

A Deep Dive into The Grief Ransomware's Capabilities

Prepared by: Date: LIFARS, LLC 12/30/2021

EXECUTIVE SUMMARY

Grief ransomware is the successor of the DoppelPaymer ransomware, which emerged from the BitPaymer ransomware. Grief is deployed in an environment already compromised by Dridex and where the threat actor performed post-exploitation activities using Cobalt Strike. The ransomware is obfuscated and employs anti-analysis techniques that include API hashing, Vectored Exception Handling (VEH) manipulation, the Heaven's Gate technique, encrypt relevant data using RC4. Grief runs with specific parameters computed based on the victim's environment and crashes if no/incorrect parameters are provided (if you have been a victim of Grief ransomware, please contact us). The malware deletes all Volume Shadow Copies using vssadmin and Diskshadow and disables Microsoft Defender Antivirus. The encrypted files have the ".payOrgrief" extension, and the malware imports an RSA public key that will be used to encrypt the generated AES file encryption keys.



ANALYSIS AND FINDINGS

SHA256: 2d1d08fce7156053c017825b722968b3117c9230412f4e7da5f89699ec9913cd

The DLL file is one of the most challenging malware samples we've even analyzed because of the multiple layers of obfuscation, API hashing, Vectored Exception Handling, and relevant strings decrypted at runtime using RC4. We will sequentially explain how we've overcome every obstacle and what challenges remain.

The binary has only one export function called "RoonlpvfdRoomvlof":

File Help								
ê 🖬 🗡 🗎 💡								
c:\users'	ordinal (1)	name (1)	location	duplicated (0)	anonymous (0)	gap (0)	forwarded (1)	
dul indicators (wait)	1	RoonlpvfdRoomvlof	kernel32.Sleep	12	2	-	×	
···• virustotal (network error)				_				
 dos-stub (356 bytes) 	-							
🗆 file-header (Sep.2021)								
🗆 optional-header (GUI)								
🗆 directories (6)								
🗆 sections (1/5)								
 libraries (5/14) 								
	-							
tls-callbacks (n/a)								
abc strings (wait)								
🗐 manifest (n/a)								
certificate (n/a)								
overlay (wait)								

Figure 1

The malware retrieves the path of the executable file of the current process (which in our case is rundll32.exe) using the GetModuleFileNameW API:

edx= <kernel .text:6A982</kernel 	<pre>6A9829 6A9</pre>	84 C7 4 88 C7 0 C1 88 1 C7 89 4 C8 89 4 CF FF D eF1]eNameW>	6 08 04 01 (6 00 00 00 00 (5 <u>2C 80 98 (</u> 4 24 34 C 24 30 2 (76A74F90)	00 00 00 5A	mov dword ptr mov dword ptr mov dword ptr mov edx,dword mov dword ptr mov dword ptr cal edx	ds:[esi+ ds:[esi] ptr ds:[ss:[esp+	s],104 ,0 <&GetModuleFi 34 <mark>1</mark> ,eax	leNamew>]	edx: edx: edx:	:Ge ∨ > Defa 3:	75tatusword 0000 75w_B 0 x875w_C3 0 x875w 75w_C1 0 x875w_C 0 x875w 75w_SF 0 x875w_P 0 x875w 75w_SF 0 x875w_P 0 x875w aut (stdal) [esp] 00000000 [esp] 00000000 [esp+3] 00000104 [esp+4] 00050074	ES 0
Dump 1	Dump 2	Ump 3	Ump 4	Dump 5	🛞 Watch 1	[x=] Locals	2 Struct		009AF0F0 009AF0F4			
Address He	Y				ASCTT	1			009AF0F8	00000104	4	

Figure 2

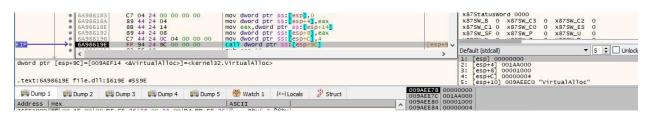
The process gets a module handle for a module called "self.exe":

• 6A982ABE	89 44 24 1C	mov dword ptr ss:[esp+1C],eax	[esp+1	
GA982AC2			ecx:Gt v	Default (stdcall) 1: [esp] GAB2E92A L"self.exe"
ecx= <kernel32.getmodulehand< td=""><td></td><td></td><td></td><td>1: [esp+4] 009AF14E L"C:\\Windows\\SysWOW64\\rur 3: [esp+8] 0000104 4: [esp+C]_00650074</td></kernel32.getmodulehand<>				1: [esp+4] 009AF14E L"C:\\Windows\\SysWOW64\\rur 3: [esp+8] 0000104 4: [esp+C]_00650074
		e Manual e fuita e Alena	009AF0F0 GAB	22E92A L"self.exe"

Figure 3



VirtualAlloc is utilized to allocate memory in the address space of the current process (0x1000 = **MEM_COMMIT**, 0x4 = **PAGE_READWRITE**):





The binary writes a new executable to the newly created memory area and transfers the execution flow to a function inside it. The LoadLibraryA routine is used to load multiple DLLs into the address space of the process:

