

# A Detailed Analysis of The Last Version of Conti Ransomware

Prepared by: Date: LIFARS, LLC 10/08/2021

# **EXECUTIVE SUMMARY**

Conti ransomware has been sold as a RaaS (Ransomware as a Service) in underground forums and it's usually deployed by other malware such as TrickBot and BazaLoader/BazarLoader. It can run with one of the following parameters: "-p", "-m", "-size", "-log" and "-nomutex". A new mutex called "YUIOGHJKCVVBNMFGHJKTYQUWIETASKDHGZBDGSKL237782321344" can be created to ensure that only one instance of ransomware is running at a single time. The malware has the ability to only encrypt network shares ("-m net" parameter), local drives ("-m local" parameter), or both of them ("-m all" parameter). The volume shadow copies are deleted using wmic and COM objects. The algorithm used to encrypt files is ChaCha8, with the key and nonce being encrypted using an RSA public key.



## **ANALYSIS AND FINDIGS**

SHA256: 4bfd58d4e4a6fe5e91b408bc190a24d352124902085f9c2da948ad7d79b72618

The malware obfuscates the stack strings and implements multiple custom algorithms to decrypt them. An example of a decryption algorithm is shown below, along with the decrypted string:

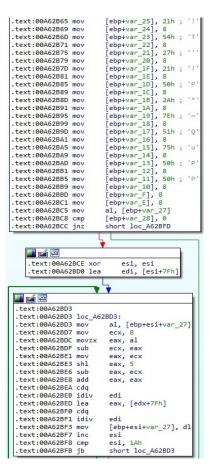


Figure 1

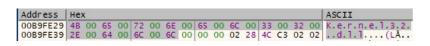


Figure 2

The relevant APIs are imported dynamically at runtime using some hashing algorithms (the first parameter is a hash value; the second parameter is an offset). The return value is placed into the EAX register:



004784AC       6A 20         004784AC       67 44 24 10 66 C1 18 00         004784BC       67 44 24 10 66 C1 18 00         004784BC       68 09 EA 93 E5         004784BC       68 89 EA 93 E5         004784BC       67 44 24 14 66 C1 18 00         004784BC       68 89 EA 93 E5         004784BC       68 89 EA 93 E5         004784BC       67 44 24 14 66 C1 18 00         004784BC       74 24 14 66 C1 18 00         004784C0       Conti A647C0         conti A647C0       Conti A647C0         .text:004784C8 conti .exe:\$184C8 #178C8         @@Dump 1       @Dump 2         @Dump 1       Dump 3         @Dump 1       Dump 2         @Dump 2       Dump 3         @Dump 5       @Watch 1         Ne=lLocals       \$2 Struct	x87StatusWord 0000 x87Sw_E8 0 x87Sw_C2 0 x87Sw_C2 0 x87Sw_E1 0 x87Sw_C0 0 x87Sw_E5 0 x87Sw_S5 0 x87Sw_D 0 x87Sw_U 0 Default (stdcall)	

Figure 3

The binary retrieves the command-line string for the process by calling the GetCommandLineW API:

00A784CD 00A784D0	83 C4 08 FF D0	add esp,8 call eax	>		Default (stdcall)
<pre>eax=<kernel32.getcommandl .text:00a784d0="" conti.exe:<="" pre=""></kernel32.getcommandl></pre>					2: [esp+4] 00A92070 conti.00A92070 3: [esp+8] 00863000 4: [esp+C] 001BC166
			00B9FE80 000	0000	00

Figure 4

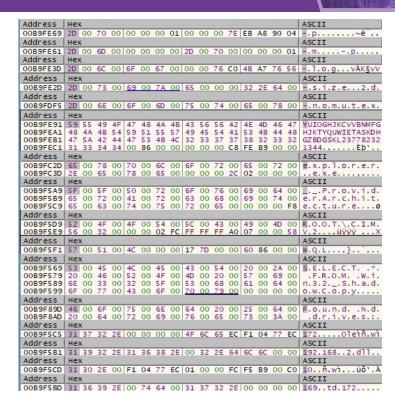
CommandLineToArgvW is utilized to extract an array of pointers to the command line arguments, as shown in figure 5:

	00A77E54     00A77E55	56		pu	sh ecx sh esi					x87SW_S	F 0 x875W_	P 0	x87SW_U	0
EIP	→ 00A77E50				11 eax				>	Default (st	tdcall)	1 "\ "C •	▼ 5	Unlock
	2.CommandLir 256 conti.ex									2: [esp 3: [esp 4: [esp	+4] 00B9FE7 +8] 00A9207 +C] 72654B0	0 0 cont		
Dump 1	Dump 2	🜉 Dump 3	Dump 4	Dump 5	👹 Watch 1	[x=] Locals	2 Struct	00B9FDD4 ( 00B9FDD8 (			C:\\Users\\	\/D	esktop\\c	onti.exe∖""

Figure 5

The following strings have been decrypted using algorithms like the one presented in figure 1:







The executable creates a mutex called "YUIOGHJKCVVBNMFGHJKTYQUWIETASKDHGZBDGSKL237782321344" (if the malware runs with the "-nomutex" parameter, then no mutex is created):



Figure 7

GetNativeSystemInfo is used to retrieve information about the system:

00A786E6 51 00A786E7 FF D0	push ecx call eax	>	Default (stdcall)
eax= <kernel32.getnativesysteminfo> (76A75030) .text:00A786E7 conti.exe:\$186E7 #17AE7</kernel32.getnativesysteminfo>			2: [esp+4] 0000000 3: [esp+8] 00A92070 conti.00A92070 4: [esp+C] 00863000
	- <u>Manual III II (0)</u>	00B9FE7C 00B9	FED8 &L"WinSta0\\Default"



The malicious file creates 2 (which is the number of processors) threads that will handle the files encryption, as we'll describe in the upcoming paragraphs:



				Q1Q100000015
eax= <kernel32.createthree .text:00a7ca42<="" th=""><th></th><th>push 0 push 0 push conti.A7C7D0 push 0 call eax</th><th>&gt;</th><th>x87StatusWord 0000 x87Sw_B 0 x87Sw_C3 0 x87Sw_C2 0 x87Sw_C1 0 x87Sw_C0 0 x87Sw_E5 0 x87Sw_S5 0 x87Sw_P 0 x87Sw_U 0 v Default (stdcall) v 5 ↓ Unlod 1: [esp] 00000000 2: [esp+4] 0000000 3: [esp+4] 0000000 3: [esp+4] 00042044 conti.00A92044</th></kernel32.createthree>		push 0 push 0 push conti.A7C7D0 push 0 call eax	>	x87StatusWord 0000 x87Sw_B 0 x87Sw_C3 0 x87Sw_C2 0 x87Sw_C1 0 x87Sw_C0 0 x87Sw_E5 0 x87Sw_S5 0 x87Sw_P 0 x87Sw_U 0 v Default (stdcall) v 5 ↓ Unlod 1: [esp] 00000000 2: [esp+4] 0000000 3: [esp+4] 0000000 3: [esp+4] 00042044 conti.00A92044
💭 Dump 1 💭 Dump 2 🐙	Dump 3 💭 Dump 4	🛛 Dump 5 🛛 🛞 Watch 1 🛛 🕼 🖉 Struc	t 0089FE5C 000	
Address Hex 0089FED8 09 00 00 00 00 0089FEE8 01 00 00 00 01 0089FEE8 12 00 01 18 00	00 00 00 08 21 00 00 00	00 01 00		

Figure 9

The executable takes a snapshot of all processes in the system by calling the CreateToolhelp32Snapshot routine ( $0x2 = TH32CS\_SNAPPROCESS$ ):

EIP	00A7BEBB 00A7BEBD 00A7BEBF	6A 00 6A 02 FF D0		pu: ca	sh 0 sh 2 1 eax	_	_		>	Def	75W_SF 0 x875W_P ault (stdcall) [esp] 00000002	0 x875₩_U ▼ 5	0
	32.CreateTool EBF conti.exe			890)						2:3:4:			
🚛 Dump 1	Dump 2	Dump 3	Ump 4	Dump 5	💮 Watch 1	[x=] Locals	Struct	00B9FC18 00B9FC1C					

Figure 10

The processes are enumerated using the Process32FirstW and Process32NextW APIs:

Image: Struct structure       Image: Structure	0001D8 89FC48
eax= <kernel32.process32nextw> (76AB03D0) .text:00A7C0AA conti.exe:\$1C0AA #184AA</kernel32.process32nextw>	x875W_5F 0 x875W_P 0 x875W_U 0 Default (stdcall)

Figure 12

The malware searches for the "explorer.exe" process and saves its ID into a buffer for later use. The CoInitializeEx function is utilized to initialize the COM library for use by the thread, as highlighted below:

EIP	● 00A66BD0 00A66BD0 ● 00A66BE0	6A 00 FF D0		pu	sh 0 sh 0 11 eax	_	_		~ >	Defa	75W_SF 0 x875W_P ault (stdcall) [esp] 00000000	 0
	BEO conti.ex									2: 3: 4:	[esp+4] 00000000 [esp+8] 00000003	
Dump 1	Dump 2	Dump 3	Dump 4	Dump 5	🛞 Watch 1	[x=] Locals	Struct	00B9F4B0 00 00B9F4B4 00				





The CoInitializeSecurity API is used to register security and set the default security values for the current process ( $0x3 = RPC\_C\_IMP\_LEVEL\_IMPERSONATE$ ):

00/ 00/ 00/ 00/ 00/ 00/ 00/ 00/ 00/ 00/		00 00 00 00 00 00 00 00 00 00 00 00 00	pu pu pu pu pu pu ca	ish 0 ish 0 ish 3 ish 3 ish 0 ish 0 ish 0 ish FFFFFFF ish 0 ill eax			_			3: [esp+8] 4: [esp+C]	Empty) : Empty) : x87SW_C3 x87SW_C0 x87SW_P	0 x875W_U	mpty) mpty) 2 O 5 O
Dump 1	p 2 🛄 Dump 3	3 💭 Dump 4	Dump 5	🛞 Watch 1	[x=] Locals	Struct	(	0089F494 0089F498	FFFFFF	FF			
Address         Hex           02C41510         7.4         0.8         0.0           02C41520         AB         AB         AB           02C41530         E6         3F         36           02C41530         E6         3F         36           02C41540         01         00         00           02C41550         00         00         00           02C41550         00         00         00	AB EE FE EE F DC AD 95 00 1 00 00 00 00 0	E 00 00 00 00 B 5C AB B3 76 0 00 00 00 00	00 00 00 00 0D F0 AD BA 00 00 00 00	««««îþîþ æ?6Ü∖«*	v.ð.°		-^	0089F49C 0089F4A0 0089F4A4 0089F4A8 0089F4AC 0089F4B0 0089F4B4	000000 000000 000000 000000 000000	00 00 03 00 00			

Figure 14

The malware uses COM objects and wmic in order to delete the volume shadow copies on the system. It calls the CoCreateInstance function with the CLSID {4590F812-1D3A-11D0-891F-00AA004B2E24}, which creates an IWbemLocator object:

EIP	00A66FB8     00A66FB9     00A66FC9     00A66FC0     00A66FC2 <b>00A66FC 00A66FC €</b>	68 08 6A 01 6A 00 68 E8 FF D0	<u>91 A8 00</u>	pi pi pi pi	ush ecx ush conti.A8 ush 1 ush 0 ush conti.A8 all eax	91E8			>	x87SW_B x87SW_C1 x87SW_SF Default (std 1: [esp]	00A891E8 C	0		0
	se.CoCreateIn									2: [esp+ 3: [esp+ 4: [esp+		cont	i.00A89208	3
.text:00A66	SFC7 conti.ex	e:\$6FC/ #6:	SC/								01 00000000			
Dump 1	Dump 2	Dump 3	Ump 4	Dump 5	🥘 Watch 1	[x=] Locals	Struct	00B9F4A8	00000	000	.00A891E8			
Address He 00A891E8 11	EX L F8 90 45 3A	1D D0 11	89 1F 00 AA	00 4B 2E 24	ASCII	a.K.\$		^ 0089F4A0 0089F480 0089F484	00A89	208 conti	.00A89208			

Figure 15

A new IWbemContext object is created with the CLSID {674B6698-EE92-11D0-AD71-00C04FD8FDFF}:

00467385     00467385     00467385     00467385     00467385     6A 00     0046738C     6A 00     0046738C     68 F8 91 A8 00     0046738E     F8 00	push ecx push conti.A89218 push 1 push 0 push conti.A891F8 call eax	~ ~	x875tatusWord 0000         x875w_E0       x875w_E2       0         x875w_E1       x875w_C2       0         x875w_ST       x875w_ST       0         x875w_ST       x875w
<pre>eax=<combase.cocreateinstance> (76BC0480) .text:00A67393 conti.exe:\$7393 #6793</combase.cocreateinstance></pre>			2: [csp+4] 0000000 3: [csp+4] 0000001 4: [csp+6] 0000001 4: [csp+6] 00A89218 conti,00A89218
Image: Second		00B9F4A8 00000 00B9F4AC 00000	001 218 conti.00A89218

Figure 16

The ConnectServer method is utilized to connect to the "ROOT\CIMV2" namespace:



ODA67985 ODA67987 ODA67987 ODA67981 ODA67981 ODA67981 ODA67985 ODA67	push esi push dword ptr ss:[ebp-85C] push push o push o push o push o push o push o push ex push ex call dword ptr ds:[edx+C]	x87TW_2 3 (Empty) x87TW_3 3 (Empty) x87TW_4 3 (Empty) x87TW_5 3 (Empty) x87TW_6 3 (Empty) x87TW_7 3 (Empty) x87StatusWord 0000 x87SW_61 0 x87SW_C3 0 x87SW_C2 0 x87SW_51 0 x87SW_20 0 x87SW_50 0 x87SW_5F 0 x87SW_P 0 x87SW_U 0
dword ptr [edx+C]=[wbemprox.6F9A10D8]=wbemprox.6F9A2B           .text:00A679AD conti.exe:\$79AD #6DAD           ## Dump 1         ## Dump 2           Dump 1         ## Dump 2           Address   Hex         0045500 43 00143 00 40 00	5 🛞 Watch 1 [x=] Locals 🎾 Struct	1:         [esp] 02C4DFA8           2:         [esp+3] 02C50FLC L"ROOT\\CIMV2"           3:         [esp+8] 00000000           0089F491         02C4DFA8           0089F492         02C50FLC           0089F493         02C50FLC           0089F494         02C50FLC           0089F494         02C50FLC           0089F494         02C50FLC           0089F494         02C50FLC
0089F5D9         52         00         4F         00         54         00         5C         00         32         00         49         00         40         00         50         01         43         00         49         00         40         00         50         00         50         00         00         00         00         00         20         27         FF         FF         FF         FF         A0         70         00         00         00         20         27         CF         FF         FF         FF         A0         70         00         <	18	0089F4A4 00000000 0089F4AC 0000000 0089F4AC 0000000 0089F4B4 00250630 0089F484 0089F630

Figure 17

The binary sets the authentication information that is used to make calls on a proxy via a CoSetProxyBlanket API call  $(0xA = RPC_C_AUTHN_WINNT, 0x3 = RPC_C_AUTHN_LEVEL_CALL, 0x3 = RPC_C_IMP_LEVEL_IMPERSONATE):$ 

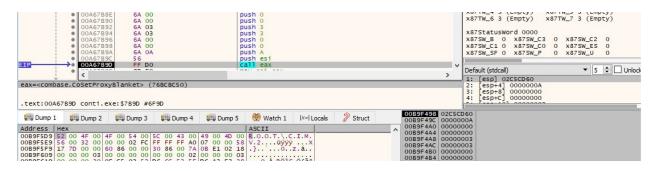


Figure 18

The following WQL (SQL for WMI) query is executed by the ransomware:

.text:00AG	7FF7 conti.exe:	-	Dump 4 💭 Dump !	5 🛞 Watch 1	[x=] Locals	Struct	00B9F4A0 020 00B9F4A4 020	5875-	
dword ptr	00A67FF3 00A67FF5 00A67FF6 00A57FF6 00A57FF7 <	8B 11 57 51 FF 52 50 rox.6BCD1500]=	fastprox.6BCFFF	mov edx, dword push edi push ecx call dword pt 30		-	>	× Di	875W_C1 0 x875W_C0 0 x875W_E5 0 875W_SF 0 x875W_P 0 x875W_U 0 efault (stdcall) ▼ 5 ↓ Unlock : [esp] 02C5CD60 : [esp+4] 02C5B754 L"WQL" : [esp+4] 02C5B754 L"WQL" : [esp+6] 02C5B14A L"SELECT * FROM Win32_Sh
	<ul> <li>00A67FE4</li> <li>00A67FE6</li> <li>00A67FE8</li> <li>00A67FE9</li> </ul>	6A 00 6A 30 50 C7 85 AC F	7 FF FF 00 00 00	push 0 push 30 push eax mov dword ptr	ss: ebp-85	41.0		×	87TW_6 3 (Empty) x87TW_7 3 (Empty) 87StatusWord 0000 87SW_8 0 x87SW_C3 0 x87SW_C2 0

