## **Reversing Cerber - RaaS**

E rinseandrepeatanalysis.blogspot.com/2018/08/reversing-cerber-raas.html

## James Haughom

Exe	einfo PE - ver.0.0.4.9 by A.S.L - 1008+64 sign 2018.01.16	_		×		
	Eile : cerber.exe	<i>"</i> ₽ <u>н</u>				
	Entry Point : 00003670 00 < EP Section : .text	6	2			
6	File Offset : 00002A70 First Bytes : 81.EC.D4.02.0(	•		Plug		
-	Linker Info : 6.00 SubSystem : Windows GUI	PE	3	$\widehat{\mathbf{\Omega}}$		
	File Size : 0004341Ch < № Overlay : 0003941C	0	12 E	S.		
	Image is 32bit executable         RES/OVL : 0 / 85 % 2014	如	Sec			
1	Found I sign 2631 : [Nullsoft PiMP Stub -> SFX]	Scan / t	3.22	Rip		
1	Lamer Info - Help Hint - Unpack info		- 11 F	1423.03		
0.0	try : 7zip to view files - www.7-zip.org or Total Commander + plugin - I 😒 💈					
	The second s		- 53	1		

Cerber has established itself as one of the most successful ransomware families to date. Distributed as Raas (Ransomware as a Service), the malware has retained popularity with over 6 known variants.

The malware is packed with Nullsoft PiMP (plugin Mini Packager) which hides much of the malware's true functionality.

einfo PE - ver.0.0.4.9 by A.S.L - 1008	_		×		
Eile : cerber.exe	Section 199		₽н	3	
Entry Point : 00003670 00 <	EP Section :	.text	1		
File Offset : 00002A70	First Bytes :	81.EC.D4.02.00	•		Plug
Linker Info: 6.00	SubSystem :	Windows GUI	PE	-	
File Size : 0004341Ch < 🛚	Overlay :	0003941C	0	1	S.
Image is 32bit executable	RES/OVL : 0	<b>/ 85</b> % 2014	M		
Found I sign 2631 : [Nullsoft PiMP Stub	Scan / t	1.22	Rip		
Lamer Info - Help Hint - Unpack info		0000		11-1	1422030
try: 7zip to view files - www.7-zip.org	or Total Comm	ander + plugin - I	V 7		≥>
	einfo PE - ver.0.0.4.9 by A.S.L - 1008 File : cerber.exe Entry Point : 00003670 oo < File Offset : 00002A70 Linker Info : 6.00 File Size : 0004341Ch < № Image is 32bit executable Found I sign 2631 : [Nullsoft PiMP Stut Lamer Info - Help Hint - Unpack info try : 7zip to view files - www.7-zip.org	einfo PE - ver.0.0.4.9 by A.S.L - 1008+64 sign 201  File : cerber.exe Entry Point : 00003670 oo < EP Section : File Offset : 00002A70 First Bytes : Linker Info : 6.00 SubSystem : File Size : 0004341Ch < N Overlay : Image is 32bit executable RES/OVL : 0 Found I sign 2631 : [Nullsoft PiMP Stub -> SFX] Lamer Info - Help Hint - Unpack info try : 7zip to view files - www.7-zip.org or Total Comm	einfo PE - ver.0.0.4.9 by A.S.L - 1008+64 sign 2018.01.16 File : cerber.exe Entry Point : 00003670 oo < EP Section : .text File Offset : 00002A70 First Bytes : 81.EC.D4.02.0( Linker Info : 6.00 SubSystem : Windows GUI File Size : 0004341Ch < N Overlay : 0003941C Image is 32bit executable RES/OVL : 0 / 85 % 2014 Found I sign 2631 : [Nullsoft PiMP Stub -> SFX] Lamer Info - Help Hint - Unpack info try : 7zip to view files - www.7-zip.org or Total Commander + plugin - I	einfo PE - ver.0.0.4.9 by A.S.L - 1008+64 sign 2018.01.16	einfo PE - ver.0.0.4.9 by A.S.L - 1008+64 sign 2018.01.16 – File : cerber.exe Entry Point : 00003670 oo < EP Section : .text File Offset : 00002A70 First Bytes : 81.EC.D4.02.00 Linker Info : 6.00 SubSystem : Windows GUI PE File Size : 0004341Ch < N Overlay : 0003941C Image is 32bit executable RES/OVL : 0 / 85 % 2014 Found I sign 2631 : [Nullsoft PiMP Stub -> SFX] Lamer Info - Help Hint - Unpack info try : 7zip to view files - www.7-zip.org or Total Commander + plugin -1

As a whole, the malware's entropy is quite high (6.1), indicating packed code.

🗹 pestudio 8.74 - Malware Initial Asse	ssment - www.winitor	com
File Help		
📽 🖬 🗡 📋 🤶		
C:\users\rem\desktop\cerber.	property	value
indicators (3/12)	md5	2D6ACE7910F84EB775272A6590453A0E
virustotal (network error)	sha1	31EA288D300EE4DB56480670A6B9D2280709D157
dos-stub (!This program c	sha256	81DCFC070E35CE217BA829CD400598DA209E0953DF0BC365C06AB21BCEEE98E0
file-header (May.2014)	first-bytes (hex)	4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00 B8 00 00 00 00 00 00 00 40 00 00 00 00 00
disectories (2)	first-bytes (text)	M Z
directories (3)	size	275484 bytes
libraries (8)	entropy	6.134
imports (169/16/1/1/40)	imphash	n/a
	imphash cpu	n/a 32-bit
	imphash cpu signature	n/a 32-bit n/a
	imphash cpu signature entry-point (hex)	n/a 32-bit n/a 81 EC D4 02 00 00 53 55 56 57 6A 20 33 ED 5E 89 6C
	imphash cpu signature entry-point (hex) file-version	n/a 32-bit n/a 81 EC D4 02 00 00 53 55 56 57 6A 20 33 ED 5E 89 6C n/a
imports (169/16/1/1/40)	imphash cpu signature entry-point (hex) file-version file-description	n/a 32-bit n/a 81 EC D4 02 00 00 53 55 56 57 6A 20 33 ED 5E 89 6C n/a n/a
imports (169/16/1/1/40)     wports (0)     worts (0)     worts (0)     worts (c)     resources (6)     web strings (56/20/0/14/3307)     w☆t debug (n/a)     manifest (invoker)	imphash cpu signature entry-point (hex) file-version file-description file-type	n/a 32-bit n/a 81 EC D4 02 00 00 53 55 56 57 6A 20 33 ED 5E 89 6C n/a n/a executable
imports (169/16/1/1/40)     wports (0)     worts (0)     worts (0)     worts callbacks (n/a)     worts cstrings (56/20/0/14/3307)     works cstrings (56/20/0/14/3307)     works cstrings (invoker)     works constraints (invoker)	imphash cpu signature entry-point (hex) file-version file-description file-type subsystem	n/a 32-bit n/a 81 EC D4 02 00 00 53 55 56 57 6A 20 33 ED 5E 89 6C n/a n/a executable GUI
imports (169/16/1/1/40)      wports (0)      wo tls-callbacks (n/a)      imsources (6)      wo tsrings (56/20/0/14/3307)      with debug (n/a)      manifest (invoker)      wersion (n/a)      wersion (n/a)      with debug (n/a)      wersion (n/a)      wer	imphash cpu signature entry-point (hex) file-version file-description file-type subsystem compiler-stamp	n/a 32-bit n/a 81 EC D4 02 00 00 53 55 56 57 6A 20 33 ED 5E 89 6C n/a n/a executable GUI Sun May 11 16:05:21 2014

The malware contains an anomalous PE section ".ndata" which has a virtual size much higher than its physical/raw size. This indicates that there is a good chunk of code that won't present itself until runtime, once the malware is loaded in memory.

Sections ====================================								
========== Name Entropy	· VirtAddr	VirtSize	RawSize	MD5				
.text	0x1000	0x6594	0x6600	dd54ddfd1ff2f722ee8321d66d57b679				
.rdata 5.134399	0×8000	0x13a8	0x1400	fc77a9dcff2680a45ea84b8557e704f2				
.data 3.549948	0xa000	0x3a1358	0x1600	fca7580013b2f0103369e99ff7b40893				
.ndata 0.000000	0x3ac000 [SUSPICIO	0x10000 DUS]	0×0	d41d8cd98f00b204e9800998ecf8427e				
.rsrc 4.069419	0x3bc000	0xa18	0xc00	d31b58bbb85f4dc33599a78f3ac0eddc				

pestudio marks the .text/.code section as highly entropic (6.4), this is where the actual code/instructions are stored in a PE.