02F91B45 02F91B48 02F91B48 02F91B4B	89 14 24 88 4D E4 89 45 88	<pre>mov dword ptr ss:[esp].edx mov ecx,dword ptr ss:[ebp-1C] mov dword ptr ss:[ebp-48].eax</pre>		x875W_C1 0 x875W_C0 0 x875W_E5 0 x875W_SF 0 x875W_P 0 x875W_U 0
	FF D1	call ecx	>	Default (stdcall)
ecx= <kernel32.loadlibrarya< th=""><th>▷ (76A75980)</th><th></th><th></th><th>2: [esp+4] 02F93020 L"ntdll.dll" 3: [esp+8] 048800E8 "PE" 4: [esp+C] 6A580000</th></kernel32.loadlibrarya<>	▷ (76A75980)			2: [esp+4] 02F93020 L"ntdll.dll" 3: [esp+8] 048800E8 "PE" 4: [esp+C] 6A580000

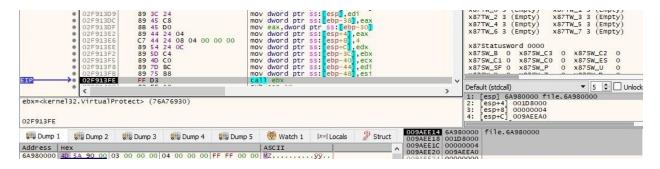
Figure 5

The GetProcAddress API is utilized to retrieve the address of export functions from multiple DLLs:

	 02F91B9C 02F91B9F 02F91BA3 	89 04 89 4C 8B 4D	24 04	mov	dword ptr dword ptr ecx,dword	ss:[esp+4]	,ecx		xa	875W_C1 0 x875 875W_SF 0 x875	5W_P 0 x87	SW_ES 0 SW_U 0
ecx= <kernel< th=""><th>O2F91BA6 < I32.GetProcAd</th><th>FF D1</th><th></th><th></th><th>l ecx</th><th></th><th></th><th>></th><th>11000000</th><th>ault (stdcall) [esp] 76FE000 [esp+4] 048D0 [esp+8] 048CB [esp+C] 048D0</th><th>0 ntdll.76FE ODA "RtlComp 014</th><th></th></kernel<>	O2F91BA6 < I32.GetProcAd	FF D1			l ecx			>	11000000	ault (stdcall) [esp] 76FE000 [esp+4] 048D0 [esp+8] 048CB [esp+C] 048D0	0 ntdll.76FE ODA "RtlComp 014	
Dump 1	Dump 2	Dump 3	Dump 4	Dump 5	🛞 Watch 1	[x=] Locals	Struct			0 ntdll.76FE00 A "RtlComputed		

Figure 6

The process changes the protection of the memory area where the malicious DLL resides by calling the VirtualProtect routine ($0x4 = PAGE_READWRITE$):







The original DLL's code is modified, and a different DLL file appears in place of it. After the modifications are done, the memory protection is changed again ($0x2 = PAGE_READONLY$):

Address He	N.				ASCII			009AEE1C 000 009AEE20 000		
Dump 1	Ump 2	Dump 3	Dump 4	Dump 5	🛞 Watch 1	[x=] Locals	Struct	009AEE18 000		
IP edx= <kernel 02F91497</kernel 	02F9148D 02F91490 02F91490 02F91494 02F91494 02F91494 02F91494 32.VirtualP	8B 4D 89 4C 8B 55 FF D2	A8 24 OC C4	mo mo	v ecx,dword v dword ptr v edx,dword l edx	ptr ss:[ebp ss:[esp+C]	ecx	,	x875W_C1 0 x875W x875W_SF 0 x875W Default (stdcall) 1: [esp] 6A980000 2: [esp+4] 000040 3: [esp+8] 0000000 4: [esp+6] 0000000	C0 0 x875W_E5 0 P 0 x875W_U 0 ▼ 5 ↓ Unloc file.6A980000 0 2
	 02F9147A 02F9147D 02F91485 	C7 44	24 04 00 04 24 08 02 00	00 00 mo	v dword ptr v dword ptr v dword ptr	ss:[esp+4]	400		x87StatusWord 000 x87SW_B_0_x87SW	

Figure 8

The binary disables the **DLL_THREAD_ATTACH** and **DLL_THREAD_DETACH** notifications for the newly created DLL:

<u>=1P</u>	02F915D5 02F915D7 02F915DD 02F915EB		08 30 F9 02 70 FF FF FF	mo mo ca	v dword ptr v edx,dword v dword ptr edx	ptr ds: [<&	DisableThrea	dLibraryCall	V Defi	75W_C1 0 x875W_C0 75W_SF 0 x875W_P ault (stdcall)	0 x875₩_U 0
edx= <kernel: 02F915E3</kernel: 		readLibrary	Calls> (76A)	746A0)	👹 Watch 1	[x=] Locals	Struct		2: 3: 4: A98000	[esp] 6A980000 fil [esp+4] 00188C72 [esp+8] 00000002 [esp+C] 009AEEA0	e.64980000
Address He 6A980000 4D 6A980010 B8 6A980020 00 6A980030 00 6A980040 0E 6A980050 69	x 5A 90 00 03 00 00 00 00 00 00 00 00 00 00 00 00 1F BA 0E 00 73 20 70 72 20 62 65 20	00 00 00 0 00 00 00 0 00 00 00 0 00 00 0	04 00 00 00 40 00 00 00 00 00 00 00 00 00 00 00 21 B8 01 4C	FF FF 00 00 00 00 00 00 00 00 00 00 E8 00 00 00 CD 21 54 68 61 6E 6E 6F 44 4F 53 20	ASCII MZ@ @ is program		<u>a</u>	009AEE1C 0 009AEE20 0 009AEE24 0 009AEE28 0 009AEE2C 0 009AEE30 0 009AEE34 0 009AEE38 6	0188C7 000000 09AEEA 000000 48D00A 48CB01 000000 48D00C 48D00C A98000 000000	2 0 8 0 0 1 1 1 1 6 4 980000	



The final DLL represents the last stage of Grief ransomware. It has 5 export functions, however, only one is relevant in our analysis: DllRegisterServer. The other 4 exports jump in the middle of other functions, and we believe the threat actor didn't intend to use any of them:

Help								
🖬 🗡 📋 🤶								
c:\users\\desktop\rundll3	ordinal (5)	name (5)	location	duplicated (0)	anonymous (0)	gap (0)	forwarded (0)	
III indicators (wait)	5	DIIUnregisterServer	.text:6A994B40	22	2	140		
virustotal (network error) dos-stub (168 bytes)	4	DIIRegisterServer	.text:6A982618	100		1.5		
I file-header (May.2021)	3	Dilinstali	.text:6A9948A8	0.65	*		-	
optional-header (entry-po	2	DIIGetClassObject	.text:6A98B120	12.5	2	1323	-	
 directories (5) 	1	DIICanUnloadNow	.text:6A994D74	252			-	
- sections (entry-point)	-							
🗆 libraries (3)								
o tls-callbacks (n/a)								
- 🔂 resources (n/a)								
abc strings (wait)								
	-							
- Certificate (n/a)								

Figure 10



An important hint which suggests that the file is encrypted/obfuscated is the lack of imports: GetCommandLineW, IstrcpyW, CommandLineToArgvW, and RtIComputeCrc32. Grief, like its predecessor DoppelPaymer [1], is designed to run only with specific argument(s), otherwise it will crash. The ransomware extracts the arguments using the GetCommandLineW and CommandLineToArgvW APIs.

The malware computes the CRC32 checksum of the last argument, adds 0x1EC6086B to the result, and finally adds the instruction pointer address to this final value (figure 11 is almost identical to the figure presented at [1] regarding the DoppelPaymer Control Flow Obfuscation). If no arguments/incorrect arguments are provided, the ransomware crashes. This action represents an anti-sandbox technique and a drawback for malware analysis (if you're not the victim, of course):

🚺 🚄 🖼	-
	; Exported entry 4. DllRegisterServer
.text:6A982618	,
.text:6A982618	
.text:6A982618	
.text:6A982618	public DllRegisterServer
	DllRegisterServer proc near
.text:6A982618	
.text:6A982618	arg 0= dword ptr 4
	arg 4= byte ptr 8
.text:6A982618	
.text:6A982618	<pre>lea eax, [esp+arg_4]</pre>
.text:6A98261C	
.text:6A982621	push eax
.text:6A982622	call ds:lstrcpyW
.text:6A982628	call ds:GetCommandLineW
.text:6A98262E	mov edx, eax
.text:6A982630	lea eax, [esp+ <mark>0</mark>]
.text:6A982633	push eax
.text:6A982634	push edx
.text:6A982635	call ds:CommandLineToArgvW
.text:6A98263B	_
.text:6A98263D	
.text:6A982640	mov edx, 7FFFFFFh
.text:6A982645	
.text:6A982646	mov [esp+0], eax
.text:6A982649	
.text:6A98264C	
.text:6A982651	mov edx, [esp+
.text:6A982654	add eax, eax
.text:6A982656	
.text:6A982657	
.text:6A98265A	
.text:6A98265C	
.text:6A982662	
.text:6A982667	L 1 0_ 32
.text:6A98266B	
.text:6A982670	L 01,
.text:6A982674	
	DllRegisterServer endp
.text:6A982674	

Figure 11



We were able to find good insights even without the required arguments, based on the analysis of the most complex functions.