Figure 19

For each volume shadow copy, the binary extracts its ID using the Get method:



ODA685DF         S0         push eax           D00A685DF         FF B5 A8 F7 FF FF         push dword ptr ss: [ebp-856]           D00A685E0         FF B5 A8 F7 FF FF         push dword ptr ds: [ebp-856]           dword ptr [ssi+10]         FF S6 10         Call dword ptr ds: [esi+10]           dword ptr [ssi+10]=[6BCD1BD0 <fastprox.&cwbemobject::get>]=<fastprox.cwbemobject::get></fastprox.cwbemobject::get></fastprox.&cwbemobject::get>	eax:L"         x875W_5F 0 x875W_P 0 x875W_U 0
.text:00A685E6 conti.exe:\$85E6 #79E6	3: [esp+6] 00000000           4: [esp+2] 0089F058           5: [esp+10] 00000000           0089EF00         030EE888           0089EF04         0089F05D           L"TD"

Figure 20

The following string that contains a process name with parameters is decrypted:

Address	He	(						111-1									ASCII
																	c.m.de.x.e
00B9EF39	2F	00	63	00	20	00	43	00	3A	00	5C	00	57	00	69	00	/.cC.:.\.W.i.
00B9EF49	6E	00	64	00	6F	00	77	00	73	00	5C	00	53	00	79	00	n.d.o.w.s.\.S.y.
																	s.t.e.m.3.2.\.w.
00B9EF69	62	00	65	00	6D	00	5C	00	57	00	4D	00	49	00	43	00	b.e.m.\.W.M.I.C.
00B9EF79	2E	00	65	00	78	00	65	00	20	00	73	00	68	00	61	00	e.x.es.h.a.
00B9EF89	64	00	6F	00	77	00	63	00	6F	00	70	00	79	00	20	00	d.o.w.c.o.p.y
00B9EF99	77	00	68	00	65	00	72	00	65	00	20	00	22	00	49	00	w.h.e.r.e".I.
00B9EFA9	44	00	3D	00	27	00	25	00	73	00	27	00	22	00	20	00	D.=.'.%.s.'."
00B9EFB9	64	00	65	00	6C	00	65	00	74	00	65	00	00	00	00	00	d.e.l.e.t.e

Figure 21

Wow64DisableWow64FsRedirection is utilized to disable file system redirection for the current thread:

• 00A68B10	51	push ecx			
EIP → 00A68B11 <	FF DO	call eax	eax:Wo	Default (stdcall) 1: [esp] 00B9F088	▼ 5 🖨 Unlock
<pre>eax=<kernel32.wow64disablew .text:00a68b11="" conti.exe:\$8<="" pre=""></kernel32.wow64disablew></pre>		(76A76A30)		2: [esp+4] 00000003 3: [esp+8] 00000005 4: [esp+C] 00000005 5: [esp+10] 030C0000	
all a sur a sur a		am 00 (5)	00B9EF14 00B9F	088	

Figure 22

The executable deletes each volume shadow copy that corresponds to the ID extracted above using the CreateProcessW API (0x08000000 = **CREATE\_NO\_WINDOW**):

	ODA655C3     ODA655C4     ODA655C4     ODA655C4     ODA655C4     ODA655C5     ODA655C5     ODA655D5     ODA655D5     ODA655D7     ODA655D7	Jush ecx, Jea ecx, dword ptr ss:[ebp-850] push 0 push 0 push 0 push 0 push 0 push 0 push 0 push 0 push 2 push ecx, dword ptr ss:[ebp-808] push ecx dword ptr ss:[ebp-808] push ecx dword ptr ss:[ebp-808] push ecx dword ptr ss:[ebp-808] push ecx dword ptr ss:[ebp-808]	ecx:L" ecx:L" ecx:L" ecx:C v	Default (stdcall)         ▼         5         ↓         Unlock           1:         [esp+3]         00000000         2:         [esp+4]         0089E708         L"cmd.exe         /c C:\\Windows\\Sy           3:         [esp+4]         00800000         4:         [esp+4]         00000000
Address         Hex           Od892708         63           O0892708         63           O0892718         2F           O0892728         66           O0892738         73           O089278         60           O089278         73           O0892787         77           O0892788         44           O0892788         40           O0892788         34           O0892788         36           O0892788         37	Bump 2         Bump 2         Bump 2           00         60         64         00         25           00         63         00         26         00         47           00         64         00         25         00         47           00         74         00         74         00         74           00         64         00         25         00         65         00         60         00         50         00         65         00         60         00         60         00         50         00         65         00         60         00         60         00         60         00         60         00         60         00         60         00         60         00         60         00         60         00         60         00         60         00         60         00         10         00         10         00         10         00         10         00         10         10         10         10         10         10         10         10         10         10         10         10         10         10         10         10         10	ASCII 0 E.m.de.x.e., 0 C.m.de.x.e., 0 C.m.de.x.e., 0 S.t.e.m.3.2., W. 0 S.t.e.m.3.2., W. 0 S.t.e.m.3.2., W. 0 J.e.m., W.M.I.C. 0e.x.e., S.h.a. 0 d.o.w.c.o.p.y., 0 w.h.e.r.e., ".I. 0 J.e., '.4.A.8.9. 0 Z.S.1.85.2.7. 1 4.6.4.8.0.2.6.8.8. 0 J.8.2.6.6.1.4.9. 0 J.e., '.4.4.8.9.	00821678         00000           00821670         0085           00825670         0085           00825684         00000           00825684         00000           00825680         08000           00825680         08000           00825680         08000           00825680         08000           00825680         08300           00825680         08360           00825680         08360           00825684         00000           0835684         000100           0835684         000100           0835684         000100           0835684         000100           0835684         000100           0835684         000100           0835684         000100           0835684         000100           0835686         000000           0835686         000000           0835686         000000           0835686         000000           0835686         000000           08356868         000000           08356868         000000           08356868         000000	708 L'cmd.exe /c C:\\Windows\\System32\\wbem\\WMI( 000 000 000 000 000 000 000 0

Figure 23

The malware restores file system redirection by calling the Wow64RevertWow64FsRedirection routine:



eax= <kernel32.wow64revertwow64fsredirection> (76A76A50)</kernel32.wow64revertwow64fsredirection>	2: [esp+4] 00000003 3: [esp+6] 00000005 4: [esp+C] 0000005 5: [esp+10] 030C0000 0089EFF4 0000000
COAGREGI 56 push esi     Coagregation of the si     Coagregation of the si	eax:wo > Default (stdcall)

Figure 24

The valid drives on the system are retrieved by calling the GetLogicalDriveStringsW function:

	00A74F1C     00A74F1D	56 57	push esi push edi	a second s		875W_U 0
EIP	00A74F1E	FF DO	call eax	eax:Ge v	Default (stdcall) 1: [esp] 00000009	▼ 5 😫 🗌 Unlock
	rel32.GetLogicalDri 74F1E conti.exe:\$1		080)		1: [esp+4] 030DB1A8 3: [esp+8] 00000003 4: [esp+C] 00000005 5: [esp+10] 0000005	
Ump 1	Ump 2	Dump 3 🔛 Dump 4	📖 Dump 5   🛞 Watch 1 🛛 🗱 Locals 🖉 Struct	00B9F858 000000 00B9F88C 030DB1		

Figure 25

There is a function call to WSAStartup that initiates the use of the Winsock DLL:

	<ul> <li>00A7B34A</li> <li>00A7B34B</li> </ul>	51 68 02 02 00	00	oush ecx oush 202					75W_SF 0 x875W_P 0 x8	
EIP	→• 00A7B350	FF D0		all eax			eax:WS V	Defa	ault (stdcall) [esp] 00000202	▼ 5 🗘 🗌 Unlock
	.WSAStartup> () 350 conti.exe:							2: 3: 4:	[esp+4] 00B9F740 [esp+8] 00000000 [esp+C] 030E1D90 L"D:\\" [esp+10] 00000003	
🚚 Dump 1	Dump 2	Dump 3 🛛 Dump	4 💷 Dump 5	🛞 Watch 1	[x=] Locals	Struct	10 000002 14 00B9F7			

Figure 26

A new socket is created by the process  $(0x2 = AF_INET, 0x1 = SOCK_STREAM, 0x6 = IPPROTO_TCP)$ :



Figure 27

The malicious process calls the WSAIoctl function with the **SIO\_GET\_EXTENSION\_FUNCTION\_POINTER** command code in order to invoke an extension function, as shown in figure 28:





				OL OIDSOOD LL
eax= <ws2< td=""><td>OOA79925     OOA79925     OOA79927     OOA79929     OOA79920     OOA79924     OOA79924     OOA79932     OOA79932     OOA79932     OOA79937     OOA79937     OOA79937     OOA79937     OOA79932     OOA79932</td><td>6A 00 6A 00 51 6A 04 6B F4 IF A9 00 6A 10 6B F4 IF A9 00 6A 10 58 06 00 00 C8 53 FF D0 F1F0)</td><td>push 0 push 0 push ecx push ecx push conti.A91FF4 push 10 legtecx.dword ptr ss:[ebp-14] push Cabooos push ebx call eax</td><td>eax:WS         *         S (Empty)         x87TW_3 3 (Empty)           x87TW_4 3 (Empty)         x87TW_5 3 (Empty)         x87TW_5 3 (Empty)           x87TW_6 3 (Empty)         x87TW_7 3 (Empty)           x87TW_5 0 (Empty)         x87TW_7 3 (Empty)           x87TW_5 0 (Empty)         x87TW_2 0 (Empty)           x87TW_6 3 (Empty)         x87TW_2 0 (Empty)           x87SxL0 x87TW_0 0 000         x87SW_2 0 (Empty)           x87SW_5 0 (Empty)         x87SW_2 0 (Empty)</td></ws2<>	OOA79925     OOA79925     OOA79927     OOA79929     OOA79920     OOA79924     OOA79924     OOA79932     OOA79932     OOA79932     OOA79937     OOA79937     OOA79937     OOA79937     OOA79932     OOA79932	6A 00 6A 00 51 6A 04 6B F4 IF A9 00 6A 10 6B F4 IF A9 00 6A 10 58 06 00 00 C8 53 FF D0 F1F0)	push 0 push 0 push ecx push ecx push conti.A91FF4 push 10 legtecx.dword ptr ss:[ebp-14] push Cabooos push ebx call eax	eax:WS         *         S (Empty)         x87TW_3 3 (Empty)           x87TW_4 3 (Empty)         x87TW_5 3 (Empty)         x87TW_5 3 (Empty)           x87TW_6 3 (Empty)         x87TW_7 3 (Empty)           x87TW_5 0 (Empty)         x87TW_7 3 (Empty)           x87TW_5 0 (Empty)         x87TW_2 0 (Empty)           x87TW_6 3 (Empty)         x87TW_2 0 (Empty)           x87SxL0 x87TW_0 0 000         x87SW_2 0 (Empty)           x87SW_5 0 (Empty)         x87SW_2 0 (Empty)
.text:00	A7993D conti.exe:\$1	993D #18D3D		3: [esp+8] 0089F5FC 4: [esp+C] 00000010 5: [esp+10] 00A91FF4 conti.00A91FF4
00895600	Hex B9 07 A2 25 F3 DD 95 44 40 30 D8 F8	Dump 3         Dump 4         Dump 4           60         46         8E         E9         76         58         74           89         01         72         88         74         00	ASCII 5 3E 1.6%óÝ F.évà.t.>	O055525C         0000210           008575C0         28000006           O08575C1         000011           008575C1         0000010           O08575C1         0000010           008575C1         0000010           O08575C1         0000010           0085750         0000010           O085750         0000000           0085750         0000000           O085750         0000000

Figure 28

The gethostname routine is utilized to retrieve the host name for the local computer:

	00A795D     00A795E	2 51	01 00 00		push 100 push ecx						SW_SF 0 x87SW_P 0	x875W_U 0
EIP	→• 00A795E	E FF DO			call eax				eax:ge v		ult (stdcall) [esp] 00B9F50C	🔻 💈 🛨 Unloc
	2.gethostname 95E3 conti.ex									2: 3: 4:	[esp+4] 00000100 [esp+8] 00000000 [esp+8] 0089F99C [esp+C] 0089F99C [esp+10] 00056A50	
Dump 1	Dump 2	Ump 3	Dump 4	Dump 5	🛞 Watch 1	[x=] Locals	2 Struct	00B9F50 00B9F50	4 00B9F5 8 000001			

Figure 29

The malicious file retrieves host information that corresponds to the host extracted above:

EIP 00A79604 00A79605	51 FF D0	push ecx call eax	ecx: "D eax:ge >	Default (stdcall)
<pre>eax=<ws2_32.gethostbyname> .text:00A79605 conti.exe:\$1</ws2_32.gethostbyname></pre>				2: [esp+4] 3: [esp+8] 4: [esp+6] 5: [esp+10]
		Dunie F Market I Ivaliante () church	00B9F508 00B9F5	OC   "DESKTOP-



The CreateIoCompletionPort API is used to create an I/O (input/output) completion port that is not yet associated with a file handle (0xFFFFFFF = **INVALID\_HANDLE\_VALUE**):



Figure 31

The ARP table is extracted by calling the GetIpNetTable routine, and the result is stored in a MIB\_IPNETTABLE structure:



Address   He	av		ASCTT		00B9F4D4 00000000	
Dump 1	Dump 2	Dump 3 📖 Dump 4	🕮 Dump 5 👹 Watch 1 🛛 🕬	als 🖉 Struct	00B9F4CC 030EA938 00B9F4D0 00B9F5D4	
	0F07 conti.exe:\$				2: [esp+4] 0089F5D4 3: [esp+6] 0000000 4: [esp+C] 0000000 5: [esp+10] 030E1D90 L"D:\\"	
	<pre>api.GetIpNetTable</pre>	(71025250)			1: [esp] 030EA938	
IP	->• 00A79F07	FF DO	call eax		eax:Ge ✓ Default (stdcall)	
	<ul> <li>00A79F05</li> <li>00A79F06</li> </ul>	51 53	push ecx push ebx		x875W_SF 0 x875W_P 0 x875W_U 0	
	• 00A79F03	6A 00	push 0		x87SW_C1 0 x87SW_C0 0 x87SW_ES 0	
					0100000	111 Ø 1 100 n.o.
					Renter II 50 jr 60 no 10 00000 11 51 0	

Figure 32

Each IP address extracted above is converted into a string (dotted-decimal format):

• 00A7A21D	57	push edi		
	FF D0	call eax	eax:in v	Default (stdcall) 🔻 5 🖨 🗌 Unlock
eax= <ws2_32.inet_ntoa> (76F</ws2_32.inet_ntoa>				1: [esp] 020AA8C0 2: [esp+4] 00000000 3: [esp+8] 030E1D90 L"D:\\" 4: [esp+C] 0000000
.text:00A7A21E conti.exe:\$1	LA21E #1961E			5: [esp+10] 00A91FE8 conti.00A91FE8
		Duran E Markah 1 (well early 🧐 Church	00B9F4D4 020AA	800

Figure 33

The malware is only interested in local IP addresses because it compares every IP address with the prefixes "172.", "192.168.", "10." and "169.". The binary creates 2 new threads via a function call to CreateThread:

ODA78870 GA 00     ODA7877 GA 00     ODA7874 GA 00     ODA7876 GA <u>00</u> ODA7876 GA <u>00</u> ODA7878 GA <u>00</u> ODA7878 GA <u>00</u> ODA7878 GA <u>00</u> ODA7870 GA <u>00</u> ODA787	push 0 push 0 push 0 push conti.A7A880 push 0 push 0 call eax	eax:Cr v	
.text:00A7B87F conti.exe:\$1B87F #1AC7F		3: 4: 5:	[esp+8] 00A7A880 conti.00A7A880 [esp+C] 00000000
Image: Dump 1         Image: Dump 2         Image: Dump 3         Image: Dump 4         Image: Dump 5           Address         Hex         0089F58D         31.63.92         00.74.64.00         31.37.32         22.00         00.00	Watch 1  x=  Locals Z Struct 00B9F6 00B9F6	00 0000000 04 0000000 08 00A7A880 0C 0000000 10 0000000 14 0000000	cont1.00A7A880
	Figure 34		
ODA7BA34     GA 00     ODA7BA35     GA 00     ODA7BA35     GA 00     ODA7BA36     ODA7BA38     GA 00     ODA7BA3A     GA 00     ODA7BA3A     GA 00     ODA7BA3F     GA 00     ODA7BA41     GA 00     ODA7BA41     GA 00     ODA7BA43     GA     ODA7BA4     ODA7B	push 0 push 0 push 0 push conti.A7AF90 push 0 push 0 push 0 call eax	eax:Cr v	875tatusword 0000 875W_8 0 x875W_C3 0 x875W_C2 0 875W_C1 0 x875W_C5 0 875W_57 0 x875W_C 0 x875W_E5 0 875W_57 0 x875W_P 0 x875W_U 0 fault (stdcall)
eax= <kerne132.createthread> (76A745B0) .text:00A7BA43 conti.exe:\$1BA43 #1AE43</kerne132.createthread>		1:22	[esp+8] 00A7AF90 conti.00A7AF90 [esp+C] 00000000
Image: Comp 1         Image: Comp 2         Image: Comp 3         Image: Comp 4         Image: Comp 5           Address         Hex         Obs95500         33         36         39         9C         F9         B9         00         20         FC         E8         76         51         DA         85         22         F           00895520         FF         FF         C2         C6         E5         07         45         A2         76         FF         FF         FC         C1         E5         07         45         A2         76         FF         FF         FC         C1         E5         76         91         45         A2         76         FF         FF         FC         C1         E5         76         91         45         A2         76         FF         FF         FC         C1         56         77         45         A2         76         FF         FF         FC         FF         FC         76         76         76         76         76         76         76         76         76         76         76         76         76         76         76         76         76         76         76	Watch 1  x=  Locals Struct 00B9F6 ASCII	500         00000000           504         00000000           508         00A7AF90           50C         00000000           510         00000000	conti.00A7AF90