🗹 pestudio 8,74 - Malware Initial Assessment - www.winitor.com									
File Help									
📽 🖬 🗡 🗎 💡									
c:\users\rem\desktop\cerber.	property	value	value	value	value	value			
indicators (3/12)	name	.text	.rdata	.data	.ndata	.rsrc			
virustotal (network error)	md5	DD54DDFD1FF2F722EE8	FC77A9DCFF2680A45EA	FCA7580013B2F0103369	n/a	D31B58BBB85F4DC3359			
dos-stub (! I his program c	file-ratio (14.50 %)	9.48 %	1.86 %	2.04 %	0.00 %	1.12 %			
file-neader (May.2014)     entional header (GIII)	virtual-size (3905196 bytes)	26004 bytes	5032 bytes	3806040 bytes	65536 bytes	2584 bytes			
directories (2)	virtual-address	0x00001000	0x00008000	0x0000A000	0x003AC000	0x003BC000			
sections (14 50%)	raw-size (39936 bytes)	26112 bytes	5120 bytes	5632 bytes	0 bytes	3072 bytes			
libraries (8)	raw-address	0x00000400	0x00006A00	0x00007E00	0x00000000	0x00009400			
	cave (684 bytes)	108 bytes	88 bytes	0 bytes	0 bytes	488 bytes			
	entropy	6.486	5.134	3.550	n/a	4.069			
	entry-point (0x00003670)	x		-	-	-			
resources (6)	blacklisted	-	-	-	-	-			
abc strings (56/20/0/14/3307)	writable	-	-	x	x	-			
∰ debug (n/a)	executable	x	-	-	-	-			
	shareable	-	-	-		-			
1.0 version (n/a)	discardable	-	-	-	-	-			
certificate (n/a)	cachable	х	x	x	x	x			
overlay (Nullsoft)	pageable	х	х	x	х	х			
	initialized-data	-	х	x	-	х			
	uninitialized-data	-	-		x	-			
	readable	х	х	x	х	х			

I conducted this analysis in a host-only virtual network with a Windows 10 VM routing its network traffic to a REMnux VM running fakedns, iNetSim, and Wireshark. The malware happily runs in the VM with security tools running, dropping several files to disk. The number of bytes written to the file 'collages.dll' is the same as the virtual size of the '.ndata' section.

💐 Proce	💐 Process Monitor - Sysinternals: www.sysinternals.com									
File Edit	File Edit Event Filter Tools Options Help									
🚔 🔡	☞ 🖩 🛠 🕸 C) 🗢 🛆 🖲 🛤 🖡 戱 🔜 🔤									
Time	Process Name	PID Operation	Path	Result	Detail					
11:50:	📲 cerber.exe	6668 🔜 WriteFile	C:\Users\REM\AppData\Local\Temp\floppy_disk.png	SUCCESS	Offset: 0, Length: 3,676, Priority: Normal					
11:50:	Bcerber.exe	6668 🔜 WriteFile	C:\Users\REM\AppData\Local\Temp\floppy_disk_disabled.png	SUCCESS	Offset: 0, Length: 1,416, Priority: Normal					
11:50:	Bcerber.exe	6668 🔜 WriteFile	C:\Users\REM\AppData\Local\Temp\flat.xsl	SUCCESS	Offset: 0, Length: 1,858, Priority: Normal					
11:50:	Bcerber.exe	6668 🔜 WriteFile	C:\Users\REM\AppData\Local\Temp\Diacid.cpmO	SUCCESS	Offset: 0, Length: 32,768, Priority: Normal					
11:50:	. 🕞 cerber.exe	6668 🔜 WriteFile	C:\Users\REM\AppData\Local\Temp\Diacid.cpmO	SUCCESS	Offset: 32,768, Length: 32,768, Priority: Normal					
11:50:	Bcerber.exe	6668 🔜 WriteFile	C:\Users\REM\AppData\Local\Temp\Diacid.cpmO	SUCCESS	Offset: 65,536, Length: 32,768					
11:50:	. 💦 cerber.exe	6668 🔜 WriteFile	C:\Users\REM\AppData\Local\Temp\Diacid.cpmO	SUCCESS	Offset: 98,304, Length: 32,768, Priority: Normal					
11:50:	Bcerber.exe	6668 🔜 WriteFile	C:\Users\REM\AppData\Local\Temp\Diacid.cpmO	SUCCESS	Offset: 131,072, Length: 32,768					
11:50:	. 🕞 cerber.exe	6668 🔜 WriteFile	C:\Users\REM\AppData\Local\Temp\Diacid.cpmO	SUCCESS	Offset: 163,840, Length: 15,723					
11:50:	. 🕞 cerber.exe	6668 🔜 WriteFile	C:\Users\REM\AppData\Local\Temp\collages.dll	SUCCESS	Offset: 0, Length: 32,768, Priority: Normal					
11:50:	Bcerber.exe	6668 🔜 WriteFile	C:\Users\REM\AppData\Local\Temp\collages.dll	SUCCESS	Offset: 32,768, Length: 32,768, Priority: Normal					
11:50:	: Pcerber.exe	6668 🔜 WriteFile	C:\Users\REM\AppData\Local\Temp\collages.dll	SUCCESS	Offset: 65,536, Length: 17,408					
11:50:	. 🕞 cerber.exe	6668 🔜 WriteFile	C:\Users\REM\AppData\Local\Temp\nsmAD93.tmp\System.dll	SUCCESS	Offset: 0, Length: 11,776, Priority: Normal					

The malware then makes a few modifications to the registry. A couple of these modifications have to do with what the user is presented with (Wallpaper), the rest have to do with network activity. This malware (interestingly enough) does not establish persistence, just encrypts and exits.

Process Name	PID Operation	Path	Result	Detail
cerber.exe	4436 KRegSetValue	HKCU\Software\Classes\Local Settings\MuiCache\a\52C64B7E\LanguageList	SUCCESS	Type: REG_MULTI_SZ, Length: 20, Data: en-US, en
cerber.exe	4436 RegSetValue	HKCU\Control Panel\Desktop\Wallpaper	SUCCESS	Type: REG_SZ, Length: 88, Data: C:\Users\REM\AppData\
Ecerber.exe	4436 式 RegSet Value	HKCU\Software\Microsoft\Windows\CurrentVersion\ApplicationAssociationToasts\htafilehta	SUCCESS	Type: REG_DWORD, Length: 4, Data: 0
Ecerber.exe	4436 式 RegSetValue	HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\FileExts\.hta\OpenWithProgids\htafile	SUCCESS	Type: REG_NONE, Length: 0
Ecerber.exe	4436 式 RegSetValue	HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\FileExts\.hta\OpenWithProgids\htafile	SUCCESS	Type: REG_NONE, Length: 0
Ecerber.exe	4436 KRegSetValue	HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\ProxyBypass	SUCCESS	Type: REG_DWORD, Length: 4, Data: 1
Ecerber.exe	4436 式 RegSet Value	HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\IntranetName	SUCCESS	Type: REG_DWORD, Length: 4, Data: 1
Ecerber.exe	4436 式 RegSetValue	HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\UNCAsIntranet	SUCCESS	Type: REG_DWORD, Length: 4, Data: 1
erber.exe	4436 KRegSetValue	HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\AutoDetect	SUCCESS	Type: REG_DWORD, Length: 4, Data: 0
Ecerber.exe	4436 式 RegSetValue	HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\ProxyBypass	SUCCESS	Type: REG_DWORD, Length: 4, Data: 1
Ecerber.exe	4436 KRegSetValue	HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\IntranetName	SUCCESS	Type: REG_DWORD, Length: 4, Data: 1
Ecerber.exe	4436 KRegSetValue	HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\UNCAsIntranet	SUCCESS	Type: REG_DWORD, Length: 4, Data: 1
Ecerber.exe	4436 式 RegSetValue	HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\AutoDetect	SUCCESS	Type: REG_DWORD, Length: 4, Data: 0
cerber.exe	4436 RegSetValue	$\label{eq:heat} HKCU\ \ Software\ \ Microsoft\ \ Windows\ \ Current\ \ Version\ \ Explorer\ \ FileExts\ \ ipg\ \ Open\ \ WithProgids\ \ pegfile\ \ Microsoft\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ $	SUCCESS	Type: REG_NONE, Length: 0

The malware spawns an instance of itself, which then opens the ransom note '\_HELP\_DECRYPT\_N0BR8ST0\_.hta'. The filename for this ransom note appears to be unique to the system, format is '\_HELP\_DECRYPT\_[A-Z0-9]{8}\_.hta' <- simple regex for the 8 digit alphanumeric string.