The first anti-analysis technique we present consists of inserting lots of "int 3" (0xCC) instructions in the code. This technique is like the one employed by Dridex and explained at [2]. An example of such instructions is shown in figure 12:

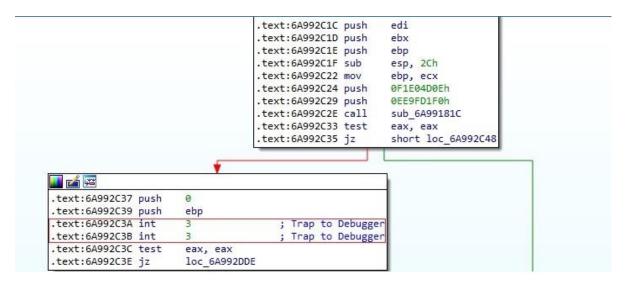


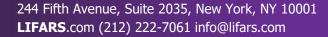
Figure 12

Grief registers a new customized Exception Handler by calling the RtlAddVectoredExceptionHandler API:

(mark)	6AA518C8 68 40 1C A5 6A push dump_patched. 6AA51C40 push 1							x875W_SF 0 x875W_P 1 x875W_U 0					
EIR	→• 6AA518CF				all ecx			ec	Default (stdca			▼ 5 💠 🗌 Unlock	
	.RtlAddVectore								3: [esp+8]	6AA51C40	dump_patche dump_patche		
Ump 1	Dump 2	Ump 3	Ump 4	💷 Dump 5	🥘 Watch 1	[x=] Locals	Struct	00D4F900 00D4F904		patched. 6A	A51C40		

Figure 13

The exception handler displayed in figure 14 expects an exception code as an argument. Whether the exception code is 0xC000005 (**ACCESS_VIOLATION**), 0xC00000FD (**STATUS_STACK_OVERFLOW**), and 0xC0000374 (Heap Corruption), the malware kills itself by calling the NtTerminateProcess API. If the exception code is 0x80000003 (**EXCEPTION_BREAKPOINT**), the function mimics the "call eax" instruction, which means that two "int 3" instructions can be interpreted as a "call eax" instruction. We've patched the binary by replacing the "0xCCCC" bytes with "0xFFD0".





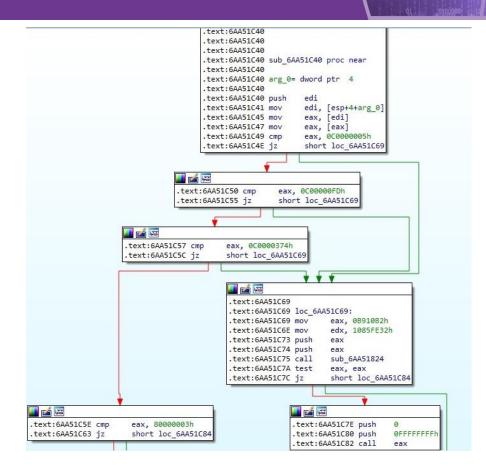


Figure 14

Grief implements API hashing in multiple functions. The first argument is the hashed DLL name, and the 2nd argument is the hashed API name:

L alasta	6AA51C69 6AA51C6 6AA51C6E 6AA51C6E 6AA51C7 6AA51C7						.085FE32						X875W_B 0 X875W_C3 0 X875W_C2 0 X875W_C1 0 X875W_C0 0 X875W_E5 0 X875W_5F 0 X875W_P 1 X875W_U 0					
dump_patche	<pre>6AA51C75 6AA51824 C75 dump_pate</pre>	1 25 60	FB FF FF		all dump_p		1824			*	1:	[esp+8]		ntd11.7	70F931C	5 🖨 🗌 Unloc		
		1	160	14.5	26		(3)			E7CC 00B								
Ump 1	🕮 Dump 2	Ump 3	Ump 4	🚛 Dump 5	👹 Watch 1	[≭=] Locals	2 5	Struct		E7CC 008 E7D0 008								

Figure 15

• 6A7C63 • 6A7C63	90 68 12 99 17 75	push D8EFD506 push shell32.75179912		x875w_SF 0 x875w_P 1 x875w_U 0
		call dump_patched, 6A7C181C	~ ~	Default (stdcall)
<pre>dump_patched.6A7C181C .text:6A7C6395 dump_pa</pre>	tched.dll:\$16395 #16395			2: [csp+4] D8EF0506 3: [csp+8] 04BD4C60 L"Microsoft Enhanced RSA ar 4: [csp+C]_00002800
Dump 1 Dump 2	💭 Dump 3 🛛 💭 Dump 4	Dump 5 👹 Watch 1 🛛 🕅 Locals 🎾 Struct	00CEF944 7517 00CEF948 D8EF	79912 shell32.Ordinal93+9E8222 D506

Figure 16



A snippet of one of the functions that parse the PEB (Process Environment Block) structure, performs XOR operations, and determines which APIs should be used, is shown below:



Figure 17

The result of the above operations, which is the address of an API, is stored in the EAX register. For example, figure 18 reveals an API that is used to kill the current process:

0 0 0 0 0	6AA51C6E 6AA51C73 6AA51C74 6AA51C75 6AA51C75 6AA51C7C 6AA51C7C 6AA51C7E 6AA51C80 6AA51C82	88 82 10 89 00 8A 32 FE 85 10 50 50 85 AA FB FF FF 85 CO 74 06 6A 00 6A FF FF D0	mov eax, 191082 mov eax, 1085F532 push eax push eax call dump_patched, 6AA51824 test eax, eax Je dump_patched, 6AA51684 push 0 push 0 p	eax eax eax eax	A0:1m_2 3 (Empty) A0:1m_3 3 (Empty) X87Tw_4 3 (Empty) X87Tw_5 3 (Empty) X87Tw_6 3 (Empty) X87Tw_7 3 (Empty) X875tatusWord 0020 X875w_6 0 X875w_6 0 X875w_6 0 X875w_6 0 X875w_6 0 X875w_6 0 X875w_6 0 0 X875w_6 0 0 X875w_6 0 0 X875w_6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
-00		11:\$11C82 #11C82		0004E7CC FFFF 0004E7D0 0000	1: [esp] FFFEFEF 2: [esp+4] 00000000 3: [esp+6] 770F931C ntdll.770F931C 4: [esp+C] 77040925 ntdll.77040925 FFFF