Figure 35

PostQueuedCompletionStatus is utilized to send an I/O completion packet to the completion port created earlier (**dwCompletionKey** = 0x1):



EIF	push 0 push 1 push 0 push est call eax		
<pre>eax=<kernel32.postqueuedcompletionstatus> (76A75C .text:00A7BC48 conti.exe:\$1BC48 #1B048</kernel32.postqueuedcompletionstatus></pre>	20)	2: [esp+4] 00000200 3: [esp+8] 0000001 4: [esp+2] 0000000 5: [esp+1] 0000000	
Homp 1         Homp 2         Homp 3         Homp 4         Homp 4           Address         Hex         Hex         Address         Hex         Hex	Dump 5 🔮 Watch 1 [x=] Locals 🤌 Struct	001916003         0000032C           00897610         00000000           00897611         00000000           00897614         00000000	

Figure 36

### **THREAD ACTIVITY – SUB\_A7AF90 FUNCTION**

The file creates a queue for timers (which are objects that allow the user to specify a function that will be called at a particular time):

COATAFCD FF DO	call eax	eax:CT ∨ Default (stdcall) ↓ 5 €) 1: [esp] 0047AF90 conti.0047AF90	Unlock
eax= <kernel32.createtimerqueue> (76A745E0) .text:00A7AFCD conti.exe:\$1AFCD #1A3CD</kernel32.createtimerqueue>		12 [ESD] UDAAAS E CONTI UDAAS 90 21 [ESD+4] 0000000 CONTI .00A7590 22 [ESD+4] 0000000 42 [ESD+C] 00000000 53 [ESD+10 0000000	



The ransomware attempts to extract the I/O completion packet from the I/O completion port (sent by the main thread) by calling the GetQueuedCompletionStatus routine:





A new socket is created by calling the WSASocketW API ( $0x2 = AF_INET$ ,  $0x1 = SOCK_STREAM$ ,  $0x6 = IPPROTO_TCP$ ,  $0x1 = WSA_FLAG_OVERLAPPED$ ):

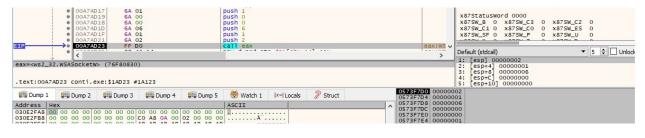


Figure 39

The bind routine associates the local address with the above socket:



Figure 40

CreateIoCompletionPort is utilized to associate the socket created above with the I/O completion port. After this operation is complete, the process can receive notifications of the completion of I/O operations involving the socket handle (**CompletionKey** =  $0x^2$ ):

EIP	00A7AD 00A7AD 00A7AD 00A7AD 00A7AD	DA 6A 0 DC 56 DD 57 DE FF D	0		push 0 push 2 push esi push edi call eax			ę	eax:Cr v	Default (stdcall)	
eax= <kernel:< th=""><th></th><th></th><th></th><th>30)</th><th></th><th></th><th></th><th></th><th></th><th>1: [esp] 00000338 2: [esp+4] 0000032C 3: [esp+8] 0000002 4: [esp+C] 00000000 5: [esp+10] 030E0508</th><th></th></kernel:<>				30)						1: [esp] 00000338 2: [esp+4] 0000032C 3: [esp+8] 0000002 4: [esp+C] 00000000 5: [esp+10] 030E0508	
Dump 1	Dump 2	🚛 Dump 3	💷 Dump 4	💷 Dump 5	🛞 Watch 1	[x=] Locals	Struct	0573F7D	8 000003 C 000003		
Address He					ASCII				0 0000000		

Figure 41

The binary converts a port number (445) from network byte order to host byte order:

• 00A7AEE1	68 BD 01 00 00	push 1BD			N B 0
	FF DO	call eax	eax:nt v	Default (stdcall) 1: [esp] 000001BD	▼ 5 💠 🗌 Unlock
<pre>eax=<ws2_32.ntohs> (76F8820 .text:00A7AEE6 conti.exe:\$1</ws2_32.ntohs></pre>				1: [esp+4] 030E0508 3: [esp+8] 030E0508 4: [esp+C] FF000000 5: [esp+10] 0000002	
all all		And	0573F7F4 000003	LBD	

Figure 42

The malware tries to connect to different IP addresses on port 445 (192.168.10.x and 192.168.164.x) using the LPFN\_CONNECTEX function, as described below:

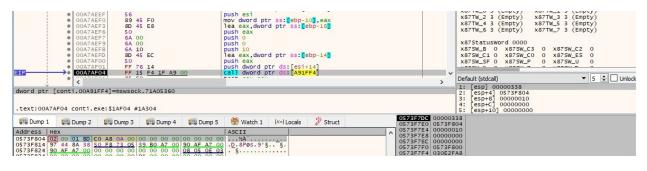


Figure 43

The CreateTimerQueueTimer routine is used to create a timer-queue timer, which expires at a specific time (0x7530 = 30000ms = 30 seconds) and then a callback function is called:



O0A7B23D 6A 00 00A7B23F 6A 00 00A7B28F 6A 00 00A7B28F 6A 00 00A7B291 6B 30 75 00 00 00A7B295 65 60 AF A7 00 00A7B295 51 00A7B295 51 00	push 0 push 7530 push 0 push 0 push conti.A7AF60 push ecx conti.A7AF60 push ecx conti.A7AF60 push ecx continue ax	eax:Cr vefault (sto	s (empLy) xb/1w_/ s (empL) usword 0000 0 x875w_C3 0 x875w_C2 1 0 x875w_C3 0 x875w_C2 1 0 x875w_C9 0 x875w_U 5 0 x875w_P 0 x875w_U idal)	0
eax= <kernel32.createtimerqueuetimer> (76A745F0) .text:00A7B29F conti.exe:\$1B29F #1A69F</kernel32.createtimerqueuetimer>		2: [esp- 3: [esp- 4: [esp-	+4] 030E0508 +8] 00A7AF60 conti.00A7AF60 +C] 00000000 +10] 00007530	
Image: Constraint of the state of	ASCII	0573F808 0573F838 0573F808 030E0508 0573F80C 00A74F60 conti 0573F810 0000000 0573F814 00007530 0573F818 0000000 0573F816 0000000		

Figure 44

The setsockopt API is utilized to set the **SO\_UPDATE\_CONNECT\_CONTEXT** option, which updates the properties of the socket after a connection is established ( $0xFFFF = SOL_SOCKET$ ,  $0x7010 = SO_UPDATE_CONNECT_CONTEXT$ ):

<ul> <li>00A78081</li> <li>6A 00</li> <li>00A78083</li> <li>68 10 70 00 00</li> <li>00A78088</li> <li>68 FF FF 00 00</li> <li>00A78080</li> <li>56</li> </ul>	push 0 push 0 push 7010 push FFFF push esi call eax	eax:se v	x8/svLatusword 0000 x875W_E0 x875W_C3 0 x875W_C2 0 x875W_C1 0 x875W_C0 0 x875W_E5 0 x875W_5F 0 x875W_F 0 x875W_U 0 Default (stdcall) ▼ 5 0 Unlock
<pre>eax=<ws2_32.setsockopt> (76F7F880) .text:00A7B08E cont1.exe:\$1B08E #1A48E</ws2_32.setsockopt></pre>			1: [csp+4] 0000FFFF 3: [csp+4] 0000FFFF 4: [csp+6] 00000000 5: [csp+10] 00000000
Image: Second system         Image: Se	Watch 1         Ix= Locals         Struct         0573F8           ASCII         0573F8         0573F8	0C 000003 10 0000FF 14 000070 18 000000 1C 000000	338 FF D10

Figure 45

The file retrieves the **SO\_CONNECT\_TIME** option, which represents the number of seconds a socket was connected (0xFFFF = **SOL\_SOCKET**, 0x700C = **SO\_CONNECT\_TIME**):

OQA7B0B2 51     OQA7B0B2 51     OQA7B0B2 51     OQA7B0B2 51     OQA7B0B2 51     OQA7B0B2 51     OQA7B0B2 68 0C 70 00 00     OQA7B0B2 68 0C 70 00 00     OQA7B0B2 68 0C 7F F0 00     OQA7B0B2 FF 00 00     OQA7B0B2 FF 00 00     OQA7B0B2 FF	push eck lea eck dword ptr ss:[esp+28] push eck push rooc push FFFF push rest all eak call ea call eak call ea call eak call eak	x875tatusWord 0000 x875W_B 0 x875W_C3 0 x875W_C2 0 x875W_C1 0 x875W_C0 0 x875W_E5 0 x875W_5 0 x875W_D 0 x875W_U 0 Default (stdcall) ▼ 5 ↓ Unlock
eax= <ws2_32.getsockopt> (76F86E70)</ws2_32.getsockopt>		2: [esp+4] 0000FFFF
		3: [esp+8] 0000700C
.text:00A7B0C3 conti.exe:\$1B0C3 #1A4C3		4: [esp+C] 0573F844 5: [esp+10] 0573F840
Ump 1 Ump 2 Ump 3 Ump 4 Ump 5		FF
Address Hex	ASCII 0573F814 000070 0573F818 0573F8	
0573F804 00 00 00 FC 00 00 38 03 00 00 FF FF 00 0	0 ü 8 ÿÿ 0573F81C 0573F81C 0573F8	

Figure 46

Whether the sample has successfully established a connection to a particular IP address, then it calls the WSAAddressToStringW routine to convert the components of that sockaddr structure into a human-readable string:

	push ecx lea ecx_dword ptr ds:[esi+4] push ecx push 0 push 10 lea ecc dword ptr ss:[ebp-10] pearect dword ptr ss:[ebp-10]	eax:ws > Defaul 2: [	statusword 0000 W_B 0 x875W_C3 0 W_C1 0 x875W_C0 0 W_SF 0 x875W_P 0 t (stdcall) esp1 0573F800 esp+4] 00000010	w_/ S (Empry) X875W_C2 0 X875W_ES 0 X875W_U 0 X875W_U 0 ▼ 5 € Uniod
.text:00A7AC06 conti.exe:\$1AC06 #1A006		4: 6	esp+8] 00000000 esp+C] 030F0F24 esp+10] 0573F810	
💷 Dump 1 🚛 Dump 2 🚛 Dump 3 🚛 Dump 4 🚛 Dump 5	🐨 Watch 1 🛛 🗱 Locals 🖉 Struct	573F7E4 0573F800 573F7E8 00000010		
Address Hex 0573F800 02 00 00 00 C0 A8 0A 00 C5 B0 A7 00 38 03 00 0 0575F800 02 00 00 00 C0 A8 0A 00 C5 B0 A7 00 38 03 00 0	ASCII 00	573F7EC 00000000 573F7F0 030F0F24 573F7F4 0573F810		







PostQueuedCompletionStatus is utilized to send an I/O completion packet to the completion port created before (**dwCompletionKey** = 0x3):

	push 0 push 3 push 0 push esi call eax	eax: Fo V	x875W_B 0 x875W_C2 0 x875W_C2 0 x875W_C1 0 x875W_C5 0 x875W_E5 0 x875W_5F 0 x875W_P 0 x875W_U 0 Default (stdcall) ▼ 5 \$ I ulado
<pre>eax=<kernel32.postqueuedcompletionstatu: .text:00A7AF82 conti.exe:\$1AF82 #1A382</kernel32.postqueuedcompletionstatu: </pre>	s> (76A75C20)	>	1: [esp] 0000032C 2: [esp+4] 0000000 3: [esp+5] 00000003 4: [esp+C] 00000003 5: [esp+1] 00A7AF60 conti.00A7AF60
Ump 1 Ump 2 Ump 3 Ump 3	ump 4 💭 Dump 5 🛞 Watch 1 🛛 [x=] Locals	0483F950         000003           0483F964         000000           0483F964         000000           0483F968         000000	000



The binary shuts down send operations for the socket  $(0x1 = SD\_SEND)$ :

00A7B207     00A7B209	6A 01 56	push 1 push esi	x875W_SF 0 x875W_P 0 x875W_U 0
	FF DO	call eax	eax:sh ∨ > Default (stdcall)
<pre>eax=<ws2_32.shutdown> (76F8 .text:00A7B20A conti.exe:\$1</ws2_32.shutdown></pre>			2: [ssp41] 00000001 3: [ssp43] 0047AF90 cont1.00A7AF90 4: [ssp42] 0047AF90 cont1.00A7AF90 5: [ssp+10] 0000000
💭 Dump 1 💭 Dump 2 💭 [	Dump 3 💭 Dump 4 💭	Dump 5   Watch 1 🛛 🕸 Struct	057212313 00000338  0573F81C  00000001

Figure 49

### **THREAD ACTIVITY – SUB\_A7A880 FUNCTION**

The NetShareEnum function is utilized to retrieve information about the network shares available on other computers:

<ul> <li>00A7A518</li> <li>80 40 F0</li> <li>00A7A518</li> <li>00A7A516</li> <li>80 40 F8</li> <li>00A7A517</li> <li>81 FF</li> <li>00A7A528</li> <li>61 FF</li> <li>00A7A525</li> <li>60 40 EC</li> <li>00A7A526</li> <li>6A 01</li> <li>00A7A528</li> <li>63 S3</li> </ul>	push eck lea ecx,dword ptr ss:[ebp-10] push ecx lea ecx,dword ptr ss:[ebp-6] push ecx push ffFFFFFF lea ecx,dword ptr ss:[ebp-14] push ecx push ecx ell eax	ebx:L" eax:Ne	AST M_2 3 (LIMPL3) AST M_3 3 (LIMPL3) AST M_2 3 (LIMPL3) AST M_3 3 (LIMPL5) AST M_4 3 (LIMPL5) AST M_3 3 (LIMPL5) AST M_4 3 (LIMPL5) AST M_3 3 (LIMPL5) AST M_4 3 (LIMPL5) AST M_4 3 (LIMPL5) AST M_4 3 (LIMPL5
eax= <srvcli.netshareenum> (73844310) .text:00A7A529 conti.exe:\$1A529 #19929</srvcli.netshareenum>		23	: [esp+4] 00000001 : [esp+8] 055FFCF4 : [esp+C] FFFFFFFF : [esp+10] 055FFD00
💭 Dump 1 💭 Dump 2 💭 Dump 3 💭 Dump 4 💭 Dump 5	🤓 Watch 1 🛛 🖉 Locals 🖉 Struct 055FF	C74 030F0F24 C78 00000001	
Address         Hex           055FFCF4         00	Ajšlý, fešt. Š.	C7C 055FFCF4 C80 FFFFFFF C84 055FFD00 C88 055FFCF8 C8C 055FFCF6	- ) 3

Figure 50

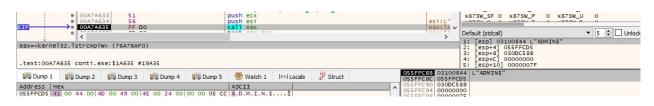
Some strings that will be written in the log file (if logging mode is enabled) are also decrypted using custom algorithms:

Address	He	ĸ															ASCII
055FFCD5	41	00	44	00	4D	00	49	00	4E	00	24	00	00	00	05	CC	A.D.M.I.N.\$Ì
Address	He	ĸ															ASCII
055FFCAD	46	00	6F	00	75	00	6E	00	64	00	20	00	73	00	68	00	F.o.u.n.ds.h.
055FFCBD	61	00	72	00	65	00	20	00	25	00	73	00	2E	00	00	00	a.r.e%.s
Address	He	ĸ															ASCII
055FFD21	53	00	74	00	61	00	72	00	74	00	69	00	6E	00	67	00	S.t.a.r.t.i.n.g.
055FFD31	20	00	73	00	65	00	61	00	72	00	63	00	68	00	20	00	.s.e.a.r.c.h
055FFD41	6F	00	6E	00	20	00	73	00	68	00	61	00	72	00	65	00	o.ns.h.a.r.e.
055FFD51	20	00	25	00	73	00	2E	00	00	00	00	00	54	11	03	40	.%.ST.@