Process	Description Image Path	Command
svchost.exe (6380)	Host Process f c:\windows\system32\svchost.exe	c:\windows\system32\svchost.exe -k localsystemnetworkrestricted -p -s WdiSystemHost
svchost.exe (5080)	Host Process f c:\windows\system32\svchost.exe	c:\windows\system32\svchost.exe -k localsystemnetworkrestricted -p -s PcaSvc
svchost.exe (176)	Host Process f c:\windows\system32\svchost.exe	c:\windows\system32\svchost.exe -k netsvcs -p -s Appinfo
consent.exe (3092)	Consent UI for c:\windows\system32\consent.exe	consent.exe 176 312 000001FACFF22990
svchost.exe (7088)	Host Process f C:\WINDOWS\system32\svchost.exe	C:\WINDOWS\system32\svchost.exe -k appmodel -p -s tiledatamodelsvc
WmiApSrv.exe (4864)	WMI Performa C:\WINDOWS\system32\wbem\WmiApSrv.exe	C:\WINDOWS\system32\wbem\WmiApSrv.exe
svchost.exe (3428)	Host Process f C:\WINDOWS\system32\svchost.exe	C:\WINDOWS\system32\svchost.exe -k netsvcs -p -s wisvc
Isass.exe (764)	Local Security C:\WINDOWS\system32\lsass.exe	C:\WINDOWS\system32\lsass.exe
fontdrvhost.exe (864)	Usermode Fon C:\WINDOWS\system32\fontdrvhost.exe	"fontdrvhost.exe"
csrss.exe (672)	Client Server C:\WINDOWS\system32\csrss.exe	%SystemRoot%\system32\csrss.exe ObjectDirectory=\Windows SharedSection=1024,20480,768 Win
winlogon.exe (756)	Windows Logo C:\WINDOWS\system32\winlogon.exe	winlogon.exe
fontdrvhost.exe (872)	Usermode Fon C:\WINDOWS\system32\fontdrvhost.exe	"fontdrvhost.exe"
dwm.exe (848)	Desktop Wind C:\WINDOWS\system32\dwm.exe	"dwm.exe"
Explorer.EXE (3936)	Windows Expl C:\WINDOWS\Explorer.EXE	C:\WINDOWS\Explorer.EXE
vmtoolsd.exe (5824)	VMware Tools C:\Program Files\VMware\VMware Tools\vmtoolsd.exe	"C:\Program Files\VMware\VMware Tools\vmtoolsd.exe" -n vmusr
Regshot-x64-ANSI.exe (70)	Regshot 1.9.0 C:\Program Files\Regshot\Regshot-x64-ANSI.exe	"C:\Program Files\Regshot\Regshot-x64-ANSI.exe"
🖃 对 Procmon.exe (5756)	Process Monitor C:\Program Files (x86)\Process Monitor\Procmon.exe	"C:\Program Files (x86)\Process Monitor\Procmon.exe"
Procmon64.exe (3300)	Process Monitor C:\Users\REM\AppData\Local\Temp\Procmon64.exe	"C:\Users\REM\AppData\Local\Temp\Procmon64.exe" /originalpath "C:\Program Files (x86)\Process
cerber.exe (6668)	C:\Users\REM\Desktop\cerber.exe	"C:\Users\REM\Desktop\cerber.exe"
	C:\Users\REM\Desktop\cerber.exe	"C:\Users\REM\Desktop\cerber.exe"
mshta.exe (6716)	Microsoft (R) C:\Windows\SysWOW64\mshta.exe	"C:\Windows\SysWOW64\mshta.exe" "C:\Users\REM\Desktop\_HELP_DECRYPT_N0BR8ST0hta".

Like most ransomware, a ransom note is dropped to the desktop, the Wallpaper is changed, and encrypted files are tagged with a weird file extension '.94d4'. Another file extension found during analysis was '.bde6' - this value appears to be randomly generated.



The malware contacts hundreds of hosts over port 6892.

Respuesta:	0.12.15.97.in-addr.arpa> 192.168.230.146
Respuesta:	1.12.15.97.in-addr.arpa> 192.168.230.146
Respuesta:	2.12.15.97.in-addr.arpa> 192.168.230.146
Respuesta:	3.12.15.97.in-addr.arpa> 192.168.230.146
Respuesta:	4.12.15.97.in-addr.arpa> 192.168.230.146
Respuesta:	5.12.15.97.in-addr.arpa> 192.168.230.146
Respuesta:	6.12.15.97.in-addr.arpa> 192.168.230.146
Respuesta:	7.12.15.97.in-addr.arpa> 192.168.230.146
Respuesta:	8.12.15.97.in-addr.arpa> 192.168.230.146
Respuesta:	9.12.15.97.in-addr.arpa> 192.168.230.146
Respuesta:	10.12.15.97.in-addr.arpa> 192.168.230.146
Respuesta:	11.12.15.97.in-addr.arpa> 192.168.230.146
Respuesta:	12.12.15.97.in-addr.arpa> 192.168.230.146
Respuesta:	13.12.15.97.in-addr.arpa> 192.168.230.146
Respuesta:	14.12.15.97.in-addr.arpa> 192.168.230.146
Respuesta:	15.12.15.97.in-addr.arpa> 192.168.230.146
Respuesta:	16.12.15.97.in-addr.arpa> 192.168.230.146
Respuesta:	17.12.15.97.in-addr.arpa> 192.168.230.146
Respuesta:	18.12.15.97.in-addr.arpa> 192.168.230.146
Respuesta:	19.12.15.97.in-addr.arpa> 192.168.230.146
Respuesta:	20.12.15.97.in-addr.arpa> 192.168.230.146
Respuesta:	21.12.15.97.in-addr.arpa> 192.168.230.146
Respuesta:	22.12.15.97.in-addr.arpa> 192.168.230.146
Respuesta:	23.12.15.97.in-addr.arpa> 192.168.230.146
Respuesta:	24.12.15.97.in-addr.arpa> 192.168.230.146
Respuesta:	25.12.15.97.in-addr.arpa> 192.168.230.146
Respuesta:	26.12.15.97.in-addr.arpa> 192.168.230.146
Respuesta:	27.12.15.97.in-addr.arpa> 192.168.230.146
Respuesta:	28.12.15.97.in-addr.arpa> 192.168.230.146
Respuesta:	29.12.15.97.in-addr.arpa> 192.168.230.146
Respuesta:	30.12.15.97.in-addr.arpa> 192.168.230.146
Respuesta:	31.12.15.97.in-addr.arpa> 192.168.230.146

Source	Destination	Protocol	Lengtl Info
192.168.230.146	192.168.230.147	DNS	100 Standard query response 0xd577 PTR 1.24.239.91.in-addr.arpa A 192.168.230.146
192.168.230.146	192.168.230.147	DNS	100 Standard query response 0x05ab PTR 2.24.239.91.in-addr.arpa A 192.168.230.146
192.168.230.146	192.168.230.147	DNS	100 Standard query response 0xbbdb PTR 3.24.239.91.in-addr.arpa A 192.168.230.146
192.168.230.146	192.168.230.147	DNS	100 Standard query response 0xe8de PTR 4.24.239.91.in-addr.arpa A 192.168.230.146
192.168.230.146	192.168.230.147	DNS	100 Standard query response 0xcc16 PTR 5.24.239.91.in-addr.arpa A 192.168.230.146
192.168.230.146	192.168.230.147	DNS	100 Standard query response 0x2ff2 PTR 6.24.239.91.in-addr.arpa A 192.168.230.146
192.168.230.146	192.168.230.147	DNS	100 Standard query response 0xa56d PTR 7.24.239.91.in-addr.arpa A 192.168.230.146
192.168.230.146	192.168.230.147	DNS	100 Standard query response 0xb92f PTR 8.24.239.91.in-addr.arpa A 192.168.230.146
192.168.230.146	192.168.230.147	DNS	100 Standard query response 0xdc63 PTR 9.24.239.91.in-addr.arpa A 192.168.230.146
192.168.230.146	192.168.230.147	DNS	101 Standard query response 0xa1b1 PTR 10.24.239.91.in-addr.arpa A 192.168.230.146
192.168.230.146	192.168.230.147	DNS	101 Standard query response 0xa8a9 PTR 11.24.239.91.in-addr.arpa A 192.168.230.146
192.168.230.146	192.168.230.147	DNS	101 Standard query response 0x3e18 PTR 12.24.239.91.in-addr.arpa A 192.168.230.146
192.168.230.146	192.168.230.147	DNS	101 Standard query response 0x3550 PTR 13.24.239.91.in-addr.arpa A 192.168.230.146
192.168.230.146	192.168.230.147	DNS	101 Standard query response 0x604f PTR 14.24.239.91.in-addr.arpa A 192.168.230.146
192.168.230.146	192.168.230.147	DNS	101 Standard query response 0x2773 PTR 15.24.239.91.in-addr.arpa A 192.168.230.146
192.168.230.146	192.168.230.147	DNS	101 Standard query response 0x2299 PTR 16.24.239.91.in-addr.arpa A 192.168.230.146
192.168.230.146	192.168.230.147	DNS	101 Standard query response 0x353b PTR 17.24.239.91.in-addr.arpa A 192.168.230.146
192.168.230.146	192.168.230.147	DNS	101 Standard query response 0xb3b4 PTR 18.24.239.91.in-addr.arpa A 192.168.230.146
192.168.230.146	192.168.230.147	DNS	101 Standard query response 0x7ff3 PTR 19.24.239.91.in-addr.arpa A 192.168.230.146
192.168.230.146	192.168.230.147	DNS	101 Standard query response 0x4cdb PTR 20.24.239.91.in-addr.arpa A 192.168.230.146
192.168.230.146	192.168.230.147	DNS	101 Standard query response 0xbf3a PTR 21.24.239.91.in-addr.arpa A 192.168.230.146
192.168.230.146	192.168.230.147	DNS	101 Standard query response 0xf297 PTR 22.24.239.91.in-addr.arpa A 192.168.230.146
192.168.230.146	192.168.230.147	DNS	101 Standard query response 0xfffd PTR 23.24.239.91.in-addr.arpa A 192.168.230.146
192.168.230.146	192.168.230.147	DNS	101 Standard query response 0x77b0 PTR 24.24.239.91.in-addr.arpa A 192.168.230.146
192.168.230.146	192.168.230.147	DNS	101 Standard query response 0x5435 PTR 25.24.239.91.in-addr.arpa A 192.168.230.146
192.168.230.146	192.168.230.147	DNS	101 Standard query response 0x3387 PTR 26.24.239.91.in-addr.arpa A 192.168.230.146
192.168.230.146	192.168.230.147	DNS	101 Standard query response 0x6d96 PTR 27.24.239.91.in-addr.arpa A 192.168.230.146

The same UDP packet is sent over and over.