Figure 18

Capa [3] has been used to detect any encryption algorithms in our malicious DLL. It has identified the RC4 algorithm in sub_6A996248 based on the structure of the operations:



111111111	dista activa Ref. Exit
namesp	ce data-manipulation/encryption/rc4
author	moritz.raabe@mandiant.com
scope att&ck	function Defense Evasion::Obfuscated Files or Information [T1027]
mbc	Derense Evasion::Optiuscated Files of information [1027] Envotemenby::Encympt pata::Red [CD027]0001 _ Coveremenby::Encymption Key::Red KSA [C0028,002]
exampl	
5AF8AE	E9EB51E:0x405C42、73CE04892E5F39EC82B00C02FC04C70F:0x40646E
function:	n © 0x6A996428
an	
	ubscope:
	and: = initialize S
	charateristic: tight Toop @ 0x6A99647c
	number: 0x100 @ 0x6A996490
	and: = initialize S
	characteristic: tight loop @ 0x6A99649E
	number: 0x100 @ 0x6A99648C
	count(mnemonic(ex)cours)): 2 or more @ 0x6A99649E, 0x6A9964A3, 0x6A9964AF, 0x6A9964D9, and 6 more r:= modulo key length
	r: = madulo key length memonic: jdiw @ 0.86A996480
	ce data-manipulation/encryption/rc4
author	ce oata-manipulation/encryption/rc4 moritz-raabe@manidiant.com
	function
scope att&ck	Defense Evasion::Obfuscated Files or Information [T1027]
mbc examp]	Cryptography: Encrypt Data::Rc4 [C0027.009], cryptography: iGenerate #seudo-random Sequence::Rc4 PRGA [C0021.004] • J4040Afreg804097C6ABR60991FBI 30:0x4030B0, 3440A3F89804097.76AB86C991EBI 30:0x403E5(0), 32401AA8E3705A6256053748E59705A:0x4049F0, 73CE04892E5F39Ec82B00C02FC04c70F:0x4064C6
	5 3 44043/B300(9)//C0ABB0(B331/B1)0.0X4030B0, 3440405/B30009//C0ABB0(B331/B1)0.0X403E0, 32401ABAE3/0C0//440(3/03704045/0, 75C040525/F35C02B0(C02/C0+04045/0 M & 0X6A996428
and:	
C0	nt(characteristic(n2xor)): 1 @ 0x6A996507
01	ount(mnemonic(movzx)): 4 or more @ 0x6A99649E, 0x6A9964A3, 0x6A9964AF, 0x6A9964D9, and 6 more
co	nt(characteristic(calls from)): 4 or fewer
CO	nt(basicblock): 4 or more @ (x6A996428, 0x6A99643E, 0x6A996446, 0x6A996454, and 14 more
	ch: contain loop @ 0x6A996428
	characteristic: loop @ 0x6A996428
1000	characteristic: tight loop @ 0x6A99647c, 0x6A99649E
	ional: r:

Figure 19



Figure 20

The CryptAcquireContextW routine is utilized to acquire a handle to a key container within a CSP (cryptographic service provider). The arguments are szProvider = "Microsoft Enhanced RSA and AES Cryptographic Provider", $0x18 = PROV_RSA_AES$, and $0xF0000000 = CRYPT_VERIFYCONTEXT$:



6A7C63A2 68 00 00 00 F0 6A7C63A9 6A7C63A9 6A7C63A9 6A7C6349 6A7C634 6A7C64 6A7C64 6A7C64 6A7C64 6A7C64 6A7C64 6A7C64	push F000000 push 18 push dword ptr ss:[esp+6]	0t 000000 11 01 01 000 x875W_E 0 x875W_C2 0 x875W_C2 0 x875W_C1 0 x875W_C2 0 x875W_E5 0
eax= <advapi32.cryptacquirecontextw> (73A60380) .text:6A7C6380 dump_patched, d11:\$16380 #16380</advapi32.cryptacquirecontextw>	push o push edx call eax	x875W_SF 0 x875W_P 1 x875W_U 0 Default (stdcall) V 5 Unlock 1: [ssp+3] 0000000 3: [ssp+4] 0000000 3: [ssp+4] 0000000 3: [ssp+4] 0: ssp+4] 0: ssp+4] 0: ssp+4] 0: ssp+4] ssp+4] 0: ssp+4] ssp+4]
Hardress Hex He	ASCII 00 M.1.c.r.o.s.o.f. 00 te.n.h.a.n.c. 00 e.dR.S.A. a. 00 n.dA.E.S. c. 00 r.y.p.t.o.g.r.a. 00 p.h.1.c., P.r.o.	OCCEF938 OOCEF954 OCCEF93C 0000000 OCCEF93C 00000000 OCCEF93S0 000000000 OCCEF93S0 000000000000000000000000000000000000

Figure 21

The function identified above is utilized to decrypt relevant strings using the RC4 algorithm. The RC4 key is changing frequently and has 48 bytes. We enumerate a list of decrypted strings and their explanations according to our analysis and the OSINT.