The "ADMIN\$" share will not be targeted by the malware (the others will be encrypted):





## **THREAD ACTIVITY – SUB\_A7C7D0 FUNCTION**

CryptAcquireContextA is used to obtain a handle to a key container within a CSP (0x18 = **PROV\_RSA\_AES**, 0xF0000000 = **CRYPT\_VERIFYCONTEXT**):

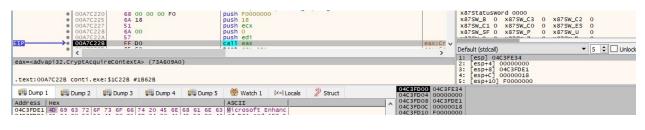


Figure 53

### An RSA public key is imported via a CryptImportKey function call:

<pre>00A7C821 00A7C822 00A7C824 00A7C826 00A7C826 00A7C826 00A7C828 00A7C88 00A7C88 00A7C88 00A7C88 00A7C88 00A7C8</pre>		push ecx push o push looo push conti,A90488 push conti,A90488 call eax	x87StatusWord 0000 x87Sw_B 0 x87Sw_C 0 0 x87Sw_C 2 0 x87Sw_C 10 x87Sw_C 0 0 x87Sw_U 0 x87Sw_S 5 0 x87Sw_P 0 x87Sw_U 0 x87Sw_S 5 0 x87Sw_P 0 x87Sw_U 0 → 1 (sep) 009EFE8 x8CPAcquireContext> 2: (sep) 009EFE8 x8CPAcquireContext> 2: (sep) 009EFE8 x8CPAcquireContext> 2: (sep+3) 0000000 3: (sep+10) 0000000
Image         Image         Image         Image         Image           Address         Hex         Image         Ima	00 00 52 53 41 31 00 10 D5 80 A0 0F 88 A0 86 05 CF 3A 80 0F 85 00 86 05 CF 3A 80 0F 45 00 86 05 CF 3A 80 0F 45 05 CF 38 00 50 24 27 25 05 88 45 EB 90 50 24 27 25 05 88 45 EB 90 50 24 27 25 05 88 45 EB 90 50 7 00 A8 F0 32 O3 55 FC 48 08 73 02 89 A8 48 33 C3 168 F6 56 C7 80 38 F7 C0 1D 00 70 05 A6 07 00 FD FD 59 02 33 28 7C 75 4D C4 6C 91 30 28 7C 75 4D C4 6C 91 30 28 7C 75 4D C4 80 89 68 45 51 40 C4 89 68 45 51 40 C4 89 68 45 51 40 C4 89 68 15 51 60 C5 A0 77 89 F8 15 70 68 66 77 00 F0 59 78 97 51 60 00 C5 A0 77 59 61 35 50 60 C5 A0 77 59 61 35 50 75 95 78 80 75 95 78 80 78 50 50 75 95 78 80 75 95 78 78 80 75 95 78 78 78 78 78 78 78 78 78 78 78 78 78	FE E0 [ q[0 6%]à 91 F4 q[0 6%]à 93 F4 q[0 6%] F4 69 éf: swifd0x1, E <sup>-1</sup> 48 67 I zê, ] <sup>n</sup> àÊ; Hg FC 08 0, §; '%M, ô-U, FB 13 m.d.8, UUK, s. 'U, FB 13 m.d.8, UUK, s. 'U, FB 02 8, B, B, H, M, C,	Ocception         030EFC88           04C3FEC0         00001000           04C3FEC0         00001000           04C3FEC1         00001000           04C3FEL1         00000000           04C3FEL3         00000000           04C3FEL4         0042041           04C3FEL4         0042044           04C3FEL4         0042044           04C3FEL4         0042044           04C3FEL4         0042044           04C3FEL4         0042044           04C3FEL4         0042047           04C3FEL4         0042047           04C3FEL4         0042047           04C3FEL4         0042044



The process creates a file called "readme.txt" in every folder that it encrypts (0x40000000 = **GENERIC\_WRITE**, 0x2 = **CREATE\_ALWAYS**):



04C3F6FC 00000000

Figure 55

The following 4-byte values suggest that the encryption algorithm is a ChaCha cipher (Ref. <u>https://arxiv.org/pdf/1907.11941.pdf</u>):

.text:00A766EA p	ush es	i	
.text:00A766EB p	ush ed	i	
.text:00A766EC m	iov [e	bp+var_80],	61707865h
.text:00A766F3 m	iov [e	<pre>bp+var_7C],</pre>	3320646Eh
.text:00A766FA m	iov [e	bp+var_78],	79622D32h
.text:00A76701 m	iov [e	bp+var_74],	6B206574h
.text:00A76708 m		<pre>bp+var_44],</pre>	eax
.text:00A7670B c	all <mark>su</mark>	b_A65AF0	

Figure 56

The encrypted content of the ransom note is decrypted using the ChaCha algorithm, and the file is populated by calling the WriteFile routine, as highlighted in figure 57.

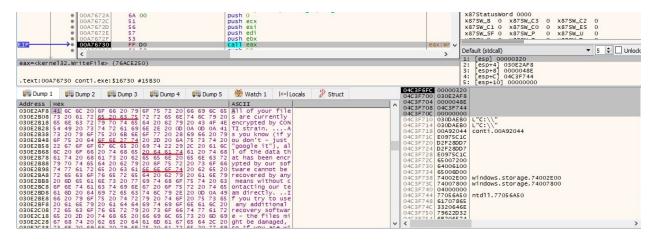


Figure 57

The files are enumerated in the targeted directory using the FindFirstFileW and FindNextFileW APIs:



€ 00A76937 51 00A76938 56 00A76938 FF D0 c c c c c c c c c c c c c c c c c c c	push ecx push esi call eax	x875M_SF 0         x875M_P         0         x875M_U         0           eaxiFiv
text:00A76939 conti.exe:\$16939 #15D39 # Dump 1 🙀 Dump 2 👯 Dump 3 🗱 Dump 4 👯 D	Dump 5 🔞 Watch 1 🛛 Ix=l Locals 🎾 Struct	3:         [esp+8]         00A92050         conti.00A92050           4:         [esp+6]         030E2000         U"C:\"           5:         [esp+10]         00A92044         conti.00A92044           04C3F7E4         04C3FF840         [U"C:\\="
	Figure 58	
00A772CC 51 00A772CD 53 00A772CE FF D0	push ecx push ebx call eax	eax;Fi         x875W_SF 0 x875W_P 0 x875W_U 0           Default (stdcall)         5 0
<pre>vext:004772CE conti.exe:\$172CE #166CE</pre>		1: [esp1] 030EFED8 2: [esp+4] 04C3FED8 3: [esp+4] 04A92050 conti.00A92050 4: [esp+6] 00A92050 conti.00A92050 4: [esp+10] 00A92044 conti.00A92044
Dump 1 Dump 2 Dump 3 Dump 4 Dump 4	Dump 5   Watch 1 🛛 🕸 Struct	04C3F7E0 030EFED8 04C3F7E4 04C3F8A0

Figure 59

There is a comparison between the directory name and a list of directories that will be skipped by the ransomware:

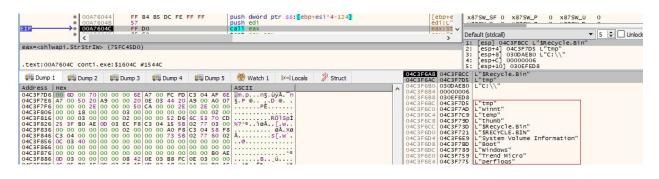


Figure 60

The PathIsDirectoryW routine is utilized to determine whether a path is a valid directory:

O0A7707C O0A7707D	50 FF 15 04 91 A8 00	push eax call dword ptr ds:[<&PathIsDirectoryw>]	eax:L"	Default (stdcall)
dword ptr [00A89104 <conti. .text:00A7707D conti.exe:\$1</conti. 		vapi.PathIsDirectoryW>		1: [csp14] 00A9205 Conti.00A92050 3: [csp+4] 00A92050 conti.00A92050 3: [csp+8] 030E2000 L"C:\\" 4: [csp+C] 00A92044 conti.00A92044 5: [csp+10] 000007A0
		- Maria 1. 1. 1. 6	04C3F7E4 030C54	480  L"C:\\\$Recycle.Bin"

Figure 61

The following files/files extensions will also be skipped by Conti:

00A764 00A764 00A765 <	4A 57	push conti.A904A0 push edi call eax		e	di:L" ax:St v D	efault (stdcall)	P 0 x875₩_U 0
eax= <sh]wapi.strstriw> .text:00A7644B conti.e</sh]wapi.strstriw>					234	: [esp+4] 00A904A0 : [esp+8] 030DAEB0	) L"C:\\" E L"WINRE_BACKUP_PARTITION.M
💭 Dump 1 🛛 💭 Dump 2	💷 Dump 3 🛛 💭 Dump 4 💈	🗒 Dump 5 🛛 👹 Watch 1 🛛 💷 L	ocals 🖉 Struct	04C3F730	00A904A0	L"\$WINRE_BACKUP_	PARTITION.MARKER"
04C3F7E6 A7 00 50 20 A 04C3F7F6 00 00 00 2E 0 04C3F806 00 00 18 00 0 04C3F806 25 3F 0AE 0 04C3F836 25 3F 0AE 0 04C3F836 C3 04 00 00 0 04C3F846 C3 04 00 00 0	(9)         00         00         20         0E         03         44         20         A9           (0)         00         00         50         CA         00         00         2E         00           (0)         00	00 02 00 53 70 CDRÖlspi 77 03 00 %?°°.iøÅ[.w. 04 58 F8		04C3F738 04C3F73C 04C3F74C 04C3F744 04C3F744 04C3F744 04C3F74C 04C3F75C 04C3F75C 04C3F75C	030EFED8 04C3F7D5 04C3F7C9 04C3F7BD 04C3F7BD 04C3F7B1 04C3F7A5 04C3F781	L"WINRE_BACKUP_P, L".exe" L".dll" L".lnk" L".sys" L".msi" L"ceadme.txt" L"CONTI_LOG.txt"	ARTITION.MARKER"

Figure 62



The sample descrypts the following DLL names: OleAut32.dll, Rstrtmgr.dll, Iphlpapi.dll, Netapi32.dll, Advapi32.dll, Kernel32.dll, Shell32.dll, Shlwapi.dll, ws2\_32.dll, User32.dll, ntdll.dll, Ole32.dll. The GetModuleHandleA function is utilized to retrieve a handle for these DLLs. The malware generates 32 random bytes by calling the CryptGenRandom routine (this will be used as the ChaCha key):

EIP	00A707E	D 6A 2 F FF 7 2 FF D	5 08 D		push edi push 20 push dword p call eax	State of the state of the state of the	+8]	eax:cr >	x8 Defa	875W_C1 0 x875W_C0 0 x875W_E5 0 875W_SF 0 x875W_P 0 x875W_U 0 efault (stdcall)
	32.CryptGenRa 7F2 conti.exe								2: 3: 4:	: [esp+8] 04C3F8F8
Dump 1		Uump 3	💷 Dump 4	🚛 Dump 5	Watch 1	[x=] Locals	Struct	04C3F734 030EF 04C3F738 00000 04C3F73C 04C3F	020	
Address He 04C3F8F8 A0		CO 17 CB 7 BO DE BD F	D 37 96 12 5 B3 11 91	57 FE 98 E8 87 43 04 53	ASCII	.wþ.è				

Figure 63

There is also a call to CryptGenRandom that generates 8 random bytes, which will be used as the ChaCha8 nonce (this is the moment when we can tell for sure that the encryption algorithm for files is ChaCha8):

<ul> <li>00A70813</li> <li>00A70814</li> <li>00A70816</li> </ul>	56 6A 08 FF 75 08	push esi push 8 push dword ptr ss: ebp+	-		x875W_C1 0 x875W_C0 0 x875W_E5 0 x875W_5F 0 x875W_P 0 x875W_U 0
COA70819	FF DO	call eax		eax:Cr v	Default (stdcall)
eax= <advapi32.cryptgenrandom .text:00A70819 conti.exe:\$10</advapi32.cryptgenrandom 					1: [csp] 0300180 (article context) 2: [csp+4] 0000008 3: [csp+6] 04(3F8F0 4: [csp+C] 030DAEB0 L"C:\\" 5: [csp+10] 04(3FBCE L"WINRE_BACKUP_PARTITION.MA
💭 Dump 1 🚺 Dump 2 💷 D	ump 3 🔛 Dump 4 🔛 Dump	5 🛞 Watch 1 🛛 [x=] Locals	& Struct	04C3F734 030EFCI 04C3F738 000000	08
Address Hex Address Hex 04C3F8F0 B0 9B 9F EA AF 2B 0	C 48 A0 5D 5B 37 2C C0 17	ASCII CB *ê+ÌH ][7,À.Ë	1	04C3F73C 04C3F8	0

Figure 64

The ChaCha8 key and nonce are encrypted using the RSA public key:

00A708F1 6A 01     00A708F3 6A 00	push 20C push ecx push edi push 0 push 1 push dword ptr ss:[ebp+C] call eax	eax:Cr v	A0: ITM_0 5 (EmpLy)         A0: ITM_r 5 (EmpLy)           x875tuLsWord 0000         x875W_C2 0           x875W_B 0         x875W_C2 0           x875W_S 0         x875W_C2 0           x875W_S 0         x875W_S 0           x875W_S 0         x875W_U 0           Default (stdall)         ▼           1:         [esp1 03064208 <dcpgenkey>           2:         [esp4] 00000000</dcpgenkey>
.text:00A708F8 conti.exe:\$108F8 #FCF8			3: [esp+8] 00000001 4: [esp+C] 0000000 5: [esp+10] 04C3F918
Ump 1 Ump 2 Ump 3 Ump 4 Ump 5	🧐 Watch 1 🛛 🖉 Locals 🖉 Struct 04C3F72	4 030E42 8 000000	00
Address Hex	A3C11 04C2E72	C 000000	
04C3F918 A0 50 58 37 2C C0 17 C8 7D 37 96 12 57 F 98 E3 04C3F928 85 2C 58 88 9A 80 DE 80 F5 83 11 91 75 40 4 53 04C3F938 80 98 9F EA AF 28 CC 48 00 00 00 00 00 00 00 00 00 04C3F938 80 98 9F EA AF 28 CC 48 00 00 00 00 00 00 00 00 00 00	μ, [., *p½0*C.S 04C3F73 04C3F73	4 04C3F9 8 04C3F7 C 000002	18 CC

Figure 65

The ransomware retrieves file system attributes for the targeted file:

■IP 00A6F270 00A6F271	56 FF DO	push esi call eax	esi:L" eax:Ge v	Default (stdcall)	▼ 5 🚔 🗌 Unlock
<pre>eax=<kernel32.getfileattribu .text:00a6f271="" conti.exe:\$f2<="" pre=""></kernel32.getfileattribu></pre>				2: [esp+4] 04C3 3: [esp+8] 04C3	F8F0 F8A0 &L"C:\\\$WINRE_BACKUP_PARTITI
		🛲 Duna E 🦓 Washed Teallands 🧐	O4C3F4FC 030E18	BF8 L"C:\\\$WINRE_	BACKUP_PARTITION.MARKER"

Figure 66



The CreateFileW API is utilized to open the targeted file (0xC0000000 = **GENERIC\_READ** | **GENERIC\_WRITE**, 0x3 = **OPEN\_EXISTING**):

ODAFF28A GA 00     ODAFF28A GA 03     ODAFF28A GA 03     ODAFF28C GA 00     ODAFF28C	push 0 push 0 push 0 push 0 push 0 push comoono push est call eax	esi:L" eax:Cr v	x87TW_6 3 (Empty) x87TW_7 3 (Empty) x875tatusWord 0000 x875W_6 0 x87SW_C3 0 x87SW_C2 0 x875W_6 0 x87SW_C0 0 x87SW_E5 0 x875W_5F 0 x87SW_F 0 x87SW_U 0 Default(stdcal)
eax= <kernel32.createfilew> (76ACDDE0) .text:00A6F2C6 conti.exe:\$F2C6 #E6C6</kernel32.createfilew>			2: [esp+4] C0000000 3: [esp+8] 00000000 4: [esp+C] 00000000 5: [esp+10] 00000003
Jump 1         Jump 2         Jump 3         Jump 4         Jump 5           Address         Hex         04C3F918         ZC         SE         96         D         28         28         F6         78         39         FE         62         79         4E         Dump 5           04C3F918         ZC         SE         96         D         28         82         B8         F6         78         39         FE         62         79         4E         70         4C4         59         F0         46         C6         81         C2         A4         84         04C3F928         82         88         F6         78         39         FE         62         79         4E         72         4E         79         4E         72         4E         79         4E         74 <t< td=""><td>Watch 1         IX=I Locals         IX=Struct         04C3F           ASCII         04C3F         04C3F         04C3F           (I_M.WD+, 0x9bbyWr         04C3F         04C3F         04C3F           • 260rm.80F4.A*         04C3F         04C3F         04C3F</td><td>4E4         030E1B           4E8         C00000           4E0         000000           4F0         000000           4F4         000000           4F8         000000           4F8         000000           4F6         000000</td><td>00 00 03 00</td></t<>	Watch 1         IX=I Locals         IX=Struct         04C3F           ASCII         04C3F         04C3F         04C3F           (I_M.WD+, 0x9bbyWr         04C3F         04C3F         04C3F           • 260rm.80F4.A*         04C3F         04C3F         04C3F	4E4         030E1B           4E8         C00000           4E0         000000           4F0         000000           4F4         000000           4F8         000000           4F8         000000           4F6         000000	00 00 03 00