					*eth0
File	Edit View Go Cap	oture Analyze Statistic	s Telephony Wireless	Tools Help	
	5 X A L 0 3	R O O 🖄 (+ +) 📃 🛛			
Ap	ply a display filter	<ctrl-></ctrl->			
	Time	Course	Destination	Ductored	
NO.	Time	Source	Destination	Protocol	Lengti info
	345 71.237363	192.168.230.147	91.239.24.31	UDP	67 61226 → 6892 Len=25
	346 71.237520	192.168.230.147	91.239.24.32	UDP	67 61226 → 6892 Len=25
	347 71.237526	192.168.230.147	91.239.24.33	UDP	67 61226 → 6892 Len=25
	348 71.237626	192.168.230.147	91.239.24.34	UDP	67 61226 → 6892 Len=25
	349 71.237631	192.168.230.147	91.239.24.35	UDP	67 61226 → 6892 Len=25
	350 71.237697	192.168.230.147	91.239.24.36	UDP	$6761226 \rightarrow 6892$ Len=25
	351 71.237777	192.168.230.147	91.239.24.37	UDP	67 61226 → 6892 Len=25
	352 71.237833	192.168.230.147	91.239.24.38	UDP	67 61226 → 6892 Len=25
	353 71.237868	192.168.230.147	91.239.24.39	UDP	$67\ 61226 \rightarrow 6892$ Len=25
	354 71.237917	192.168.230.147	91.239.24.40	UDP	67 61226 → 6892 Len=25
	355 71.237955	192.168.230.147	91.239.24.41	UDP	67 61226 → 6892 Len=25
	356 71.238067	192.168.230.147	91.239.24.42	UDP	67 61226 → 6892 Len=25
	357 71.238073	192.168.230.147	91.239.24.43	UDP	67 61226 → 6892 Len=25
	358 71.238074	192.168.230.147	91.239.24.44	UDP	67 61226 → 6892 Len=25
	359 71.238130	192.168.230.147	91.239.24.45	UDP	67 61226 → 6892 Len=25
	360 71.238174	192.168.230.147	91.239.24.46	UDP	67 61226 → 6892 Len=25
	361 71.238218	192.168.230.147	91.239.24.47	UDP	67 61226 → 6892 Len=25
	362 71.238263	192.168.230.147	91.239.24.48	UDP	67 61226 → 6892 Len=25
	363 71.238307	192.168.230.147	91.239.24.49	UDP	67 61226 → 6892 Len=25
	364 71.238350	192.168.230.147	91.239.24.50	UDP	67 61226 → 6892 Len=25
	365 71.238398	192.168.230.147	91.239.24.51	UDP	67 61226 → 6892 Len=25
			Wireshark · Follow UD	<b>DP Stream (u</b>	dp.stream eq 133) · wireshark_pcap_eth0_20180806101326_eFEu7K - • ·
19	01d408d1fec04469a1	100008			

The first half of the string is the same as the unique string at the end of the provided URL in the ransom note.

CERBER RANSOMWARE: Instructions			Ø )	<
CERBER RANSOMWARE	🗹 English			^
Instructions				
Can't you find the necessary files?				
Is the content of your files not readable?				
It is normal because the files' names and the data in your files have been encrypted by "Cerber Ransomware".				
It means your files are NOT damaged! Your files are modified only. This modification is reversible.				
From now it is not possible to use your files until they will be decrypted.				
The only way to decrypt your files safely is to buy the special decryption software "Cerber Decryptor".				
Any attempts to restore your files with the third-party software will be fatal for your files!				
You can proceed with purchasing of the decryption software at your personal page:				
http://p27dokhpz2n7nvgr.onion.to/191D-408D-1FEC-0446-95E5				
If this page cannot be opened click here to get a new address of your personal page.				
a ma kana a shaara a shaara a na ara a na a shaara a fam karaam kafaa				~
🗄 Search Windows	<u>^</u>	ሳ) ⋤	10:18 A 8/6/20	M 18

## LET THE REVERSING COMMENCE!!!

The process tree from behavioral analysis showed the original instance of cerber launching a new cerber, this turned out to be pretty interesting. Notice the sixth value pushed onto the stack for the CreateProcess API Call, the value 4 is passed for the dwCreationFlags argument. The 4 indicates that this process is created in a suspended state. Do I sense code injection?

mov edi,edi	CreateProcess	/	^	Hide FPU
push dword ptrss: [ebp-24] push dword ptrss: [ebp-14] push dword ptrss: [ebp-14]				EAX 0019F328 EBX 0019F326 ECX FFFF79F EDX 0000007 "PE" SDV 0000007 "PE" SDV 019F238 ESI 76C14480 <kernel32.createprocessa> EDI 02913758 "\"C:\\Users\\REM\\Desktop\\cerber.exe\"" EIP 767693C2 kernelbase.767693C2</kernel32.createprocessa>
push 0 call skernelbase.CreateProcessInternalA> pop ebp ret 28 int3 int3 int3 int3 int3			EZCO	EFLAGS 00000297 ZF 0 PF 1 AF 1 JF 0 SF 1 DF 0 EF 1 TF 0 IF 1 LastError 000000CB (ERROR_ENVVAR_NOT_FOUND) LastStatusC0000100 (STATUS_VARIABLE_NOT_FOUND)
int3 int3 int3 int3				- 0028 0053 Default (stdcall)
int3		>	1 2 3 4 5	L: [esp+4] 0000000 E: [esp+8] 02913758 "\"C:\\Users\\REM\\Desktop\\cerber.exe\"" E: [esp+10] 0000000 E: [esp+14] 0000000 E: [esp+14] 0000000
Watch 1   Image Locals 2 Struct		0019F238 0019F23C 0019F240 0019F240 0019F248 0000000 0019F248 0000000 0019F250 000000 0019F250 000000 0019F258 0000000 0019F258 0000000 0019F260 0019F268 0000000	5 0 8 0 0 0 0 4 0 0 0 0 0 0 0 0 0 0 0 0 0	return to collages.64B4A785 from ??? "\"C:\\Users\\REM\\Desktop\\cerber.exe\""
őVa	× .	<		

The malware then takes a fairly common path for the injection. Reads memory of the newly spawned cerber.exe, the parameter '2F4' is a handle to said cerber.

★x32dbg - File: cerber.exe - PID: 4776 - Module: kernel32.dll - Thread: Main Thread 2128 File View Debug Trace Plugins Executives Octions Help Mar 4 2018		- 0	×
😂 🖸 🖬 🚽 🖩 🖞 🚓 👒 🕹 🛊 🔩 📓 🖉 😔 🅢 🦧 fx #   Az 🖺 🗐 🧶	Properties -	- 0	×
CU ♥ Graph Log ♪ Nots Breakpoints Memory Map ② Cal Stack ♥ SEH ◎ Script ♥ Symbols Bis FFF move defined in 88 EC production to the set of the	General Statistics Performance Threads Token Modules Memory Environment Handles GPU Disk and Ne         Hide unnamed handles           Type         Name         Hide unnamed handles           Type         Name         Handle*           Horosoft Construction         0-2304           Process cerber-zee (1680)         0-274           Thread         cerber-zee (1680)         0-274           File         C:\Windows/fortsl\StaticZehe.det         0-226           File         C:\Windows/fortsl\StaticZehe.det         0-226           File         C:\Windows/fortsl\StaticZehe.det         0-226           Section         C:\Windows/fortsl\StaticZehe.det         0-226           Section         Sestion Home291592727         0-228           Mutant         \Sessionsl1\BaseNamedDijctsl\MSCTFMADLealAutor         0-226           Key         HKUM\SYSTEMControlse001/ControlNkij.anguage Groups         0-226           Key         HKUM\SYSTEMControlse001/ControlNkij.anguage Groups         0-226           Key         HKUM\SYSTEMControlse001/ControlNkij.anguage Groups         0-226           Key         HKUM\SYSTEMControlse001/ControlNkij.anguage Groups         0-226           File         C:\WindowSHEMAPLEHEMERSHEMAPLEHEMERTS         0-226           Key         HKUM\SYSTEMControlse001/ControlNkij.	stwork Commer	nt 🔹
Construction         Construction         Int3 cit (cit) percention         RegCloseke           edi=61 * a*         -text:76c15030 kernel32.dll:s15030 #6030          -texter         -text:76c15030 kernel32.dll:s15030 #6030          -text:77c15030 kernel32.dll:s15030 #6030          -text:77c15030 kernel32.dll:s15030 #6030          -text:77c15030 kernel32.dll:s15030 #6030          -text:77c15030 kernel32.dll:s15030 #6030          -text:77c17030 kernel32.dll:s15030 #6030          -text:77c17030 kernel32.dll:s15030 #6030          -text:77c17030 kernel32.dll:s15030 kernel330 kernel33	Key         HBLMISOFTWARE.WWWWB32Node/MicrosoftWindows(LarmetVersionExplore)         0-200           Key         HBLMISSTEMCortrolSoftWindows(LarmetVersionExplore)         0-220           Key         HBLRMISSTEMCortrolSoftWindows(LarmetVersionExplore)         0-220           Key         HBLRMISSTEMCortrolSoftWindows(LarmetVersionExplore)         0-220           Section         C:Windows/Registration/R00000000010.00         0-20           File         C:Windows/Registration/R0000000000001.4.0b         0-220           Section         UbserklamedObjectsConcatalogCache00-280         0-280           File         UbserklamedObjects/_SectD         0-280           File         UbserklamedObjects/_SectD         0-266	Clos	> se
Indeneneri 82 en el nal 43 es be nal 48 80 3e nal 44 es be nal inderæken 2007kal Command:		Defaul	it ·

The malware then hollows out the suspended process through the WinAPI call 'UnMapViewOfSection'. The parameter '400000' is passed as the base address as to where begin hollowing/unmapping, which is the very beginning/start of the PE.