Grief doesn't encrypt the files which contain the following strings in their name and also the files that have the following extensions:

Address	Hey	ć.															ASCII
04BD0048	73	76	73	68	6F	2A	2E	65	78	65	3B	73	63	68	72	65	svsho*.exe;schre
04BD0058	2A	2E	62	61	74	3B	56	30	31	2E	6C	6F	2A	3B	56	30	*.bat;V01.lo*;V0
04BD0068	31	2E	63	68	2A	3B	56	30	31	72	65	73	2A	2E	6A	72	1.ch*;V01res*.jr
04BD0078	73	3B	52	61	63	57	GD	69	2A	2E	73	64	66	3B	57	65	s;RacWmi*.sdf;We
04BD0088	62	2A	56	30	31	2E	64	61	74	3B	64	65	66	61	75	6C	b*V01.dat;defaul
04BD0098	74	2E	72	64	70	3B	4E	54	55	53	45	52	2E	44	41	2A	t.rdp;NTUSER.DA*
																	;*.lnk;*.ico;*.i
04BD00B8	6E	69	3B	2A	2E	6D	73	69	3B	2A	2E	63	68	6D	3B	2A	ni;*.msi;*.chm;*
04BD00C8	2E	73	79	73	3B	2A	2E	68	6C	66	3B	2A	2E	6C	6E	67	.sys;*.hlf;*.lng
048D00D8	3B	2A	2E	69	6E	66	3B	2A	2E	74	74	66	3B	2A	2E	63	;*.inf;*.ttf;*.c
04BD00E8	6D	64	3B	2A	2E	4C	4E	4B	3B	2A	2E	49	43	4F	3B	2A	md;*.LNK;*.ICO;*
04BD00F8	2E	49	4E	49	3B	2A	2E	4D	53	49	3B	2A	2E	43	48	4D	.INI;*.MSI;*.CHM
																	;*.SYS;*.HLF;*.L
04BD0118	4E	47	3B	2A	2E	49	4E	46	3B	2A	2E	54	54	46	3B	2A	NG;*.INF;*.TTF;*
04BD0128	2E	43	4D	44	00	00	00	00	00	00	00	00	00	00	00	00	.CMD

Figure 22

The ransomware doesn't encrypt the files that are located in the following directories:

Address	He	<															ASCII
04BD0048	53	79	73	74	65	6D	20	56	6F	6C	75	6D	65	20	49	6E	System Volume In
04BD0058	66	6F	72	6D	61	74	69	6F	6E	3B	24	52	45	43	59	43	formation; \$RECYC
04BD0068	4C	45	2E	42	49	4E	3B	24	52	65	63	79	63	6C	65	2E	LE.BIN; \$Recycle.
04BD0078	42	69	6E	3B	57	65	62	43	61	63	68	65	3B	43	61	63	Bin;WebCache;Cac
04BD0088	68	65	73	3B	56	69	72	74	75	61	6C	53	74	6F	72	65	hes;VirtualStore



The malware also decrypts a list of environment-variable strings, which will be used as arguments for the ExpandEnvironmentStringsA function:



Address	He	<														-1	ASCII
04BD0048	25	50	72	6F	67	72	61	GD	44	61	74	61	25	5C	4D	69	%ProgramData%\Mi
04BD0058	63	72	6F	73	6F	66	74	5C	57	69	6E	64	GF.	77	73	5C	crosoft\Windows\
04BD0068	57	45	52	5C	52	65	70	6F	72	74	51	75	65	75	65	5C	WER\ReportQueue\
04BD0078	3B	25	77	69	6E	64	69	72	25	38	25	74	65	GD	70	25	;96windir96;96temp96
04BD0088	3B	25	41	50	50	44	41	54	41	25	5C	4C	6F	63	61	6C	;%APPDATA%\Local
04BD0098	5C	56	69	72	74	75	61	6C	53	74	6F	72	65	5C	3B	25	\VirtualStore\;%
04BD00A8	48	4F	4D	45	44	52	49	56	45	25	5C	44	6F	63	75	6D	HOMEDRIVE%\Docum
04BD00B8	65	6E	74	73	20	61	6E	64	20	53	65	74	74	69	6E	67	ents and Setting
04BD00C8			41	6C	6C		55	73	65		73		41	70	70	6C	s\All Users\Appl
048D00D8	69	63	61	74		6F		20	44		74		SC.	41		70	ication Data\App
048D00E8	6C	69	63	61	74	69	6F	6E			61			5C	3B	25	lication Data\;%
04BD00F8	48	4F	4D	45	44	52	49	56	45	25	5C	55	73	65	72	73	HOMEDRIVE%\Users
04BD0108		41	6C	GC	20	55	73	65	72	73		41	70	70	6C	69	\All Users\Appli
04BD0118	63		74	69	6F	6E	20	44	61		61		41	70	70	6C	cation Data\Appl
04BD0128	a state of the second	and the second	61	and in fact, the second se	69	6F		20	44		74	61			25	53	ication Data\;%S
04BD0138			74		6D	44	72	69	76		25			6F			ystemDrive%\Docu
04BD0148			6E		73	20	61	6E	64		53	65		74		6E	ments and Settin
04BD0158		73		41	6C	6C	20	55	73		72	73		41		70	gs\All Users\App
04BD0168		69		61	74	69		6E			61			5C	41	70	lication Data\Ap
04BD0178	70	6C	69	63	61	74	69	6F	6E		44	61	74		5C	3B	plication Data\;
04BD0188			79	73	74	65	GD	44	72	69	76	65		5C		73	%SystemDrive%\Us
04BD0198	65	10 m	73	5C	41	6C	6C	20	55	73	65	72	73	5C	41	70	ers\All Users\Ap
048D01A8	70		69	63	61	74		6F	6E	20	44	61	74	1.00		41	plication Data\A
04BD01B8	70	70	6C	69	63	61	74	69	6F	6E	20	44	61	74	61	5C	pplication Data\