Figure 67

The malware comes with two hard-coded lists of file extensions that will be encrypted. It's important to mention that if the file extension doesn't belong to these lists, it will be partially encrypted using a different execution flow that will be explained later (the full lists are available in the Appendix):

IP		A6D660	FF D	0		call eax					ax:St V De	fault (stdcall)	)		▼ 5 ‡ 🗌 Unl
a na sa la Juna	<		C 45 D (2)								> 1:	[esp] 03	BOE1C30 L"C:	\\BOOTSECT.	
eax= <sh1wa< td=""><td>api.strst</td><td>r1W&gt; (75)</td><td>-C45D0)</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>2:</td><td>[esp+4]</td><td>04C3F6FD L" 04C3F918</td><td>. 4dd"</td><td></td></sh1wa<>	api.strst	r1W> (75)	-C45D0)								2:	[esp+4]	04C3F6FD L" 04C3F918	. 4dd"	
text:00A6											4:	[esp+C]	04C3F8F0		
LEXT: OUA6	D000 C00	LI.exe: M	7660 #C	460							5:		04C3F6FD L	".4dd"	
Ump 1	U Dum	p2 🚚	Dump 3	U Dump	1 Dump !	Watch 1	[x=] Locals	s 🐉 Struct			030E1C30 04C3F6FD		OTSECT.BAK"		
Address   H	lex					ASCII	1				04C3F918				
4C3F918	7 13 89	A4 F8 58	OB 64	37 54 3D 0	4 87 5A AB	BF C MØX. d7T	=Z«?				04C3F8F0 04C3F6FD	L".4dd"			
4C3F928 7	74 OC 78	83 A9 10	15 FD	11 B7 34 /	3 64 80 56	DD t.x.@ý. 1E ?F¢òcj.	4£d.V.			04C3EBC4	04C3F6F1	L".4d1"	1		
4C3F948 C	3 8B 22	6B 79 79	75 5A	3E 7C C5 C	5 10 76 96	97 Å. "kyyuZ>	ÂÂ. V				04C3F00D 04C3EFFD	L".accdb			
04C3F958 1	LD DD CB	0A 96 7C	DC C6	62 A7 61 C	8 47 40 09	4E ÝË  ÜÆb§ 08 [b.ß#%8.	AÈGLÉN			04C3EBD0	04C3EFED	L". accde			
4C3F968 9	08 90 B7	DD 33 41	39 25	DD E8 BD E	5 E5 33 C3	F1 Ý3A9%. Ø	Suâ3Ăñ					L".accdr			
4C3F988 A	AO ED CG	87 91 15	1F DC	51 E3 4A 9	8 F6 BE 09	7 14ÜQā	J. 0%. C			04C3EBD8	04C3EFCD 04C3EFBD	L".accdt			
								Figure 68	?						
	• 00	AGE1D4	57	4 85 8C F	E FF FF	push edi		Figure 68	2	e	di:L"	37.5W_SF 0	x875W_P	0 x875W_U	0
19	• 00			0	E FF FF			5	?	e	di:L"			0 x875W_U	
112	• 00	AGE1DB	57 FF 0	0	E FF FF	push edi		5	2	e	di:L"	fault (stdcall)	)		▼ 5 🗘 🗆 Un
		AGE1DB	57 FF 0	0	E FF FF	push edi		5	2	e	De	fault (stdcall) [esp] 03 [esp+4]	04C3F72D L"C:	\\BOOTSECT.	▼ 5 🗘 🗆 Un
		AGE1DB	57 FF 0	0	E FF FF	push edi		5	2	e	di:L" ax:St v > 1: 2: 3:	fault (stdcall) [esp] 0: [esp+4] [esp+8]	04C3F72D L"C: 04C3F72D L" 04C3F918	\\BOOTSECT.	▼ 5 🗘 🗆 Un
eax= <sh1wa< th=""><th>api.StrSt</th><th>AGE1DC AGE1DC rIW&gt; (75)</th><th>57 FF D C45D0)</th><th>0</th><th>E FF FF</th><th>push edi</th><th></th><th>5</th><th>?</th><th>e</th><th>De</th><th>fault (stdcall) [esp] 0: [esp+4] [esp+8] [esp+C]</th><th>04C3F72D L"C:</th><th>\\BOOTSECT.</th><th>▼ 5 € 🗌 Un BAK''</th></sh1wa<>	api.StrSt	AGE1DC AGE1DC rIW> (75)	57 FF D C45D0)	0	E FF FF	push edi		5	?	e	De	fault (stdcall) [esp] 0: [esp+4] [esp+8] [esp+C]	04C3F72D L"C:	\\BOOTSECT.	▼ 5 € 🗌 Un BAK''
IP eax= <sh]wa text:00A6</sh]wa 	api.StrSt	AGE1DB AGE1DC rIW> (75) ti.exe:\$	57 FF D 57 FC 45D0) E1DC #D	0 SDC		push edi call eax		op+esi*4-174	,	04C3F5B0	di:L" ax:St > ) 1: 2: 3: 4: 5: 030E1C30	fault (stdcall) [esp] 0: [esp+4] [esp+8] [esp+C] [esp+10] L "C: \\B0	30E1C30 L"C: 04C3F72D L" 04C3F918 04C3F8F0	\\BOOTSECT.	▼ 5 € 🗌 Un BAK''
eax= <sh]wa .text:00A6</sh]wa 	api.StrSt	AGE1DB AGE1DC rIW> (75) ti.exe:\$	57 FF D C45D0)	0		watch 1	ptr ss:[ek	pp+esi*4-174	,	04C3F5B0 04C3F5B4 04C3F5B8	di:L" ax:St De 1: 2: 3: 4: 5: 030E1C30 04C3F72D 04C3F718	fault (stdcall) [esp] 0: [esp+4] [esp+8] [esp+C] [esp+10] L "C: \\B0	04C3F72D L"C: 04C3F72D L" 04C3F918 04C3F8F0 04C3F8A0 &	\\BOOTSECT.	▼ 5 🗘 Ur BAK"
text:00A6	api.StrSt	AGE1DB AGE1DC rIW> (75) ti.exe:\$1	57 FF D (1) (1) (1) (1) (1) (1) (1) (1) (1) (1)	o sDC	ŧ 🚛 Dump t	push edi call eax	ptr ss: [ek	op+esi*4-174	2	04C3F5B0 04C3F5B4 04C3F5B8 04C3F5B8	di:L" ax:St De 1: 2: 3: 4: 5: 030E1C30 04C3F72D 04C3F918 04C3F918	fault (stdcall) [esp] 0: [esp+4] [esp+8] [esp+2] [esp+10] L"C: \\B0 L".vdi"	30E1C30 L"C: 04C3F72D L" 04C3F918 04C3F918 04C3F8F0 04C3F8A0 & 04C3F8A0 & 04C3F8A0 &	\\BOOTSECT. '.vdi" bL"C:\\BOOTS	▼ 5 € 🗌 Ur BAK''
eax= <sh1wa< td=""><td>api.StrSt SEIDC con US SEIDC con US Dum Hex E7 13 89 74 0C 78</td><td>AGE1DB AGE1DC rIW&gt; (75) ti.exe:\$1 p2</td><td>57 FF D C45D0) 1DC #D Dump 3 08 64 15 FD</td><td>5DC</td><td>4 Dump 1</td><td>push edi call eax i @ Watch 1 ASCII SF ÇF0X.d7T D.t.x.@ý</td><td>ptr ss:[ek</td><td>op+esi*4-174</td><td></td><td>04C3F5B0 04C3F5B4 04C3F5B8 04C3F5B8 04C3F5B0 04C3F5C0</td><td>di:L" ax:St De 1: 2: 3: 4: 5: 030E1C30 04C3F72D 04C3F918 04C3F918</td><td>fault (stdcall) [esp] 0: [esp+4] [esp+4] [esp+7] [esp+10] L"C:\\B0 L".vd1"</td><td>04C3F72D L"C: 04C3F72D L" 04C3F918 04C3F8F0 04C3F8A0 &amp;</td><td>\\BOOTSECT. '.vdi" bL"C:\\BOOTS</td><td>▼ 5 € 🗌 Ur BAK''</td></sh1wa<>	api.StrSt SEIDC con US SEIDC con US Dum Hex E7 13 89 74 0C 78	AGE1DB AGE1DC rIW> (75) ti.exe:\$1 p2	57 FF D C45D0) 1DC #D Dump 3 08 64 15 FD	5DC	4 Dump 1	push edi call eax i @ Watch 1 ASCII SF ÇF0X.d7T D.t.x.@ý	ptr ss:[ek	op+esi*4-174		04C3F5B0 04C3F5B4 04C3F5B8 04C3F5B8 04C3F5B0 04C3F5C0	di:L" ax:St De 1: 2: 3: 4: 5: 030E1C30 04C3F72D 04C3F918 04C3F918	fault (stdcall) [esp] 0: [esp+4] [esp+4] [esp+7] [esp+10] L"C:\\B0 L".vd1"	04C3F72D L"C: 04C3F72D L" 04C3F918 04C3F8F0 04C3F8A0 &	\\BOOTSECT. '.vdi" bL"C:\\BOOTS	▼ 5 € 🗌 Ur BAK''
text:00A6	Api.StrSt SEIDC con Hex E7 13 89 74 0C 78 55 46 98	AGE1DB AGE1DC rIW> (75) ti.exe:\$1 p 2 A4 F8 58 83 A9 10 B 86 A2	57 FF E C45D0) 1DC #D Dump 3 08 64 15 FD F2 63	5DC 5DC 57 54 3D (11 B7 34 / 11 B7 34 / 18 A 05 2	t ∰ Dump 1 14 87 5A AB 13 64 80 56 5 58 72 €1	push edi call eax i @ Watch 1 ASCII 3F €¤ØX.d7T DD t.x.⊕.ý 12 (?F¢Cci.	ptr ss:[ek 	op+esi*4-174	2	04C3F5B0 04C3F5B4 04C3F5B4 04C3F5BC 04C3F5C0 04C3F5C0 04C3F5C4	111L" 22:3 23:3 41:5 030E1C300 04C3F72D 04C3F918 04C3F8F0 04C3F8F0 04C3F8F0 04C3F72D	fault (stdcall) [esp] 0: [esp+4] [esp+4] [esp+2] [esp+10] L"C:\\B0 L".vd1" L".vd1" L".vd1"	30E1C30 L"C: 04C3F72D L" 04C3F918 04C3F918 04C3F8F0 04C3F8A0 & 04C3F8A0 & 04C3F8A0 &	\\BOOTSECT. '.vdi" bL"C:\\BOOTS	▼ 5 € 🗌 Ur BAK''
eax= <sh1wa text:00A6 </sh1wa 	Ap1.StrSt SE1DC con U Dum Hex E7 13 89 74 0C 78 3F 46 98 23 88 22 20 DDD CB	AGEIDE AGEIDC rIW> (75) ti.exe:\$1 p2 A4 F8 58 83 A9 10 1B 86 A2 68 79 79 0A 96 7C	57 FF C 57 57 57 57 57 57 57 57 57 57 57 57 57	0 5DC 37 54 3D ( 11 87 34 / A1 8A 05 2 36 7C C5 ( 22 A7 61 (	4	push edi call eax ASCII SF Ç =0X.d7T DD T.X.=0.ý. F ?F côci 97 Å. "kyyuz>1 E ?F côci 97 Å. "kyyuz>1 E .YE. júžbS	ptr ss: [ek 	op+esi*4-174	, ^	04C3F5B0 04C3F5B4 04C3F5B8 04C3F5B8 04C3F5B2 04C3F5C4 04C3F5C4 04C3F5C5 04C3F5C5	1111 " X:SI V De 11 22 33 44 55 04C3F918 04C3F918 04C3F918 04C3F8A0 04C3F8A0 04C3F72D 04C3F72D 04C3F72D 04C3F72B	fault (stdcall) [esp+4] [esp+4] [esp+2] [esp+10] L"C:\\B0 L".vd1" L".vd1" L".vd1" L".vd1"	30E1C30 L"C: 04C3F72D L" 04C3F918 04C3F918 04C3F8F0 04C3F8A0 & 04C3F8A0 & 04C3F8A0 &	\\BOOTSECT. '.vdi" bL"C:\\BOOTS	▼ 5 € 🗌 Ur BAK''
eax= <sh1wa text:00A6 U Dump 1 Address H 04C3F918 E 04C3F938 3 04C3F938 3 04C3F938 3 04C3F958 1 04C3F958 1</sh1wa 	Api.StrSt SE1DC con Hex F7 13 89 74 0C 78 8F 46 9B S3 8B 22 LD DD CB S2 5B 62	AGE1DB AGE1DC rIW> (75) ti.exe:\$ p 2 4 A4 F8 58 83 A9 10 B 86 A2 66 79 79 0A 96 7C S5 DF 23	57 FF C 545D0) 51DC #D Dump 3 08 64 15 FD 52 63 75 5A DC C6 8F 38	5DC 5DC 5DC 50 51 52 54 50 54 50 54 50 54 50 50 50 50 50 50 50 50 50 50	4 Dump 1 14 87 5A AB 3 64 80 56 5 58 72 C1 5 10 76 20 8 47 4C C9 4 AB C6 AD	push edi           call eax           call eax           ascili           Ascili           asc, mox.drT           f.e., mox.drT           f.e., rex.dr.drT           f.e., rex.dr.dr.dr.dr.dr.dr.dr.dr.dr.dr.dr.dr.dr.	ptr ss: eb [x=] Locals =2«? 4£d.v. .%xrA. AA.v. aÈGLÈN r«£.0	op+esi*4-174		04C3F5B4 04C3F5B4 04C3F5B8 04C3F5B8 04C3F5B8 04C3F5C0 04C3F5C4 04C3F5C6 04C3F5C6 04C3F5C6 04C3F5D4	1111 " XISI V De 11 22 33 43 53 0403F720 0403F720 0403F720 0403F720 0403F720 0403F720 0403F720 0403F721 0403F669	fault (stdcall) [esp+4] [esp+4] [esp+2] [esp+10] [esp+10] [esp+10] [esp+10] [esp+10] [esp+10] [esp+10] [esp+10] [esp+10] [esp+10] [esp+10] [esp+10] [esp+2] [esp+2] [esp+2] [esp+2] [esp+3] [esp+3] [esp+3] [esp+3] [esp+3] [esp+3] [esp+3] [esp+3] [esp+3] [esp+3] [esp+3] [esp+3] [esp+3] [esp+3] [esp+3] [esp+4] [	30E1C30 L"C: 04C3F72D L" 04C3F918 04C3F918 04C3F8F0 04C3F8A0 & 04C3F8A0 & 04C3F8A0 &	\\BOOTSECT. '.vdi" bL"C:\\BOOTS	▼ 5 \$ UI BAK"
eax= <sh1wa text:00A6 U Dump 1 Address H 4C3F918 E 4C3F928 7 94C3F948 0 4C3F948 0 4C3F948 0 4C3F958 1 4C3F968 9 4C3F978 0</sh1wa 	Ap1.StrSt SE1DC con SE1DC con Con Con Con Con Con Con Con C	AGEIDE AGEIDC rIW> (75) ti.exe:\$1 p2 A4 F8 58 83 A9 10 18 86 A2 68 79 79 0A 96 7C 68 DF 23 DD 33 41 87 91 15	57 FF C 57 57 57 57 57 57 57 57 57 57 57 57 57	5DC	4 ■ Dump 1 14 87 5A AB 13 64 80 56 15 58 72 C1 15 10 76 92 15 10	push edi call eax ASCII 3F Ç =0X.d7T 37 Å. "kyyuz>1 27 Å. "kyyuz>1 4. YE. jüzbS	ptr ss: ek	op+esi*4-174	,	04C3F580 04C3F584 04C3F584 04C3F58 04C3F58 04C3F5C4 04C3F5C4 04C3F5C4 04C3F5C4 04C3F5C4 04C3F5D0 04C3F5D0	11:11 XX:ST V De 1: 2: 3: 4: 5: 0300E1C30 04C3F720 04C3F720 04C3F720 04C3F720 04C3F720 04C3F720 04C3F720 04C3F720 04C3F720 04C3F720 04C3F721 04C3F721	fault (stdcall) [esp] 0.0 [esp+4] [esp+4] [esp+2] [esp+10] [esp+10] L": v\kar L": vkd" L": vd1" L": vkd" L": vvd1" L": vmdk" L": vmm" L": vmsn"	0052C30 L"C: 04C3F72D L" 04C3F918 04C3F918 04C3F8A0 & 04C3F8A0 & 0TSECT.BAK"	\\BOOTSECT. '.vdi" bL"C:\\BOOTS	▼ 5 🗘 Ur BAK"