🙁 v22dba - File: carbar ava - DID: 4776 - Modula: stdll dll - Thraad: Main Thraad 2129	- ñ X
File View Debug Trace Plugins Favourites Options Help Mar 4 2018	
😑 🐑 🔳 🚽 🖩 👻 🚓 👒 🌲 🛊 🦗 📓 🥔 😓 🛷 🥒 fx # 🗛 🏬	
🔤 CPU 🧶 Graph [ Log 🛐 Notes 🔹 Breakpoints 🗰 Memory Map 📋 Call Stack 🧠 S	EH 🔟 Script 🚇 Symbols 🗘 Source 🔎 References 👒 Threads 🖅 Snowman 💼 Handles 👔 Trace
B8 2A 00 00 00 mov eax,2A	eax:ZwUnmapViewOfSection, ZA: A Hide FPU
B         B	eax:ZnUmmapViewOfSection       28:'         eax:ZnUmmapViewOfSecti       EX         eax:ZnUmmapViewOfSecti       "Pt"         eax:ZnUmmapViewOfSecti       "AttimapViewOfSection"         eax:ZnUmmapViewOfSecti       "AttimapViewOfSection"         eax:ZnUmmapViewOfSecti       "AttimapViewOfSection"         eax:ZnUmmapViewOfSecti       EIP         eax:ZnUmmapViewOfSecti       EIP         eax:ZnUmmapViewOfSecti       EIP         eax:ZnUmmapViewOfSecti       EIP         eax:ZnUmmapViewOfSecti       EIP         eax:ZnUmmapViewOfSecti       EIS         eax:ZnUmmapViewOfSecti       EIS         eax:ZnUmmapViewOfSecti       EIS         eax:ZnUmmapViewOfSecti       EIS         eax:ZnUmmapViewOfSecti       EIS         eax:ZnUmmapViewOfSection       2F:'         Default (dt/cal)       * 5 : Unlocked
<	> 1: [esp+4] 000002F4 2: esp-88_000000 certer_00400000
eax= <ntdll.zwunmapviewofsection> (7765EA20) 2A '*'</ntdll.zwunmapviewofsection>	3: [esp-c] 0019F31C 4: [esp-l0] 0000004
.text:7765EA20 ntdll.dll:\$6EA20 #6DE20 <zwunmapviewofsection></zwunmapviewofsection>	5: [esp+14] 00000000
1 Dump 1 1 Dump 2 1 Dump 3 1 Dump 4 1 Dump 5 👹 Watch 1 Ix-I Locals 🌮 S	inuct 00192262 6484A006 / return to collages.6484A006 from ???
Address   Hex   ASCII	00195240 00400000 cerber.00400000
04800270488         55.90         00	0019728         0000000           0019726         0000000           0019726         0000000           0019726         0000000           0019725         0000000           0019725         0000000           0019725         0000000           0019725         0019736           0019725         0019736           0019725         0019736           0019725         0019736           0019726         0000000           0019726         0000000           0019726         0000000           0019726         0000000           0019726         0000000
Command:	Default ·

A buffer is then filled via 'RtIDecompressBuffer', which stores the contents that will be injected into the suspended process. Notice the 'MZ' header, this an executable that will be injected into the target process.

	Lange and and	Cathorne diago		
OB FF	nush ebn	ebp:RtlDecompressBuff	^	Hide FPU
88 EC FF 75 04 FF 75 05 06 C7 76 FF 75 08 06 C7 76 C 08 00 C C 08 00 C C	mov ebp(ssp push doord ptrss: ebp-4) push doord ptrss: ebp-4) push doord ptrss: ebp-8 call doord ptrds: ebp-8 call doord ptrds: ebp-8 mot 8 int3 int3	ebp:RtlDecompressBuffi ebp:RtlDecompressBuffi		EAX         76c0000         kernel32.76c00000           EBX         0000FrAE         0000FrAE           ESX         FNFFH000         1000FrAE           ESX         775C0F703 <ntdll.rtldecompressbuffer>         10000002         10000002           ESI         00000000         0000000         10000000</ntdll.rtldecompressbuffer>
edi=0 .tct:76c15080 kernel32.d11:515080 #6080 <getprocadd< td=""><td>int3 int3 int3 moved; m</td><td>GetProcessiandTeCour ebp:RtTbeconressBufFr ebp:RtTbeconressBufFr ebp:RtTbeconressBufFr GetProcessiandTeCour</td><td>*</td><td>EIP 76c15000 ckernel32.GetProcAddress&gt; EFLAGS 0000314 ZF0 0P 1 AF 1 OF 0 SF 0 DF 0 CF 0 TF 1 TF 11 LastError 000000C6 (EROR_ENVVAR_NOT_FOUND) LastStatuscologilo0 (STATUS_VARIABLE_NOT_FOUND) Cc 0032 cc 001 C 1 (sps4) 7/2600000 kernel32.76c00000 C 1 (sps4) 7/2600000 %ernel32.76c00000 C 2 (sps4) 0/286C900 °cetCommandLineA" C (sps4) 0/286C900 °cetCommandLineA"</td></getprocadd<>	int3 int3 int3 moved; m	GetProcessiandTeCour ebp:RtTbeconressBufFr ebp:RtTbeconressBufFr ebp:RtTbeconressBufFr GetProcessiandTeCour	*	EIP 76c15000 ckernel32.GetProcAddress> EFLAGS 0000314 ZF0 0P 1 AF 1 OF 0 SF 0 DF 0 CF 0 TF 1 TF 11 LastError 000000C6 (EROR_ENVVAR_NOT_FOUND) LastStatuscologilo0 (STATUS_VARIABLE_NOT_FOUND) Cc 0032 cc 001 C 1 (sps4) 7/2600000 kernel32.76c00000 C 1 (sps4) 7/2600000 %ernel32.76c00000 C 2 (sps4) 0/286C900 °cetCommandLineA" C (sps4) 0/286C900 °cetCommandLineA"
🚛 Dump 1 🚛 Dump 2 🚛 Dump 3 🚛 Dump 4	📖 Dump 5 🛛 👹 Watch 1 🛛 💷 Locals 🎾 Str	ruct 0019F28	6484A111 76c00000	return to collages.6484A111 from ???
Address Hate 07544303 Bit S - 50 0 00 03 05 00 00 04 00 00 00 00 07 FF FF 03 00 07544303 Bit S - 50 0 00 03 05 00 00 04 00 00 00 00 00 00 00 07544403 00 00 00 00 00 00 00 00 00 00 00 00 0	ASCI1 BZ	▲ 001972 001	02882980 FFFFF9 02841224 000288F 0019F318 00000000 00000000 00000000 00000000 0000	"GetcommandLineA"
Command:				Default *

The buffer is then written to the target process via 'WriteProcessMemory'. A pointer to the executable seen in the dump window is passed as the data to be written. The base address '400000', which was the start address of the hollowing, is now passed as the base address for this executable to be written to in the hollowed out process.



To intercept this executable, I followed the base address in the memory map and then dumped it. When attempting to load it into IDA, it is not recognized as a valid PE. Looking at the file in HxD, there are around 32 bytes of noise before the magic bytes of the executable.

