
FOC	ADVAPI32.dll	1	FALSE	2184	0	0	2228
F20	SHELL32.dll	1	FALSE	21D8	0	0	224A
F34	KERNEL32.dll	18	FALSE	218C	0	0	2356
F48	WINHTTP.dll	10	FALSE	21EC	0	0	2438
F5C	USER32.dll	2	FALSE	21E0	0	0	245C
<							
CERNEL32.d	I [18 entries]	1					
Call via	Name	Ordinal	Original Thunk	Thunk	Forwarder	Hint	
2008	IstrcpyA	-	231A	231A	-	62D	
200C	ExitProcess	-	2326	2326	-	15C	
2010	CreateDirectoryA	-	2334	2334	-	B4	
2014	IstrcatA	-	2306	2306	-	624	
2018	Sleep	-	2312	2312	-	575	
201C	IstrienA	-	22FA	22FA	-	633	
	ReadFile	-	2256	2256	-	46C	
2020				2262		245	
2020 2024	HeapFree	-	2262	2202	-	340	

m4n0w4r