Figure 69

The process writes the encrypted ChaCha8 key and nonce to the encrypted file:



eax= <kernel32.writefle> (76ACE250) .text:00A6E254 conti.exe:\$E254 #D654 .text:00A6E254 conti.exe:\$E254 #D654 .text:00A6E251 conti.exe:\$E254 #D654 .text:00A6E251 conti.exe:\$E254 #D654 .text:00A6E251 conti.exe:\$E254 #D654 .text:00A6E251 conti.exe:\$E254 #D654 .text:00A6E254 conti.exe:\$E254 #D6</kernel32.writefle>	push 0         push ecx         mov ecx, dword ptr ss: [ebp-c]         push edi         add ecx, ebx         push ecx         push ecx         push ecx         push dtd ptr ss: [ebp-10]         call eax         call eax         push dtd ptr ss: [ebp-10]         call eax         call eax         AB 3F (C. How, dTr=., Ze?)	x6/1%_5 (tm_b5 (tmpty) x8/ x87Statusword 0000 x87Sw_B 0 x87Sw_C3 0 x87Sw_L5 0 x87Sw_C9 0 x87Sw_C1 0 x87Sw_C1 0 x87Sw_C9 0 x87Sw_C1 0 x87Sw_C1 0 x87Sw_C9 0 x87Sw_C1 0 x87Sw_C1 0 x87Sw_C	x875W_C2 0 x875W_E5 0 x875W_U 0

Figure 70

There are 3 different cases depending on the file size: small files (< 1MB), medium files (between 1MB and 5MB), and large files (> 5MB). In the case of medium and large files, there exist 2 subcases depending on the file extension (if it belongs to the targeted lists or not). The following 10-byte buffer that contains a marker (0x24) and the file size (0x2000) is appended to the encrypted file:

	push 0 push ecx, dword ptr ss:[ebp-C] push edi add ecx, ebx push dword ptr ss:[ebp-10] call eax	x8/IW_b s (EmpLy) x8/IW_/ s (EmpLy) x8/StatusWord 0000 x8/SYW_E 0 x8/SW_C3 0 x8/SW_C2 0 x8/SW_E1 0 x8/SW_C3 0 x8/SW_E5 0 x8/SW_SF 0 x8/SW_P 0 x8/SW_U 0
eax= <kernel32.writefile> (76ACE250) .text:00A6E254 conti.exe:\$E254 #D654</kernel32.writefile>		1: [esp] 0000308 2: [esp+4] 04C8F720 3: [esp+4] 0000000A 4: [esp+C] 04C3F55C 5: [esp+10] 0000000
Ump 1 Ump 2 Ump 3 Ump 4 Um Dump 5	Watch 1 [x=] Locals Struct 04C3F534 00 04C3F538 04	C3F720
Address Hex 04C37720 24100 00 00 00 00 00 00 00 00 00 BB 46 3A 3	ASCII 04C3F540 04 04C3F540 04 04C3F544 00 04C3F544 00 04C3F54 04C	C3F55C

Figure 71

#### The binary reads the file content using the ReadFile function:

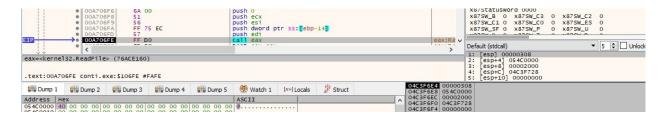


Figure 72

Address	He	<															ASCII
054C0000	EB	52	90	4E	54	46	53	20	20	20	20	00	02	08	00	00	ER.NTFS
054C0010	00	00	00	00	00	F8	00	00	3F	00	FF	00	00	08	00	00	ø?.ÿ
054C0020	00	00	00	00	80	00	80	00	GD	4A	F1	09	00	00	00	00	mJñ
054C0030	00	00	0C	00	00	00	00	00	02	00	00	00	00	00	00	00	
054C0040	F6	00	00	00	01	00	00	00	2F	AD	C9	A2	BC	C9	A2	4A	
054C0050	00	00	00	00	FA	33	CO	8E	DO	BC	00	7C	FB	68	CO	07	ú3À.Đ¼. ûhÀ.
054C0060	1F	1E	68	66	00	CB	88	16	0E	00	66	81	3E	03	00	4E	hf.Ef.>N
054C0070	54	46	53	75	15	<b>B</b> 4	41	BB	AA	55	CD	13	72	OC.	81	FB	TFSu. A≫ªUÍ.rû
054C0080	55	AA	75	06	F7	C1	01	00	75	03	E9	DD	00	1E	83	EC	Uªu.÷Áu.éÝì
0540090	18	68	1A	00	<b>B4</b>	48	8A	16	0E	00	8B	F4	16	1F	CD	13	.h´нô1.
054C00A0	9F	83	C4	18	9E	58	1F	72	E1	3B	06	OB	00	75	DB	A3	ÄX.rá;uÛ£
054C00B0	OF	00	C1	2E	OF	00	04	1E	5A	33	DB	<b>B</b> 9	00	20	2B	C8	
05400000	66	FF	06	11	00	03	16	OF	00	8E	C2	FF	06	16	00	E8	fÿè
054C00D0	4B	00	2B	C8	77	EF	<b>B</b> 8	00	BB	CD	1A	66	23	CO	75	2D	K.+Èwï»1.f#Au-
054C00E0	66	81	FB	54	43	50	41	75	24	81	F9	02	01	72	1E	16	f.ûTCPAu\$.ùr
054C00F0			BB					16		09	00	66	53	66	53	66	h.».hRhfSfSf
054C0100	55	16	16	16	68	<b>B</b> 8	01	66	61	OE	07	CD	1A	33	CO	BF	Uhfa1.3A¿
054C0110																	
054C0120	06	66	A1	11	00	66	03	06	10	00	1E	66	68	00	00	00	.fiffh
054C0130	00	66	50	06	53	68	01	00	68	10	00	<b>B</b> 4	42	8A	16	0E	.fP.Shh B
05400140	00	10	1.5	00	E 4	CD	1.7	CC.	50	E D	E 0	CC.	E O	CC.	E 0.	1	At fyrtyfy

Figure 73





The content is encrypted using the ChaCha8 algorithm implemented by Conti:

Address	Hex	(															ASCII
054C0000	5C	12	56	10	C7	23	<b>B</b> 7	0A	56	5C	66	53				E1	N.V.Ç#V\fsc.öá
054C0010	5E	OD	CD	90	B4		89				D0		B6	48	B5	A3	^.1. ;.h.õĐ ¶Hµ£
054C0020				51				55					B5	25	2D	83	òe.QU2\$@.µ%
054C0030	1E	<b>B</b> 8	16	1B				15	87	3B	32	75			72		t.ö;2u.⁼rW
054C0040		7D	93	5C			FA				E2				58		Ä}.\D.úÅ.QâøXÌ
054C0050	1E			69				CE						C8		84	.U.i.*.1Ê\$D¥¿È
054C0060	B1	4E	37	<b>B</b> 7				5E			98			F3		75	±N7·8°Ô^mµ.¥zó u
054C0070				77			9B		66		7E			~		22	UGëw.i.Yf.~.a"
054C0080				47			D4				75				52		iêZGÔ¥%Ûuó.URb
054C0090				B5				AA						E9		D2	î¤.μvª åépò
054C00A0															D7		Ä.@.qêG'Cúx÷
054C00B0		11															
054C00C0			F9												A7		¿.u}ü.>vÊ .JUe§Ő
054C00D0	1D	40		98				D4							FA		.@GáÔjÚY¾ú.
054C00E0	EB	62		6A			70				D2			F9		E2	ëbëj07pYð Ö'Eù\â
054C00F0		ЗD		70							1F				1A		Æ=úpC T.dÒ.Ŏ
054C0100	96			7A			DB							A7		BB	.4=z[T07°(.§.»
054C0110		85		E3	77	BC		40	7D	14	7D	82			92		sāw%0@}.}.i6.R
054C0120	F6			DE			70		A1	A1		OB		54		B9	ÖBaÞ"Óp'iiA.ÈT.'
054C0130	33	46	20	36	40	58	D9	A3	44		3B	54	59	CC	1A	4F	3F 6@XU£D7;TY1.0
05400140	00	70	06	D/	AE	70	61	14	DE	10	0.9	41	PC	EE	C1	AA	I'L BY AV AN

#### Figure 74

A snippet of the ChaCha8 algorithm developed by the ransomware is presented in figure 75.

🗾 🚄 🖼		
.text:00A65C83		
.text:00A65C83	loc A65	C83:
.text:00A65C83	add	edi, [ebp+var 60]
.text:00A65C86	mov	eax, [ebp+var_58]
.text:00A65C89	xor	ebx, edi
.text:00A65C8B	add	ecx, [ebp+var_78]
.text:00A65C8E	rol	ebx, 10h
.text:00A65C91	xor	edx, ecx
.text:00A65C93	add	eax, ebx
.text:00A65C95	mov	[ebp+var_5C], edi
.text:00A65C98	mov	[ebp+var_58], eax
.text:00A65C9B	xor	eax, [ebp+var_60]
.text:00A65C9E	rol	eax, 0Ch
.text:00A65CA1	add	edi, eax
.text:00A65CA3	rol	edx, 10h
.text:00A65CA6	xor	ebx, edi
.text:00A65CA8	mov	<pre>[ebp+var_5C], edi</pre>
.text:00A65CAB	mov	edi, [ebp+var_58]
.text:00A65CAE	rol	ebx, 8
.text:00A65CB1	add	edi, ebx
.text:00A65CB3	mov	[ebp+var_58], edi
.text:00A65CB6	xor	edi, eax
.text:00A65CB8	mov	<pre>eax, [ebp+var_4C]</pre>
.text:00A65CBB	add	eax, edx
.text:00A65CBD		edi, 7
.text:00A65CC0		[ebp+var_4C], eax
.text:00A65CC3	xor	eax, [ebp+var_78]
.text:00A65CC6	rol	eax, OCh
.text:00A65CC9	add	ecx, eax
.text:00A65CCB	xor	edx, ecx
.text:00A65CCD	mov	[ebp+var_84], ecx
.text:00A65CD3	mov	ecx, [ebp+var_4C]
.text:00A65CD6 .text:00A65CD9	rol	edx, 8
.text:00A65CD9	add	ecx, edx [ebp+var 60], edx
.text:00A65CDB	mov	
.text:00A65CDE	add	<pre>edx, [ebp+var_64] edx, [ebp+var 7C]</pre>
.text:00A65CE4	xor	esi, edx
.text:00A65CE6	mov	[ebp+var 4C], ecx
.text:00A65CE9	xor	ecx, eax
.text:00A65CEB	rol	esi, 10h
.text:00A65CEE	mov	eax, [ebp+var 48]
. CCAC. OUROJCEE	mov	cur, [cuptvar_40]

#### Figure 75

The encrypted data is written to the file using the WriteFile API:



ODA6E247 GA 00     ODA6E247 S1     ODA6E249 S1     ODA6E24A 88 4D F4     ODA6E24D 57     ODA6E24E 0 C8     ODA6E24E 0 C8     ODA6E24E S1     FF 75 F0     ODA6E351 FF 75 F0     ODA651 FF 75 F0     ODA651 FF 75 F0     ODA651 FF 75 F0     ODA651 FF 75     ODA651 F	push 0 push ecx push ecx add ecx,dword ptr ss:[ebp-C] add ecx,ebx push ecx push dword ptr ss:[ebp-10] call eax	x8/1w_b s (EmpLy)         x8/1w_r s (EmpLy)           x8/5w_b 0         x875w_c3         x875w_c2           x875w_c5         x875w_c0         x875w_c5         0           x875w_c5         x875w_c9         x875w_c1         0           x875w_c5         x875w_c9         x875w_c2         0           x875w_c5         x875w_c9         x875w_c9         0           x875w_c5         x875w_c9         x875w_c9         x875w_c9           Default (stdcall)         v         s \$\overline{\circles}\$         urbline{\circles}\$	lock
eax= <kernel32.writefile> (76ACE250) .text:00A6E254 conti.exe:\$E254 #D654</kernel32.writefile>		11         [esp]         00000308           21         [esp+4]         054C000           31         [esp+8]         0002000           41         [esp+4]         0002000           41         [esp+10]         00000000           51         [esp+10]         00000000	
##         Dump 1         ##         Dump 2         ##         Dump 3         ##         Dump 4         ##         Dump 5           Address         Hex         05 ± 00000         50         12         56         10         7         23         87         0A         56         52         66         53         C7         06         D6         55           05 ± 00000         50         52         56         10         C7         23         87         0A         56         52         66         53         C7         06         D6         E           05 ± 00000         50         52         56         10         C7         23         87         0A         56         52         66         53         C7         06         D6         E	ASCII	OtC93E50C         00000308           04C3F6C4         05000           04C3F6C4         050000           04C3F6C5         04C3F6C4           04C3F6C5         04C3F6C4	

Figure 76

### The ".PSFUX" extension is added to the file name:

EIP	00A70B8B 00A70B8C 00A70B8F	FF 75 EC	push esi push dword ptr ss:[ebp-14] call eax	test1:1         x87.5W_F         0         x87.5W_P         0         x87.5W_U         0           test3100
	B8F conti.exe:			1: [esp] 030E1C30 L"C:\\BOOTSECT.BAK" 2: [esp+4] 03116ED0 L"C:\\BOOTSECT.BAK.PSFUX" 3: [esp+6] 0300AEB0 L"C:\\" 4: [esp+C] 04C3FBCE L"00TSECT.BAK" 5: [esp+C] 04C3FBCE L"00TSECT.BAK"
🚛 Dump 1	Dump 2	Dump 3 🔛 Dump 4	🕮 Dump 5 🛞 Watch 1 🛛 🕬 Struct	04C3F735 030E1C30  L"C:\\BOOTSECT.BAK"  04C3F73C 03116ED0  L"C:\\BOOTSECT.BAK.PSFUX"

Figure 77

### The ransom note that is created in every encrypted directory is displayed below:

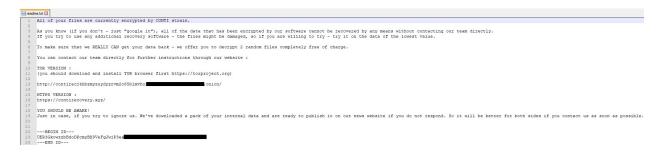


Figure 78

An example of an encrypted file (file size < 1MB) is highlighted in the next 2 pictures:



BOOTSECT.BAK.PSFUX

Offs	et(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
0000	0000	5C	12	56	1C	C7	23	B7	0A	56	5C	66	53	C7	06	D6	E1	N.V.Ç# ·.V\fSÇ.Öá
0000	0010	5E	OD	CD	90	B4	зB	89	68	04	F5	DO	20	B6	48	B5	A3	^.í. ;%h.őĐ ¶Hu£
0000	0020	F2	65	1E	51	1C	8E	93	55	32	24	A9	07	B5	25	2D	83	òe.Q.Ž"U2\$©.µ%−f
0000	0030	1E	B8	16	1B	74	13	F6	15	87	3B	32	75	8D	B2	72	57	.,t.ö.‡;2u.*rW
0000	0040	C4	7D	93	5C	44	0C	FA	C3	00	51	E2	09	83	F8	58	CC	Ä}"\D.úÃ.Qâ.føXÌ
0000	0050	1E	55	9C	69	8E	B3	10	CE	CA	24	DO	A5	BF	C8	13	84	.Uœi޳.ÎÊ\$Đ¥¿È."
0000	0060	B1	4E	37	B7	38	BO	D5	5E	6D	B5	98	A5	7A	F3	AO	75	±N7·8°Õ^mµ~¥zó u
0000	0070	55	47	EB	77	13	69	9B	59	66	0C	7E	19	EO	82	11	22	UGëw.i>Yf.~.à,."
0000	0800	69	EA	5A	47	AD	9E	D4	A5	25	DB	75	F3	97	55	52	62	iêZG.žÔ¥%Ûuó—URb
0000	0090	EE	A4	86	B5	76	02	2E	AA	AO	8E	1B	20	E5	E9	70	D2	î¤tµvª Ž. åépÒ
0000	00A0	C4	00	A9	06	71	EA	47	86	04	B9	12	5F	C7	FA	D7	F7	Ä.©.qêGt.¹. Çú×÷
0000	00B0	B7	11	0B	33	CE	88	04	7F	C6	55	Dl	D2	52	A6	F2	08	3Î^ÆUÑÒR¦ò.
0000	0000	BF	05	F9	7D	FC	2E	3E	76	CA	A8	9A	4A	55	65	A7	D5	¿.ù}ü.>vÊ″šJUe§Õ
0000	OODO	1D	40	5F	98	91	47	E1	D4	6A	DA	07	06	59	BE	FA	11	.@ ~`GáÔjÚY¾ú.
0000	00E0	EB	62	EB	6A	4F	37	70	59	FO	A8	D2	27	45	F9	5C	E2	ëbëj07pYð"Ò'Eù\â
0000	OOFO	C6	ЗD	FA	70	C7	7C	54	B7	64	07	1F	D2	8F	D5	1A	2D	Æ=úpÇ T·dÒ.Õ
0000	0100	96	34	B2	7A	5B	54	DB	37	BO	00	5F	28	07	A7	1F	BB	-4°z[TÛ7°. (.§.»
0000	0110	92	85	73	E3	77	BC	D2	40	7D	14	7D	82	ED	36	92	52	'sãw4000}.},16'R
0000	0120	F6	42	EO	DE	22	D3	70	B9	Al	Al	41	0B	C8	54	09	B9	öBàÞ"Óp¹;¡A.ÈT.¹
0000	0130	33	46	20	36	40	58	D9	A3	44	37	ЗB	54	59	CC	1A	4F	3F 6@XÙ£D7;TYÌ.O
0000	0140	0D	7C	06	B4	AF	7D	64	14	DE	4B	09	41	BC	5F	Cl	A4	. }d. ÞK. A4 Á#
0000	0150	C8	DC	EA	D2	5A	ЗF	10	32	56	79	D4	F2	ED	9E	E5	AB	ÈÜêÒZ?.2VyÔòížå«
0000	0160	A7	6B	1B	1B	OF	CA	E5	78	57	DF	2E	19	FC	1E	90	37	§kÊåxWßü7
0000	0170	F3	59	D6	DF	F5	48	7E	95	14	7C	16	CA	82	65	79	AD	óYÖßõH~•.∣.Ê,ey.
0000	0180	C5	98	52	DC	F5	C2	C4	45	1B	C2	88	49	09	lD	A4	53	Å~RÜÕÂÄE.Â^I¤S
0000	0190	EO	F2	B2	C8	DD	51	75	4E	26	C2	2B	D2	Al	F3	F6	64	àò°ÈÝQuN&Â+Ò;óöd