🕬 HxD - [C:\Use	ers\REI	M∖De	skto	p\ce	rber_	0754	44000	bin.b	oak]											
📓 File Edit	Search	n Vie	w	Anal	ysis	Extr	as W	indo	w ?											
🗋 👌 🗸 🔲	inn (	H	+ +	16		~	ANSI		`	h	ex	$\sim$								
🕅 carbor 075	14000	hin	<u>.</u>	cerh	er 07	5440	000 hir	n hak	1											
and Cerber_075	++000.1	DIN		cerb	ci_07			uk												
Offset(h)	00	01	02	03	04 (	05	06 0	7 0	8 09	9 0A	0B	0C	ΟD	0E 0	F	Se Lo	oad a new file			×
00000000	9C	00	86	02	9C (	00	86 0	2 0	0 0	00	00	00	00	00 0	0	0e.t.0e.t				
00000010	00	30	ΟA	00	00	30	0A 0	0 4	C FA	A 7F	64	00	00	00 0	4	.0€Lú.d Load	file C:\Users\REM\D	esktop\cerber_07544000.	pin.bak as	
00000020	4D	5A	90	00	03	00	00 0	0 0	4 00	00 0	00	FF	FF	00 0	0	MZÿÿ Binar	ary file			
00000030	B8	00	00	00	00	00	00 0	04	0 0(	00 00	00	00	00	00 0	0		,			
00000040	00	00	00	00	00	00	00 0	0 0	0 0(	00	00	00	00	00 0	0					
00000050	00	00	00	00	00	00	00 0	0 0	0 0(	00 0	00	FO	00	00 0	0	ð				
00000060	0E	1F	BA	0E	00 1	34	09 C	D 2	1 B	3 01	4C	CD	21	54 6	8	°´.I!,.LI!Th	essor type			
00000070	69	73 :	20	70	72	6F	67 7	26	1 61	20	63	61	6E	6E 6	F	is program canno	DC (disperamble all a	onesdae) [metana]	- Cet	
00000080	74	20	62	65	20	12	75 6	E 2	0 63	9 6E	20	44	92	53 2	0	t be run in DOS Mela	ape (disassemble all o	opcodes) [metapc]	· Set	
00000090	05	04 1	01	05	2E	50	PC 7	A 2	1 51		22	41	50		2	mode	ling cogmont 000000	Analysis	Kernel ontions 1 Kernel onti	ions 2
000000080	82	FA 1	B2	77	42 1	25	BC A	7 9	2 5	V ES	77	40	F 5	BC A	7	13 a B Club 11 S a M Club	ing segment 000000	Enabled	Remeropaono 1 Remeropa	10113 2
0000000000	82	FAI	E1 .	22	43 1	75	BC A	A 4	8 81	1 E5 ) 3F	22	44	F5	BC A	Δ	uá Cola H. 2ª Dola Loadir	ing offset )000000	Indicator enabled	Processor options	
000000000	48	8D 3	2F	AA	62	F5	BC A	A 4	1 F!	5 BD	AA	2B	F4	BC A	А	H. / * bõis* Aõis* + ôis*	-			
000000E0	5A	68	13	AA	02	F5	BC A	A 5	A 68	3 21	AA	40	F5	BC A	A	Zh	ions			
000000F0	52	69	63	68	41	5	BC A	A 0	0 00	00	00	00	00	00 0	0	RichAő4ª	Loading options	Lo	ad resources	
00000100	00	00	00	00	00	00	00 0	0 0	0 0	00	00	00	00	00 0	0		Fill segment gans	V Re	name DLL entries	
00000110	50	45	00	00	4C (	01	03 0	0 B	0 A	1 77	58	00	00	00 0	0	PEL°¤wX	nin segment gaps			
00000120	00	00	00	00	E0 (	00	03 0	1 0	B 01	L OA	00	00	FA	00 0	0	àú 🗹	Create segments	Ma	nual load	
00000130	00	4A (	02	00	00	00	00 0	0 5	4 38	3 00	00	00	10	00 0	0	.JT8	Create FLAT group	Cr	eate imports segment	
00000140	00	10	01	00	00	00	40 0	0 0	0 10	00	00	00	02	00 0	0	@	Load as code segme	ent		
00000150	05	00	01	00	00	00	00 0	0 0	5 0(	0 01	00	00	00	00 0	0					
00000160	00	70	03	00	00	04	00 0	0 0	0 0(	00 00	00	02	00	00 8	1	.p		OK Cancel	Help	
00000170	00	00 :	20	00	00	10	00 0	0 0	0 0(	0 10	00	00	10	00 0	0					

Deleting up until the 'MZ'/'4D5A' fixes the problem. This looks to be where the true payload/ransomware code lies, this is the first time we have seen crypt-related APIs. So essentially, the malware uses code injection as a way to unpack itself.

TDA - cerber_07544000.i64 (cerber_07544000.bin) C:\U	ers\REM\Desktop\cerbe	r_07544000.i64									
File Edit Jump Search View Options Windows Help											
	hot.t	n m 🛛 . ko 🖚 🖉 04	• @X								
			в	_							
📃 Library function 💻 Regular function 💻 Instruction 💷 I	Imports										
Z Functions window				N-1		Structures		Enums	ø	Exports	
Function name	Address Ordinal	Name	^	-	arg_0+ dword ptr 8			1			
ant 401000	M 00000000	CryptGetKeyParam			push 6Ch						
Sub_401000	M 0000000	CryptHashData			callSEH_prolog						
Sub_401105	N 0000000	CryptDestroyHash			and [ebp+var_28], 0						
7 sub_401107	8 0000000	CryptCreateHash			push esi ; 1p5t	tring					
J SUD_401127	10000000	CryptAcquireContextW			mov [ebp+var_1C], eax						
7 Sub_401179	8 0000000	CryptGetHashParam			nov [ebp+var_24], es1 push 13453880h ; char						
7 SUD_4011D4	M 0000000	GetSecurityDescriptorOwner			push 4 ; ucb push offset unk_412230 ; vo	old •					
7 SUD_4011E9	M 0000000	CryptGenRandom			call sub_408800 add esp, 9Ch						
Sub_401212	10000000	GetFileSecurityW			nov ebx, eax						
7 Sub_401273	8 0000000	RegQueryValueExW			mov [ebp+var_20], eax			J .			
J sub_401200	8 0000000	EqualSid				147 C	1				
7 SUD_40133D	M 0000000	CryptImportKey				loc_40103F: xor edi, edi					
7 Sub_4013A0	8 0000000	CryptDestroyKey				cmp [ebp+var_1C], er	51				
7 SUD_40140D	M 0000000	OpenProcessToken				HALL THE MENT	_				
7 SUD_40140D	10000000	GetTokenInformation				<pre>cmp byte ptr [esi], 20h jnz short loc_401070</pre>					
V SUD_4014CE V	8 0000000	DuplicateToken				1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.	_				
< >>	100000000	RegOpenKeyExW				push 2 uct					
Line 1 of 362	10000000	CheckTokenMembership				call sub_4088D0	010 -				
A Graph overview	M 0000000	ConvertSidToStringSidW				push eax ; psz	Set				
	8 0000000	RegCloseKey				call ds:StrPBrkA ; ps2					
	No000000	CreateWellKnownSid				jz loc_4010FE					
	30000000	InitializeSecurityDescriptor				Ref. ov out					
,	8 0000000	SetSecurityDescriptorDacl				add esi, eax					
	81 0000000	SelectObject				sub [ebp+var_10], eax					
	N0000000	DeleteObject				1					
	N 0000000	CreateFontW			1	oc_401070:					
	M 0000000	CreateCompatibleDC				ush 48h ; Size					
	1.00000000	GetStockObject			1	ea eax, [ebp+Dst]					
	10000000	SetPixel				all nenset					
	1 00000000	SetBkColor	~	401002:	sub 401000+2 (Sync	chronized with Hex	View-1)				

Now that the code has been injected/written, the malware must start a thread of execution to invoke the code. The context of the thread is set via 'SetThreadContext'.

🗮 x32dbg - File: cerber.exe - PID: 808 - Module: kerne	el32.dll - Thread: Main Thread 5048		- ¤ ×
File View Debug Trace Plugins Favourites	Options Help Mar 4 2018		
😂 😉 🖬 🔿 🖩 🕴 🛊 🥵 😫	🕻 🔩 🛐 🥜 😓 🕢 🥠 fx # 🛛 Az 🖺 🛔	1 👳	
📓 CPU 🏾 🌳 Graph 🛛 🗋 Log 👘 Notes	• Breakpoints 💻 Memory Map 📋 Call Stack 🧠 S	EH 👩 Script 🔮 Symbols 🛇 Source 🔎 References 😒 Threads	🐨 Snowman 📥 Handles  Trace
EIP EAX S5	mov edi,edi	SetThreadContex	A Hide FPU
- 388 EC - 79 25 6 <u>114 67 76</u> - 70 25 6 <u>114 67 76</u> - 70 2 - 70 2	<pre>mov ebg.esp pop obp ints ints ints ints ints ints ints ints</pre>	SetThreadContex	EX. 76:16140         -kernel32.setThreadContext>           EX. 0050002c         -:           EX. 000002c         -:           EX. 000000c         -:           CF.0 0F 0         0F 0           CF 0         0F 0           CF 0         0F 0           CF 0         0F 0           LastFrom 0000000c         (FERDE, ENVAR, NOT, FOIND)           LastFrom 0000000c         (FERDE, ENVAR, NOT, FOIND)
CC CC 88 FF	int3 int3 mov edi,edi	SetThreadPriorit	_ee_00.20ec_00.53
• 55 • 88 FC	push ebp		V Default (stdcall) · 5 C Unlocked
<			> 1: [esp+4] 000002BC 2: [esp+8] 045F0000
edi=0 .text:76c165A0 kernel32.dll:\$165A0 #75A0 <se< td=""><td>etThreadContext&gt;</td><td></td><td>3: [esp+C] 0000006A 4: [esp+10] 00000004 5: [esp+14] 0000000</td></se<>	etThreadContext>		3: [esp+C] 0000006A 4: [esp+10] 00000004 5: [esp+14] 0000000

And finally, the thread is resumed and the code begins executing in the target process.