Figure 24

A list of services to be stopped is also decrypted using RC4 (see figure 25). These services might lock important files such as databases, and the ransomware wouldn't be able to encrypt them.

Address	He	¢															ASCII
04BD0358	38	34	39	65	65	37	32	38	62	38	GD	73	6F	6C	61	70	#49ee728b;msolap
04BD0368	24	2A	3B	6D	73	73	71	6C	24	2A	3B	64	65	33	61	33	\$*;mssq1\$*;de3a3
04BD0378	35	35	62	3B	66	34	34	35	62	31	39	33	3B	32	39	37	55b;f445b193;297
04BD0388	33	31	34	39	34	3B	39	34	63	32	39	36	31	37	3B	34	31494;94c29617;4
04BD0398	61	31	34	33	66	30	3B	34	34	32	38	62	39	62	34	3B	a143f0;4428b9b4;
04BD03A8	63	31	62	31	66	30	66	62	3B	34	64	66	39	63	32	37	c1b1f0fb;4df9c27
04BD03B8	36	3B	65	33	64	34	36	38	39	32	3B	33	34	38	39	66	6;e3d46892;3489f
04BD03C8	39	31	3B	35	31	37	38	64	64	35	39	3B	66	66	33	36	91;5178dd59;ff36
04BD03D8	34	38	39	31	38	36	64	39	30	61	36	34	39	3B	34	36	4891;6d90a649;46
04BD03E8	35	61	34	64	61	37	3B	66	31	32	65	33	64	30	39	3B	5a4da7;f12e3d09;
04BD03F8	33	37	66	63	39	31	31	61	3B	31	37	32	62	31	66	61	37fc911a;172b1fa
04BD0408	66	3B	63	64	66	39	37	61	38	62	3B	32	31	38	31	63	f;cdf97a8b;2181c
04BD0418	31	35	65	3B	33	35	39	32	37	63	30	66	3B	36	38	38	15e; 35927c0f; 688
04BD0428	33	33	64	63	31	3B	66	32	61	65	37	39	63	3B	63	61	33dc1;f2ae79c;ca
04BD0438	66	66	31	30	62	33	3B	65	66	64	64	32	37	34	3B	66	ff10b3;efdd274;f
04BD0448	61	34	39	33	65	31	33	3B	72	65	70	6F	72	74	73	65	a493e13;reportse
04BD0458	72	76	65	72	24	2A	3B	62	32	61	31	37	62	31	38	3B	rver\$*;b2a17b18;
04BD0468	32	30	36	63	63	30	62	35	3B	66	35	38	32	66	34	66	206cc0b5;f582f4f
04BD0478	33	3B	61	33	32	32	31	65	62	31	3B	36	61	39	39	35	3;a3221eb1;6a995
04BD0488	66	63	64	3B	36	37	66	39	39	38	36	65	3B	33	36	30	fcd; 67f9986e; 360
04BD0498	62	39	37	39	39	3B	33	62	39	66	31	62	33	65	3B	61	b9799;3b9f1b3e;a
04BD04A8	36	37	37	32	63	39	36	3B	35	30	35	30	31	36	34	62	6772c96;5050164b
04BD04B8	3B	37	64	30	39	63	35	36	32	3B	37	31	34	64	64	32	;7d09c562;714dd2
04BD04C8	65	61	3B	36	38	38	32	64	37	34	34	3B	62	32	64	61	ea;6882d744;b2da
04BD04D8	35	63	34	30	3B	62	65	38	37	66	37	33	38	3B	66	63	5c40;be87f738;fc
04BD04E8	34	66	36	34	32	32	3B	63	37	62	37	61	39	3B	36	36	4f6422;c7b7a9;66
04BD04F8	61	38	65	65	61	64	3B	34	36	62	36	30	37	63	32	3B	a8eead;46b607c2;
04BD0508	64	62	31	61	63	37	62	62	3B	66	33	63	30	34	35	65	db1ac7bb;f3c045e
048D0518	34	3B	66	64	30	37	34	33	39	3B	73	71	6C	61	67	65	4;fd07439;sqlage
04BD0528	6E	74	24	2A	3B	35	62	64	33	31	61	34	61	3B	38	36	nt\$*;5bd31a4a;86
												100					at any large of t