Figure 79

BOOTSECT.	BAK.	PSFL	JХ														
Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	OD	0E	OF	
00002000	E7	13	89	A4	F8	58	0B	64	37	54	3D	04	87	5A	AB	ЗF	G.%¤øX.d7T=.‡Z«?
00002010	74	0C	78	83	A9	10	15	FD	11	B7	34	A3	64	80	56	OD	t.xf©ý. 4£d€V.
00002020	3F	46	9B	1B	86	A2	F2	63	Al	8A	05	25	58	72	Cl	1E	?F>.t¢òc;Š.%XrÁ.
00002030	C3	8B	22	6B	79	79	75	5A	3E	7C	C5	C5	10	76	96	97	ë "kyyuZ> ÅÅ.v
00002040	1D	DD	CB	0A	96	7C	DC	C6	62	A7	61	C8	47	4C	C9	4E	.ÝË ÜÆb§aÈGLÉN
00002050	92	5B	62	85	DF	23	BE	38	07	09	72	0A	AB	C6	AD	D8	′[bβ‡¾8r.≪Æ.Ø
00002060	08	9D	B7	DD	33	41	39	25	OD	F8	BD	B5	E5	33	C3	Fl	∙Ý3A9%.ø‰µå3Ãñ
00002070	AO	ED	C6	87	91	15	1F	DC	51	E3	4A	98	F6	BE	09	C7	iƇ`ÜQãJ~ö¾.Ç
00002080	54	74	CE	CC	22	8A	B5	07	34	0D	08	B4	EE	6B	ЗD	01	TtÎÌ"е.4′îk≡.
00002090	7D	1D	F2	D4	7A	FF	Fl	50	EA	61	BF	FB	Dl	BE	EC	63	}.ôÔzÿñPêa¿ûѾìc
000020A0	2C	4B	F8	B5	96	EB	34	0E	42	EC	EB	38	D8	D4	98	6A	,Køµ-ë4.Bìë8ØÔ~j
000020B0	02	20	AC	08	59	C8	21	60	47	98	8E	5B	7E	AE	B6	BF	. ¬.YÈ!`G~Ž[~⊗¶¿
000020C0	1C	91	9A	98	lF	50	B1	0D	97	EC	51	FE	E2		9A	D9	.`š~.P±.—ìQþâ.šÙ
000020D0	EA	CC	D3		28	<b>B</b> 3	7A		23	Cl		9D	49	_	51	37	êÌÓô(³zS#ÁÉ.I»Q7
000020E0	0E	65	BF	El	90	E9	1F	5F	99	68	C8	Al	86	54	AB	E6	.e;á.é™hÈ;†T≪æ
000020F0	AA	E8	1D	F9	F4	77	19	0E	C1	24	0A	89	E5	14	BD	09	*è.ùôwÁ\$.‰å.⅓.
00002100	15	5D	CA		F2	OF	F5	58	58	D4	4F	96	C5		94	BF	.]Êéò.ÕXXÔO-Å0″¿
00002110	AE	48	4A	8E	50	7C		60	ЗE	B0	8D	38	94	4B	DA	CD	⊗HJŽP Ö`>°.8″KÚÍ
00002120	9F	74	A5	28	FE	Bl	14	DB	E3	D9	ЗA	A8	Cl	34	BF	A9	Ÿt¥(þ±.ÛãÙ: Å4¿©
00002130	12	DC	57	C7	95	8E	_	8A	E5	9D	7C	63	CA	9F	DE	D6	.ÜWÇ•ŽåŠå. cÊŸÞÖ
00002140	66	21	09	38	Cl	CF		66			64			_	0A		f!.8ÁÏÊf.šd°>Q
00002150	6D	40	2E	F7	BB	03	OA		40	A6	DF	33	9B		C6	57	m@.÷»@¦ß3>`ÆW
00002160	58	0D	F9	BF	F5	4B	94	4B	F6	64	58	02	9B	52	EA		X.ù¿ÕK″KödX.>RêÍ
00002170	22	59	40	F2	22		AO	OF	F2	B3	B3	B5	Al		67	BB	"Y@ò"ò³'µ;&g»
00002180	11	DE	DO		D2		FC		2D	66	77	30	C4	_	5F	A8	.ÞÐSÒ-üî-fwOÄ
00002190	ED	DB	84		5E	75		08		E4	15	50	B5	79	AA		íÛ"i^u:.šä.Pµyªå
000021A0	91	0E	AE	54	D3	4E	OF	DE	1E	19	91	3E	4C	C7	E8	30	`.⊗TÓN.Þ`>LÇèO
000021B0	BA	86	60	СВ	68	03	DO	F4	6F	B8	E3	29	EF	7D	ЗF	4D	°†`Ëh.Đôo,ã)ï}?M
000021C0	98	8D	D6	_	DC	15	FB	0C	22	8E	CO	DE	27	80	32	ЗA	~.Ö%Ü.û."ŽÀÞ'€2:
000021D0	F9	7A	1A		3C	47	B8	54	80	A9	87	8F	DA		39	84	ùz.à <g,t€©‡.úè9"< td=""></g,t€©‡.úè9"<>
000021E0	5A	11		65	19	97	90	OF	24	C5	38	FC	EE	_	98	36	Z.~e.—\$Å8üîÏ~6
000021F0	B9	FO		BE			16			47			_	9B		8A	⁺ðq¾.d`Gè⁺v>ÜŠ
00002200	00	00	00	00	00	00	00	00	00	00	00	00	24	00	00	20	\$
00002210	00	00	00	00	00	00											

Figure 80

Whether the file size is between 1MB and 5MB and the extension is not in the targeted lists, the ransomware only encrypts the first MB of the file, and the encrypted file has the following structure:



📓 test.txt.PS	FUX																
Offset (h	) 00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	OF	
00100000	41	41	41	41	41	41	41	41	41	41	41	41	41	41	41	41	ААААААААААААААААА
00100010	F5	39	A3	4F	E1	25	32	0E	33	BF	FD	C9	ЗE	FA	C9	09	õ9£0á%2.3¿ýÉ>úÉ.
00100020	AF	FE	BB	C6	A7	34	5E	7D	85	F7	3C	10	80	49	D8	39	[p»Æ§4^}÷<.€IØ9
00100030	99	C8	E1	01	4A	E5	95	B4	A5	EB	50	6A	<b>B</b> 5	32	79	2B	™Èá.Jå•′¥ëPjµ2y+
00100040	3A	FB	64	73	28	0A	7D	15	FD	92	EE	C9	<b>B</b> 3	EB	B2	67	:ûds(.}.ý'îÉ'ë*g
00100050	99	8B	15	77	A7	DB	94	DE	9D	56	7B	D3	9E	89	AE	19	™<.w§Û″Þ.V{Óž‰©.
00100060	69	FF	02	49	38	41	DO	DF	DE	28	20	98	A6	30	5E	12	iÿ.I8AĐBÞ( ~¦0^.
00100070	1B	05	C3	8E	FA	0D	06	91	FE	2D	58	AA	E2	80	74	2C	ÃŽú`þ-Xªâ€t,
00100080	Al	2A	F5	7D	48	96	0A	60	81	40	AA	40	0C	8C	37	06	;*ő}H−.`.@ª@.Œ7.
00100090	5C	10	1E	76	05	43	_	2A	A9	lD	4A	27	A6	CF	E9	DB	\v.Cï*©.J'¦ÏéÛ
001000A0	74	3F	FE	8E	19	62	4E	58	06	0E	8B	31	C5	ЗB	DC	03	t?þŽ.bNX <lå;ü.< td=""></lå;ü.<>
001000B0	A4	5F	<b>B</b> 8	B8	E3	28	19	83	C2	<b>B</b> 3	B3	89	FD	DO	E5	BC	¤_,,ã(.f³³‰ýĐåų
001000C0	7C	FA	6B	9E	CE	2E	17	7E	82	4A	9F	00	57	06	31	BC	úkžÎ~,JŸ.W.14
001000D0	F8	89	77	F4	EB	90	8C	OF	2F	71	99	37	B8	91	BD		ø‱wôë.Œ./q™7, ખ≾.
001000E0	53	4D	DE	9C	5A	DB	81	0B	B8	ЗF	E2	0C		C2	0E	E8	SMÞœZÛ,?â. Â.è
001000F0	32	B9	78	E5	3F		52	E4	74	02	1F	B3		6C		FD	2 <b>°xå?</b> *Rät"1Cý
00100100	70	FD	DF	EB	04	2E	B3	08	C7	13	02	1D	23	46	51		рýßё°.Ç#FQ.
00100110	31	1B	89		48	03	E5	00	50						E8		l.‰4H.å.P/.ß Aè
00100120	73		2A		99	67		B1	68			EC	73		AD		s:*8™gl±h.ÿìsî./
00100130		Al		7B	71	28	_	52			ЗA				Α9	01	Ü;0{q(×R2.:;,.©.
00100140		46	05	25	2D	_	B6	_		89			29	26	36	7A	ŸF.%-è¶©+‰ÞÛ)&6z
00100150		51	-	87	65	E7	19			FO		1C				60	.QLJeç.ő.ðÄ.â.¦`
00100160		F3		89			CA				D9		9E	B4	48	72	.ó ‰°íÊ ≫≾0ÙŐž′Hr
00100170	08	91		EC	FO		5C		_		11	F9	36		A3	39	.'.ìð6\'Ö<.ù6.£9
00100180		74			7A		DA			7E		97		63	CF	2E	.tn zPÚ.k~åcÏ.
00100190	B4	7A	A0	A2	56		A6			64	9C	9E	El	28		E3	′z ¢V%≤¦dœžá(jã
001001A0		4E	16	DF	46	C3	93			27	C7	F4	21	79		95	°N.BFÃ"ÅL'Çô!y.•
001001B0	46	F8	98	4C	B7	43	E5	53	ЗD	86	84			32	8E	82	Fø~L·CåS=†"Zö2Ž,
001001C0	23	09	68	ED	DF	41	1B	4B		B1	DC	5A		FD	04	70	#.hißA.K±ÜZÆý.p
001001D0	AC	8A	72	52	57	FE	59	57	86	62	C1	63	C8	2C	17	7D	¬ŠrRWþYW†bÁcÈ,.}
001001E0	4E	38	0C	48	26	6B	AD	ЗD	FC	C6	B4	A9	B1	90		DD	N8.H&k.=üÆ'©±Ý
001001F0		Dl	E8	Fl	DE	88		E4	70	DE			F5	39	2A		/ÑèñÞ^ÿäpÞ6<õ9*.
00100200	51	5E	07	53	38	51	81	2B	7F				DC	_	89		Q^.58Q.+."ô¤Ü ‰[
00100210	00	00	00	00	00	00	00	00	00	00	00	00	26	00	10	00	· · · · · · · · · · · · · · · · · · ·
00100220	10	00	00	00	00	00											

Figure 81

Whether the file size is between 1MB and 5MB and the extension is in the targeted lists, the ransomware encrypts the entire content, and the encrypted file has the following structure:

📓 test.sql.PSFl	JX																
Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	OF	
00100000	AA	E6	32	1D	07	8C	04	D8	3C	BA	07	A4	41	DE	CA	EC	*æ2Œ.Ø<°.¤AÞÊì
00100010	D8	A9	53	62	CO	57	03	83	F1	83	AE	FE	40	22	B8	E6	Ø©SbÀW.fñf®þ@",æ
00100020	DC	C6	D4	A9	B3	<b>B</b> 3	F1	A2	8F	07	ЗA	1B	F8	75	80	65	ÜÆÔ©''ñ¢:.øu€e
00100030	35	B3	82	17	95	2A	A7	34	08	BA	D8	A3	5A	8F	DE	C5	53,*\$4.°Ø£Z.ÞÅ
00100040	22	D4	50	60	25	4E	AC	05	AA	75	52	0A	3F	4D	5D	9B	"ÔP`%N¬. *uR.?M] >
00100050	4C	BE	80	23	14	E0	AA	8E	9D	68	94	21	73	35	19	DD	L¾€#.à°Ž.h″!s5.Ý
00100060	59	CC	49	26	B8	6B	8F	FE	1C	F6	C4	26	7F	4C	B3	76	YÌI&,k.þ.öÄ&.L'v
00100070	B6	D3	<b>B</b> 5	41	C4	4D	7D	80	35	56	45	CE	3C	68	2B	59	¶ÓµAÄM}€5VEÎ <h+y< td=""></h+y<>
00100080	C0	90	D8	A3	CD	81	0A	B7	ED	EB	71	A2	94	24	51	A6	À.Ø£Í iëq¢"\$Q;
00100090	BE	94	A0	20	97	DE	87	F3	8F	99	B3	8E	Al	78	B2	0E	¾″ —₽‡ó.™³Ž;xº.
001000A0	80	ED	E8	12			6D					11			F2		€íè.Ôwmfå.Ð]òŸ
001000B0	4C	5C		36						23			E3	74	30	09	L\û6äµE.ª#ôRãt0.
001000C0	4A	BD	6D	24	DB	DF	9F	A9	B7	AD	D9	A7	92	08	DF	8D	J⊁sm\$Ûߟ©∙.Ù§′.ß.
001000D0	92	DD		11									1E	22		87	′Ý*.fc°.′ýÌC."Ї
001000E0	61	C9	95	ЗA	8A	E5	3B							48	4E	60	aÉ•:Šå;%úãèHN`
001000F0	EB	BD	B2	EF	ЗD	95	0C	58		4D			CO	28	45	A5	뉰ï=•.X.MBIÀ(E¥
00100100		10	_			57	9A	D3	6E	42	DC	DA	1D		9F		Ò. 7áWšÓnBÜÚŸk
00100110		56				AA	B2			63			F7	14	56	CF	ØV¤< 5ª * ′4c8D÷.VÏ
00100120		45				FO	39		_		_	28			D3		TE2œ©ð9{Qó}(1+Ó6
00100130	12	F6	42	28		32		89		FB			9E	64	C7	E4	.öB(ð2p‰Wû õždÇä
00100140	0B	BD	53	7F	56	F6	DE	E3		B4	88			A9	33	D7	.∺S.VöÞã"´^.Ÿ©3×
00100150		82		1A				01	FD					5F	7E		KÍ).ý"ùÙ~_~×
00100160		E2											B0			FA	.âfI༔י°5̰iMú
00100170		91					72						91		0C		Ã`.Æå÷rû7≫§¼`"
00100180			_	64			05				_		9F				#d{:;−笟lœ
00100190		98		61			E7						9D				,~úa°±çšH×K©'
001001A0				6B			92					Bl		56		2E	ÝTMk×.'Ôàê&±£VÀ.
001001B0	81	28	E2	FE	43	30	17	64	DE	87	FB		30	_	18	37	.(âþC0.dÞ‡û.0ã.7
001001C0	35	00	7C		C8					A7			35			09	5.  "E w^3§"_5,
001001D0	29	FD	CD		08					15			AA			37	)ýÍR.£Ì•`.â.ªh.7
001001E0	04		D2		F8	FF			CD			B2	CC			34	ÒwøÿÍÞ."Ì3Ê4
001001F0		31		0A			AD			61		75	70			95	.l°.sµ.¥.afupU‡•
00100200		1A			2C	80	17	B3					DA		_		¬.š†,€.'FÓ"″Ú.,»
00100210	00	00	00	00	00	00	00	00	00	00	00	00	24	00	10	00	\$
00100220	10	00	00	00	00	00											