🗱 x32dbg - File: cerber.exe - PID: 808 - Module: collages.dll - Thread: Main Thread 5048	- s ×
File View Debug Trace Plugins Favourites Options Help Mar 4 2018	
😑 😇 🖬 🛉 👬 🛊 😒 🋬 🁔 🔩 📓 🥜 😓 🚳 🏈 🌮 🏦	
🔳 CPU 🍨 Graph 🕖 Log 🕥 Notes 🔹 Breakpoints 💻 Memory Map 🍵 Cail Stack 🧠 SEH 💿 Script 😫 Symbols 🗘 Source 🔎 References 👻 Threads of	🐼 Snowman 🔥 Handles  🐔 Trace
<ul> <li>OF 8E 05 <u>C6 11 85 64</u> movsx eax, byte pt cfs: [648511C9]</li> <li>89 04 22 4 mov sk ord pt ptrs: [esp], eax</li> </ul>	A Hide FPU
E5 \$1 \$4 Fr FF       Call collarge_stiller20         88 \$1 \$5 \$1 \$1 \$1 \$5 \$45\$       mov sk, dord ptrists [14831188]         0 \$6 \$1 \$0 \$1 \$1 \$5 \$45\$       mov sk, dord ptrists [14831184]         0 \$6 \$1 \$0 \$1 \$1 \$5 \$64\$       mov sk, sk, dord ptrists [14831184]         0 \$6 \$1 \$1 \$1 \$5 \$64\$       mov sk, sk, dord ptrists [14831184]         0 \$7 \$20\$       mov sk, sk, dord ptrists [14831184]         0 \$7 \$20\$       sh = th c, c, l         0 \$7 \$20\$       mov sk, sk, dord ptrists [14831184]         0 \$6 \$1 \$1 \$15 \$56\$       mov sk, sk, byte ptrists [14831184]         0 \$6 \$1 \$1 \$15 \$56\$       mov sk, sk, byte ptrists [14831184]         0 \$6 \$1 \$1 \$15 \$56\$       mov sk, sk, byte ptrists [14831184]         0 \$6 \$1 \$1 \$15 \$56\$       mov sk, sk, byte ptrists [14831184]         0 \$6 \$1 \$1 \$15 \$56\$       mov sk, sk, byte ptrists [14831184]         0 \$6 \$1 \$0 \$1 \$2 \$1 \$2 \$1 \$65\$       mov sk, sk, byte ptrists [14831184]         0 \$6 \$8 \$1 \$2 \$1 \$2 \$1 \$6\$       mov sk, and ptrists [14831184]         0 \$6 \$8 \$1 \$1 \$2 \$6\$       mov sk, and ptrists [14831184]         0 \$6 \$8 \$1 \$1 \$5 \$6\$       mov sk, 76\$         0 \$6 \$8 \$1 \$1 \$5 \$6\$       mov sk, 76\$         0 \$6 \$8 \$1 \$1 \$5 \$6\$       mov sk, 76\$         0 \$6 \$8 \$1 \$1 \$5 \$6\$       mov sk, 76\$         0 \$6 \$8 \$1 \$1 \$5 \$6\$       mov sk, 76\$	EAX 00000076 (1''' EAX 76C1580 - <kernel32.resumethread> ECX 00000001 EDX 000000058 'X' EBP 001973k kernel32.76C00000 ESI 76C10000 kernel32.6C00000 ED1 76C10000 -<kernel32.6c00000 EFI 76C0000 -<kernel32.6c00000 EFI 76C0000 -<kernel32.6c00000 EFI 76C0000 -<kernel32.6c00000 EFI 76C0000290 EFI 76C000290 EFI 76C0000290 EFI 76C0000290 EFI 76C0000290 EFI 76C0000290 EFI 76C0000290 EFI 76C000000000000000000000000000000000000</kernel32.6c00000 </kernel32.6c00000 </kernel32.6c00000 </kernel32.6c00000 </kernel32.resumethread>
88 84 24 F0 00 00 mov eax, dword ptrss: gesprig     89 04 24     89 04 24     mov dword ptrss: gesprig     eax	LastError 000000CB (ERROR_ENVVAR_NOT_FOUND) LastStatusC0000100 (STATUS_VARIABLE_NOT_FOUND)
Bit         Die 10         55         51.21         25.5         44         Construction and the spin of the	
	1: [esp] 000002BC 2: [esp:4] EEEEE9E
ebx= <kernel32.resumethread> (76C16280) .text:64848675 collages.dll:58875 48275 <ebx =="" resumethread=""></ebx></kernel32.resumethread>	3: [esp+3] 000006A 4: [esp+0] 0000004 5: [esp+10] 00000000

Just before taking the instruction to allow 'ResumeThread' to execute, I spawned a new instance of x64dbg and attached to the still suspended cerber.exe. I then set breakpoints on thread entry and thread start to halt execution once 'ResumeThread' is called. Next, I set break points on all crypt-related APIs, as well as GetProcAddress, so that I can identify any additional code that may be dynamically loaded. The first function to be dynamically resolved via 'GetProcAddress' is 'CryptEncrypt'.



The next interesting code block was the usage of the API 'CryptStringToBinary'. This API converts a string to an array of bytes. The data to be converted is a very long base64 string.



Decoded, the string looks to be a hard-coded Public Key.



The Public Key Info suggests that this is 'RSA 1.2.840.113549.1.1 - PKCS-1' encryption.

edi=04361138 "CryptImportPublicKeyInfo" .text:76891F20 crypt32.dll:\$31F20 #31320 <cryptimportpublickeyinfo></cryptimportpublickeyinfo>	2: [050+6] UUUUUUU 3: [050+6] UUUUUUU 4: [050+6] 00195478 5: [050+14] 0436068 "BEGIN PUBLIC KEY\nMIIBIJANBgkqhkid	59w
Imp Dump 1         Imp Dump 2         Imp Dump 3         Imp Dump 4         Imp Dump 5         Watch 1         Ivel Locals         Struct           Address 1         Hex         Struct         Struct <td>ODJEJSCE         FOMO1204         return to cerber.00401204         from 77?           0019FA40         0277A420         0019FA44         0270501           0019FA44         0270501         1.2,840,113549,1.1,1"           0019FA42         02019FA42         02019FA42           0019FA42         02019FA42         0319FA42           0019FA42         0319FA42         0319FA42           0019FA42         0319FA42         0319FA42</td> <td>B</td>	ODJEJSCE         FOMO1204         return to cerber.00401204         from 77?           0019FA40         0277A420         0019FA44         0270501           0019FA44         0270501         1.2,840,113549,1.1,1"           0019FA42         02019FA42         02019FA42           0019FA42         02019FA42         0319FA42           0019FA42         0319FA42         0319FA42           0019FA42         0319FA42         0319FA42	B

Next, the malware creates a directory in the AppData folder where it stores some housekeeping data.

IDA View-A		Strings window Hex View-1 Structures Enume Interview Int						
		test ax, ax jnz short loc_40976A						
		<pre>loc_40973E: mov ebx, eax xor eax, eax push edi ; C:\Users\REM\AppData\Local\Temp\0ad3e319 mov [esi], ax call ds:GetFileAttributesW cmp eax, 0FFFFFFh jnz short loc_40975E</pre>						
	<b>-</b>							
	push push call test jmp	0 ; lpSecurityAttributes edi ; lpPathName ds:CreateDirectoryW ; C:\Users\REM\AppData\Local\Temp\0ad3e319 eax, eax short loc_409760						
Jmp Short 102_409760: jz short 102_409772								

Then a mutex is created. The name of the mutex is resolved dynamically ---- 'shell. {FB79CB8E-F0B4-4B09-A183-601B6025EC35}' and is created just before network activity occurs ('WSAStartup').

<b>II</b> 🖌 🖼		
loc 4038	BFC:	
lea	<pre>eax, [esp+4C0h+N</pre>	ame]
push	eax	; 1p
call	sub_4035C8	
рор	ecx	
lea	<pre>eax, [esp+4C0h+N</pre>	ame]
push	eax	; shell.{FB79CB8E-F0B4-4B09-A183-601B6025EC35}
push	edi	; bInitialOwner
push	edi	; lpMutexAttributes
call	ds: <mark>CreateMutex</mark> W	
call	ds:GetLastError	
cmp	eax,0B7h	
jz	short loc_40399B	
	•	
lea	eax [esn+4(0h+W	SADatal
nush	eax	: lnWSAData
push	202h	: wVersionRequested
call	ds:WSAStartup	
push	edi	: lpName
push	edi	; bInitialState
push	1	bManualReset
push	edi	; lpEventAttributes
call	ds:CreateEventW	
1		

A UDP socket is created and will be used to blast that single string to hundreds of IPs.

.text:00404258	mov	eax, esi	
.text:0040425A	push	ecx	; hostshort
.text:0040425B	mov	[ebp+to.sa_fam	ily], ax
.text:0040425F	call	ds:htons	
.text:00404265	push	17	; IPPROTO_UDP
.text:00404267	push	esi	; SOCK_DGRAM
.text:00404268	push	esi	; AF_INET // IPv4
.text:00404269	mov	word ptr [ebp+	to.sa_data], ax
.text:0040426D	call	ds:socket	
.text:00404273	mov	[ebp+s], eax	
.text:00404276	cmp	eax, 0FFFFFFFF	h
.text:00404279	jz	loc_404430	

'sendto' function is included in a loop to contact the external hosts.

🗱 x32dbg - File: cerber.exe - PID: 6064 - Module: cerber.exe - Thread: Main Thread 5084	- 5 ×
File View Debug Trace Plugins Favourites Options Help Mar 4 2018	
🗀 😳 🔳 🔶 🖩 🍷 📯 🧐 🎍 🛊 🔹 📓 🥜 😓 🕢 🖊 Ia 👢 📓 🔮	
🔤 CPU 🧶 Graph 📝 Log 🖄 Notes 🔹 Breakpoints 🗰 Memory Map 📋 Call Stack. 🚎 5EH 😿 Script. 🀏 Symbols 🛇 Source 🔑 References 🦤 Threads 🥩 Snowman 🍓 Handles 📌 Trace	
BIO         CONCERNS         FF 15 4C 13 41 00         call dword ptrais[cdsendto]         A           Hide FPU         • (00404388) 38 45 FD         • (mo eax, donor ptrais[cdsendto]         A	
Image: Provide State         V 75 04 gram         Image: Provide State         Provide State	s • Dittected
004040cl 86 /5 EE   mov a) buta attraction 1     11 [esp] 00000268	
therd ptr [0041134C <cerber.&sendtos]ssb2.32.sendtos "d55<="" 04362488="" 21="" [esp-4]="" td="">           .text:00404385 cerber.exe:\$4385 #3785         21 [esp-4] 04367488 "d55</cerber.&sendtos]ssb2.32.sendtos>	
Image: Dump 1         Image: Dump 2         Image: Dump 3         Image: Dump 4         Image: Dump 5         Watch 1         Is-el Locals         P Struct         O001993.02         O002093.02	^

Next, the encryption commences. Traversing directories via 'FindFirstFile' and 'FindNextFile'. The malware also looks for network resources to encrypt via 'WNetOpenEnum'.

www.adata Ella coder ave DID: 6064 Madula coder ave Thread 228					
Alexandre - File. Cerbertexe - Fib. 6004 - Module. Cerbertexe - Thread. 226					
File View Debug Trace Plugins Favourites Options Help Mar 4 2018					
🗀 🕽 🔳 🔿 🖩 🚦 😓 😒 🎍 🛊 🛠 📓 🥖 😓 🖉 🥒 fx 🗰	A: 💄 🗐 🔮				
🔛 CPU 🧟 Graph 🚺 Log 🛐 Notes 🔹 Breakpoints 🛲 Memory Map 📋 Call S	ack 🧠 SEH 🔟 Script 🔮 Symbols 🗘 Source 🔎 References 👾 Threads 🎻 Snowman 📩 Handles 🥂 Trace				
312	Hide FPU				
Operation         Operation <t< td=""><td>ex:L*c:\/*       EAX 0709FAA0       L*c:\/**         EAX 0709FAA0       L*c:\/**         EXX 0709FAA0       L*c:\/**         E</td></t<>	ex:L*c:\/*       EAX 0709FAA0       L*c:\/**         EAX 0709FAA0       L*c:\/**         EXX 0709FAA0       L*c:\/**         E				
.text:00407597 cerber.exe:57597 #0997					
🚛 Dump 1 🚛 Dump 2 🚛 Dump 3 🚛 Dump 4 🚛 Dump 5 👹 Watch 1 💷 Lo	als 2 Struct 0709F63C 0709F60000000000000000000000000000000000				

Once encryption completes, the ransom note is opened via 'ShellExecute' with parameters 'Open' and '\_HELP\_DECRYPT\_N0BR8ST0\_.hta'. This ransom note is far more robust than most. Most ransom notes are a simple text file, this is an .hta that has several functions, and is apparently quite universally accommodating. The ransom note even has a button to change the language.

99	= <body></body>	
100	div	<pre>/ class="container"&gt;</pre>
101		<div class="header"></div>
102		<pre><a href="#" id="change language" onclick="return changeLanguage1();" title="English">☑ English</a></pre>
103		<h1>CERBER RANSOMWARE</h1>
104		<small id="title">Instructions</small>
105	-	
106		<div id="languages"></div>
107		<p#9745; language<="" p="" select="" your=""></p#9745;>
108	¢.	<ul><li><li><li></li></li></li></ul>
109		<pre><li><a href="#" onclick="return showBlock('en');" title="English">English</a></li></pre>
110		<li><a (a="" +"="" href="#" onclick="return showBlock('ar');" title="Arabic"></a></li>
111		<li><a href="#" onclick="return showBlock('zh');" title="Chinese">中文</a></li>
112		<li><a href="#" onclick="return showBlock('nl');" title="Dutch">Nederlands</a></li>
113		<li><a href="#" onclick="return showBlock('fr');" title="French">Français</a></li>
114		<pre><li><a href="#" onclick="return showBlock('de');" title="German">Deutsch</a></li></pre>
115		<pre><li><a href="#" onclick="return showBlock('it');" title="Italian">Italiano</a></li></pre>
116		<li><a href="#" onclick="return showBlock('ja');" title="Japanese">日本語</a></li>
117		<li><a href="#" onclick="return showBlock('ko');" title="Korean">한국어</a></li>
118		<li><a href="#" onclick="return showBlock('pl');" title="Polish">Polski</a></li>
119		<pre><li><a href="#" onclick="return showBlock('pt');" title="Portuguese">Português</a></li></pre>
120		<li><a href="#" onclick="return showBlock('es');" title="Spanish">Español</a></li>
121		<li><a href="#" onclick="return showBlock('tr');" title="Turkish">Türkçe</a></li>
122	-	
123	-	

Another interesting code segment is that the .hta file checks the victim's MAC address against a few MAC addresses associated with VMware and some popular network technology companies. If there is a match, the URL in the ransom note is updated to English. Interesting!

853	showBlock (lang) ;	
854 🗄	try (	
855	<pre>var macAddress = "";</pre>	
856	var myEnum = new Enumerator (GetObject ("winmgmts: {impersonationLevel=impersonate}"). ExecQuery ("SELECT * FROM Win32 NetworkAdapterConfiguration WHERE IPEnabled = True'	'));
857 🗄	for (; !myEnum.atEnd() ; myEnum.moveNext()) {	
858	<pre>var item = myEnum.item();</pre>	
859	<pre>macAddress = item.MACAddress.substr(0, 8);</pre>	
860 -	}	
861 😝	if ("00:50:56, 00:00:29, 00:10:14, 00:05:69, 0A:00:27, 00:03:FF, 00:10:42, 00:0F:4B, 00:16:3E, 08:00:27".indexOf(macAddress) < 0) (	
862 🖨	setTimeout(function() {	
863	updateUrl("en");	
864 -	), 100);	
865 -		
866 -	}	

The most interesting part of this malware to me was how it unpacked itself through code injection and process hollowing. This malware just performs the encryption and then exits, no persistence! Most filenames are randomized to evade signature based detection, so regex will be our friends when sweeping for/detecting these artifacts. A Snort rule may be plausible due to the port (6892), but UDP can be noisy. The rule would use PCRE to match the unique long string passed over 6892.

Key Takeaways:

- Encrypts files on disk and in network shares
- Modifies registry
- Drops files on disk
- Performs code injection
- Contacts external hosts

Host-based IOCs: cerber.exe 2d6ace7910f84eb775272a6590453a0e - md5 \AppData\Local\Temp\collages.dll 2A4BF3D01B6C84A2130C110D02C772AC - md5 \AppData\Local\Temp\floppy\_disk.png \AppData\Local\Temp\floppy\_disk\_disabled.png \AppData\Local\Temp\flat.xsl \AppData\Local\Temp\0ad3e319\4f11.tmp \AppData\Local\Temp\0ad3e319\280c.tmp \AppData\Local\Temp\nshBD76.tmp\System.dll 3E6BF00B3AC976122F982AE2AADB1C51 - md5 \Desktop\\_HELP\_DECRYPT\_N0BR8ST0\_.hta - Ransom note

\Desktop\\_HELP\_DECRYPT\_N0BR8ST0\_.jpg - Wallpaper

\*.94d4 - file extension tagged onto encrypted files (randomly generated)

\*.bde6 - file extension tagged onto encrypted files (randomly generated)

\*.[a-z0-9]{4} - regex for file extension

shell.{FB79CB8E-F0B4-4B09-A183-601B6025EC35} - Mutex

\Sessions\1\BaseNamedObjects\SM0:6064:168:WilStaging\_02 - Based named object HKEY\_CURRENT\_USER\Control Panel\Desktop\WallPaper -REG\_SZ -C:\Users\REM\AppData\Local\Temp\tmpCBD8.bmp

Hard-coded Public Key:

-----BEGIN PUBLIC KEY-----

MIIBIJANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAvkty5qhqEydR9076Fevp 0uMP7IZNms1AA7GPQUThMWbYiEYIhBKcT0/nwYrBq0Ogv79K1tta04EHTrXgcAp/ OJgBhz9N58aewd4yZBm2coeaDGvcGRAc9e72ObFQ/TME/Io7LZ5qXDWzDafI8LA8 JQmSz0L+/G+LPTWg7kPOpJT7WSkRb9T8w5QgZRJuvvhErHM83kO3ELTH+SoEI53p 4ENVwfNNEpOpnpOOSKQobtIw56CsQFrhac0sQIOjek/muVluxjiEmc0fszk2WLSn qryiMyzaI5DWBDjYKXA1tp2h/ygbkYdFYRbAEqwtLxT2wMfWPQI5OkhTa9tZqD0H nQIDAQAB

-----END PUBLIC KEY-----

Network-based IOCs: 97.15.12.xxx:6892 - UDP 91.239.24.xxx: 6892 - UDP xxx.12.15.97: 6892 - UDP

xxx.24.239.91: 6892 - UDP