Figure 82



Whether the file size is greater than 5MB and the extension is not in the targeted lists, the ransomware encrypts 5 chunks of (file size/100 \* 10) bytes. In this case, this value is (0x500010/0x64 \* 0xa) = 0x7FFF8 bytes (basically, the malware encrypts 0x7FFF8 bytes, then skips some bytes, and then encrypts 0x7FFF8 bytes again and so on). The structure of the encrypted file is presented below:

📓 test.txt.PSFU	JX																
Offset(h)	00	01	02	03	04	05	06	07	08	09	ΟA	0B	0C	OD	0E	OF	
0007FF50	20	27	38	EO	A3	15	77	A8	D4	Dl	E9	AD	8F	C2	BD	63	'8à£.w"ÔÑé¾c
0007FF60	58	DB	94	0C	99	26	47	D8	F5	96	74	84	11	02	70	D3	XÛ″.™&GØõ-t"pÓ
0007FF70	13	19	5B	15	FC	BF	17	A5	3B	1F	F9	F1	0C	F2	79	63	[.ü¿.¥;.ùñ.òyc
0007FF80	D9	15	CD	F6	30	66	68	BE	2E	BF	A9	D3	1B	27	4A	96	Ù.Íö0fh¾.¿©Ó.'J-
0007FF90	38	EA	D9	1C	68	2F	2D	37	85	47	E6	F5	24	62	48	65	8êÙ.h/-7Gæõ\$bHe
0007FFA0	00	27	8E	6C	45	88	C2	26	36	35	1E	45	EA	AF	BC	0E	.'Ž1E^Â&65.Eê 4.
0007FFB0	22	45	B1	27	CE	E4	DD	12	38	13	1A	7B	23	63	6A	6D	"E±'ÎäÝ.8{#cjm
0007FFC0	F6	12	84	24	92	27	63	E6	76	5A	68	84	79	C3	FA	D2	ö."\$''cævZh"yÃúÒ
0007FFD0	19	6B	Dl	4D	8D	FC	C1	43	F7	C2	DO	7F	9C	3F	5E	49	.kÑM.üÁC÷ÂĐ.œ?^I
0007FFE0	08	5A	88	B5	F2	08	79	06	34	7A	7E	6D	C7	26	64	43	.Z^µò.y.4z~mÇ&dC
0007FFF0	81	3D	F8	92	6A	75	7C	EA	41	41	41	41	41	41	41	41	.=ø′ju êAAAAAAA
00080000	41	41	41	41	41	41	41	41	41	41	41	41	41	41	41	41	АААААААААААААААА

Figure 83

📓 test.txt.PSFU	JX																
Offset(h)	00	01	02	03	04	05	06	07	08	09	OA	0B	0C	OD	0E	0F	
004FFFE0	41	41	41	41	41	41	41	41	41	41	41	41	41	41	41	41	ААААААААААААААААА
004FFFF0	41	41	41	41	41	41	41	41	41	41	41	41	41	41	41	41	ААААААААААААААА
00500000	41	41	41	41	41	41	41	41	41	41	41	41	41	41	41	41	ААААААААААААААА
00500010	95	F5	46	25	C5	B8	F7	BA	C7	CA	6A	94	09	0B	BO	62	•õF%Å,÷°ÇÊj″°b
00500020	2F	28	EE	52	17	3C	63	FA	Cl	9C	46	F8	ED	B2	92	5B	/(îR. <cúáœføi"′[< td=""></cúáœføi"′[<>
00500030	76	C4	91	7A	79	00	79	C5	50	22	99	5F	D5	16	C2	69	vÄ`zy.yÅP"™ Õ.Âi
00500040	17	90	20	12	4A	72	10	E4	DF	BD	35	3F	CO	DA	32	3B	Jr.äß*≤5?ÀÚ2;
00500050	06	06	DF	BO	B0	E6	F9	40	50	56	4D	55	17	B9	69	2B	₿°°æù@PVMU.¹i+
00500060	05	00	79	21	65	Fl	5C	95	51	8E	CA	C2	C2	D3	75	C3	y!eñ\•QŽÊÂÂÓuÃ
00500070	48	33	C3	5D	14	21	4D	46	8F	AF	B8	FO	83	83	40	25	H3Ã].!MF., ðff@%
00500080	В9	01	22	29	CA	D9	DD	C3	F9	F5	2C	07	11	0D	9C	B5	'.")ÊÙYÃùõ,œµ
00500090	F4	BB	61	12	DD	03	Dl	D6	FE	28	94	E5	8C	21	25	74	ô»a.Ý.ÑÖþ("åŒ!%t
005000A0	8B	1F	60	95	0A	38	DC	8C	02	CE	07	7C	CB	D4	CA	7C	<.`•.8ÜŒ.Î. ËÔÊ
005000B0	C2	BE	24	18	32	87	40	B5	43	A9	EE	41	37	42	89	AO	¾\$.2‡0µC©îA7B‰
005000C0	E1	03	13	CC	85	92	46	12	68	DA	41	30	4D	F8	98	B7	á. Ì' F. hÚA0Mø~
005000D0	95	79	1F	26	F3	06	22	E5	E7	7A	E5	C9	44	7B	DC	6E	•y.&ó."åçzåÉD{Ün
005000E0	B8	EC	F7	8D	1B	81	D5	47	CE	A6	Fl	6D	A9	<b>B</b> 6	C5	0E	,ì÷ÕGΦñm©¶Å.
005000F0	9F	CF	E3	B7	19	4C	F8	52	35	60	36	25	99	D9	4A	16	ŸÏã∙.LøR5`6%™ÙJ.
00500100	93	5D	96	14	4F	2F	8A	A0	32	F3	9C	1B	CO	C8	D7	41	"]0/Š 2óœ.ÀÈ×A
00500110	26	61	D9	14	6D	34	9D	36	96	22	D7	06	87	46	78	68	&aÙ.m4.6-"×.‡Fxh
00500120	F2	51	E5	28	06	AB	EF	43	D3	24	39	BA	E0		BB	55	òQå(.«ïCÓ\$9°à‹»U
00500130	C7	6D	F3	0D	AF	E2	1C			4A	C1	A9	22	2F	50	89	Çmó. â.T.JÁ©"/P‰
00500140	1A	31	14	CE	EF	C9	1E	0D	29	D4	40	07	43	Fl	F5	FB	.1.ÎïÉ)Ô@.Cñõû
00500150	37	B3	80	03	E9	C0	DE	59	C7	2E	95	F2	53	84	10	26	7°€.éÀÞYÇ.•òS".&
00500160	44	1E	61	77	58	D3	CC	2F	0E	CE	87	06	A5	20	64	2E	D.awXÓÌ/.·.¥ d.
00500170	D7	D4	83	04	DA	CO	24	77			48	Dl				34	×Ôf.ÚÀ\$w⊗.HÑ4
00500180	7B	5C	2F	CF	E9	D4	CB	6D	74	F6	96	74	03	53	B3	38	{\/ÏéÔËmtö-t.S³8
00500190	E7	D2	8A	08	FB	10	6F	C6	0C	EB	18	CE	60	A7	B5	FD	çÒŠ.û.oÆ.ë.Î`§µý
005001A0	CE	87		E9	C7		52				1F	8D	5D	FB	FA	B5	·véÇ´RMûŸ]ûúµ
005001B0	8B	EF	D4	65	0A	52	4E	C7	09	90	A7	87	OF	5D	17	78	<ïÔe.RNǧ‡.].x
005001C0	A6	C8	57	F6	E7	A2	9A	1B	22	F8	FO	OE	8D	22	55	48	¦ÈWö碚."øð"UH
005001D0	83	80	4D		EB	17		_	8E	F9		43	92	FO		16	f€M©ë.?(Žù-C'ð
005001E0	DE	95			DA	_	81		C4			17	8F	7D	72	8C	Þ•v¦ÚÎ.ÚÄ7}}rŒ
005001F0	OD	49	El	19	88					32						34	.Iá.^"À2Uöò.t4
00500200	8C		D9		6D					Dl		16			9E		Œ¤Ùžm.Ú<`Ñù.R.ž™
00500210	00	00	00	00	00		00	00	00	00	00	00	25	32	10	00	
00500220	50	00	00	00	00	00											P

Figure 84

Whether the file size is greater than 5MB and the extension is in the targeted lists, the ransomware encrypts the entire content, and the encrypted file has the following structure:



📓 test.sql.PSF	UX																
Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	OF	
004FFFE0	67	4F	91	00	8E	DA	9D	96	D3	86	BC	1D	E9	B5	74	FC	gOʻ.ŽÚÓ†4.éµtü
004FFFF0	51	CO	98	6B	07	7D	2A	7C	98	CA	20	Cl	E9	9E	79	49	QÀ~k.}* ~Ê ÁéžyI
00500000	10	0C	DO	4C	F2	05	10	59	D6	D9	70	F4	02	88	A7	EB	ĐLòYÖÙpô.^§ë
00500010	B1	F9	17	Dl	39	14	BB	2F	A7	B2	13	8F	D9	3D	56	8C	±ù.Ñ9.»/§*Ù=VŒ
00500020	26	40	2A	E7	04	BD	39	CO	47	6E	BD	4D	4E	DC	86	80	&@*ç.¥9ÀGn⊁MN܆€
00500030	DC	7A	4D	BF	CO	E2	33	DD	B4	18	92	ЗD	C4	2B	81	FB	ÜzM¿Àâ3Ý′.′=Ä+.û
00500040	73	22	5C	EC	0C	E3	7E	94	6A	4E	EB	AA	95	AO	4D	OF	s"\ì.ã~"jN법 M.
00500050	0D	5B	B7	OD	18	4B	Cl	E3	45	17	0C	10	EE	2C	18	4C	.[ <sup>.</sup> KÁãEî,.L
00500060	CA	52	FD	63	EE	98	5F	3D	AE	16	B4	C7	F3	A3	46	7 F	ÊRýcî~ =0. Çó£F.
00500070	84	43	B9	AA	BA	22	08	79	10	57	Al	54	10	AB	60	EO	"Claow.y.W;T.«`à
00500080	92	3C	6E	1E	DC	17	18	F3	9B	32	B7	4B	3F	60	F8	BA	′ <n.üó>2 ⋅K?`ø°</n.üó>
00500090	B5	66	35	ЗA	F9	83	Dl	E6	48	51	Al	E2	CF	60	D8	8C	µf5:ùfÑæHQ;âÏ`ØŒ
005000A0	0D	D6	70	60	77	15	E4	05	DD	5B	F3	F8	46	EB	96	F8	.Öp`w.ä.Ý[óøFë-ø
005000B0	EA	A5	FA	CA	A2	8F	73	AB	9C	04	65	C9	00	СВ	7F	2C	ê¥úÊ¢.s≪œ.eÉ.Ë.,
005000C0	A0	FO	AA	35	FB	F3	ЗF	7E	B8	C3	AE	A5	4D	D9	Fl	08	ð°5ûó?~,î¥MÙñ.
005000D0	45	B2	13	DB	A4	88	2C	FO	02	ЗA	08	59	AA	22	<b>B</b> 5	58	E <sup>2</sup> .Û¤ <sup>^</sup> ,ð.:.Y <sup>ª</sup> "µX
005000E0	6C	72	92	F7	AC	5C	42	D2	2F	8A	87	81	61	58	CE	B7	lr'÷¬\BÒ/Ї.aXÎ ∙
005000F0	12	38	62	Fl	7D	B6	DA	46	48	5F	64	DD	DB	71	76	98	.8bñ}¶ÚFH_dÝÛqv~
00500100	79	6A	D3	D8	2E	8A	76	99	FO	D9	79	79	2D	F3	2B	19	yjÓØ.Šv™ðṺyy−ó+.
00500110	FA	12	3C	F2	73	OD	B6	40	82	B1	44	48	EF	74	41	86	ú.<òs.¶0,±DHïtA†
00500120	E 9	ΕO	87	74	CO	2C	32	F7	85	03	AD	СВ	F6	C5	3F	5B	éà‡tÀ,2÷ËöÅ?[
00500130	1F	6C	E8	4A	69	74	45	A5	6C	66	8F	7A	CB	EC	6B	82	.lèJitE¥lf.zËìk,
00500140	E9	8E	04	F3	A2	05	32	EC	1C	15	64	D8	D2	AO	8B	C7	éŽ.ó¢.2ìdØÒ ∢Ç
00500150	A3	61	9E	88	75	27	42	E2	4F	B1	E9	73	31	0A	85	B8	£až^u'BâO±ésl,
00500160	EE	8F	8A	3B	DB	21	17	BD	BF	2A	CC	20	9E	FE	21	BB	î.Š;Û!.≒¿*Ì žþ!»
00500170	C9	02	0E	7B	62	C8	97	27	69	E8	Dl	20	F3	08	FO	E0	É{bÈ—'ièÑ ó.ðà
00500180	59	06	81	4E	05	03	B4	40	8F	16	4D	31	C7	20	7B	CF	YN '@MlÇ {Ï
00500190	D3	01	0A	E9	14	58	35	5F	72	73	EB	75	18	79	96	56	Óé.X5_rsëu.y-V
005001A0	F9	48	EA	99	5F	32	69	62	9F	8C	A2	ED	A8	1C	63	2C	ùHê™_2ibŸŒ¢í¨.c,
005001B0	45	CD	88	<b>A</b> 6	29	17	88	C3	64	74	9A	F6	BF	FO	80	11	EÍ^¦).^Ãdtšö¿ð€.
005001C0	69	34	12	B0	EB	C7	45	6E	6A	F3	6E	10	C9	87	9D	72	i4.°ëÇEnjón.ɇ.r
005001D0	57	F8	E7	FE	E0		38	66	94	_	9D		28			DO	Wøçþà68f″ç.š(Ú^Đ
005001E0	35		13		A4	42	0E	08	47	86	7C	22	_	D8	DA		5".¿¤BG† "\$ØÚ
005001F0	CE				AO			FE	29	2E			CA			ЗD	Î-<þ ±ôþ).tþÊ.e=
00500200					09											00	Ó;€ÝŒ⊣©X.jYJ¢.
00500210	00	00	00	00	00		00	00	00	00	00	00	24	00	10	00	\$
00500220	50	00	00	00	00	00											P

*Figure 85* When the malware runs with the "-log" parameter, then the list of actions is logged in a file:

🔚 test.	log 🗵	
1	[08:19:53]	Found 2 drives:
2	[08:19:53]	C:\
3	[08:19:53]	D:\
4	[08:19:58]	Can't get file size C:\\$WINRE_BACKUP_PARTITION.MARKER. GetLastError = 0
5	[08:19:58]	Can't open file C:\bootmgr. GetLastError = 5
6	[08:20:02]	File C:\Program Files\int is already open by another program.

Figure 86



### **APPENDIX**

Lists of targeted extensions:

- .4dd .4dl .accdb .accdc .accdc .accdr .accdt .accft .adb .ade .adf .adp .arc .ora .alf .ask .btr .bdf .cat .cdb .ckp .cma .cpd .dacpac .dad .dadiagrams .daschema .db .db-shm .db-wal .db3 .dbc .dbf .dbs .dbt .dbv .dbx .dcb .dct .dcx .ddl .dlis .dp1 .dqy .dsk .dsn .dtsx .dxl .eco .ecx .edb .epim .exb .fcd .fdb .fic .fmp .fmp12 .fmpsl .fol .fp3 .fp4 .fp5 .fp7 .fpt .frm .gdb .grdb .gwi .hdb .his .ib .idb .ihx .itdb .itw .jet .jtx .kdb .kexi .kexic .kexis .lgc .lwx .maf .maq .mar .mas .mav .mdb .mdf .mpd .mrg .mud .mwb .myd .ndf .nnt .nrmlib .ns2 .ns3 .ns4 .nsf .nv .nv2 .nwdb .nyf .odb .oqy .orx .owc .p96 .p97 .pan .pdb .pdm .pnz .qry .qvd .rbf .rctd .rod .rodx .rpd .rsd .sas7bdat .sbf .scx .sdb .sdc .sdf .sis .spq .sql .sqlite .sqlite3 .sqlitedb .te .temx .tmd .tps .trc .trm .udb .udl .usr .v12 .vis .vpd .vvv .wdb .wmdb .wrk .xdb .xld .xmlff .abcddb .abs .abx .accdw .adn .db2 .fm5 .hjt .icg .icr .kdb .lut .maw .mdn .mdt
- .vdi .vhd .vmdk .pvm .vmem .vmsn .vmsd .nvram .vmx .raw .qcow2 .subvol .bin .vsv .avhd .vmrs .vhdx .avdx .vmcx .iso