Figure 25

The binary also decrypts a list of Sophos services that will be stopped:



Address	He	6															ASCII
04BD0048	33		39	34	35	35	65	34	3B	33	34	32	32	30	63	33	Be9455e4:34220c3
04BD0048	33	38	61	39	31	39	39	39	39	63	38	61	39	31	39	39	3:a919999c:a9199
04BD0058	39	39	63	38	63	36	63	66	64	61	61	64	38	39	61	31	99c;c6cfdaad;9a1
04BD00088	32	31	32	32	39	3B	35	35	32	62	39	64	64	62	38	33	21229:552b9ddb;3
	35	39	62	35		63	34	35 3B		36	37	64		65	36	30	
04BD0088					64				64				31				59b5dc4;d67d1e60
04BD0098	38	66	64	37	65	31	61	62	30	38	36	39	65	37	62	63	;fd7e1ab0;69e7bc
04BD00A8	61	35	38	65	33 3B	38	31	61	34	35	39	38	36	38 38	35	30	a5;e381a459;6850
04BD00B8	37	31	38	35			38	30	39	66		66	37		32	30	7185;5809f6f7;20
04BD00C8	33	62	62	38	39	63	38	64	64	31	30	39	31	34	34	38	3bb89c;dd109144;
04BD00D8	61	39	32	33	66	65	37	61	3B	35	35	66	30	61	36	30	a923fe7a;55f0a60
04BD00E8	34	3B	35	39	64	32	64	62	62	66	3B	32	37	34	36	32	4;59d2dbbf;27462
04BD00F8	66	66	66	38	38	63	62	34	31	35	34	63	38	39	61	34	fff;8cb4154c;9a4
04BD0108	66	37	66	34	33	3B	65	30	36	37	64	62	33	30	38	66	f7f43;e067db30;f
04BD0118	63	39	35	62	61	39	64	3B	63	63	35	66	35	62	66	31	c95ba9d;cc5f5bf1
04BD0128	38	34	36	37	32	35	35	65	34	38	37	32	64	39	62	35	;467255e4;72d9b5
04BD0138	39	63	3B	62	34	66	61	36	63	66	3B	31	37	32	62	31	9c;b4fa6cf;172b1
04BD0148	66	61	66	3B	35	65	65	65	63	37	31	35	3B	36	63	35	faf;5eeec715;6c5
04BD0158	62	31	39	32	37	3B	39	35	63	37	36	32	34	36	3B	66	b1927;95c76246;f
04BD0168	35	38	32	66	34	66	33	3B	61	33	32	32	31	65	62	31	582f4f3;a3221eb1
04BD0178	3B	39	33	61	37	66	32	32	31	3B	66	62	37	38	63	35	;93a7f221;fb78c5
04BD0188	33	3B	63	38	62	64	39	66	34	64	3B	61	65	36	34	63	3;c8bd9f4d;ae64c
04BD0198	36	62	33	ЗB	62	31	31	37	65	66	63	ЗB	33	36	30	62	6b3;b117efc;360b
04BD01A8	39	37	39	39	3B	33	62	39	66	31	62	33	65	3B	61	36	9799;3b9f1b3e;a6
04BD01B8	37	37	32	63	39	36	3B	62	65	38	37	66	37	33	38	3B	772c96;be87f738;
04BD01C8	63	39	39	31	34	37	61	36	3B	64	37	37	36	31	66	64	c99147a6;d7761fd
04BD01D8	64	3B	61	33	37	62	61	66	33	37	ЗB	61	37	62	30	39	d;a37baf37;a7b09
04BD01E8	61	65	65	3B	66	35	64	63	35	31	64	35	3B	66	39	32	aee; f5dc51d5; f92
04BD01F8	36	32	38	61	30	3B	35	32	66	32	62	38	31	31	3B	31	628a0;52f2b811;1
04BD0208	65	61	61	37	36	37	32	ЗB	65	36	33	66	39	30	30	34	eaa7672;e63f9004
04BD0218	3B	73	6F	70	68	6F	73	20	63	6C	69	65	6E	74	20	66	;sophos client f
04BD0228	69	72	65	77	61	6C	6C	2A	3B	38	39	36	64	36	39	61	irewall*;896d69a
04BD0238	37	3B	36	30	36	64	34	64	39	39	3B	65	30	33	35	37	7;606d4d99;e0357
04BD0248	36	35	38	3B	39	33	63	34	31	37	34	34	3B	61	63	31	658;93c41744;ac1
04BD0258	34	64	63	39	30	3B	64	61	65	62	64	61	33	62	38	73	4dc90;daebda3b;s
04BD0268	6F	70	68	6F	73	20	6D	63	73	2A	3B	61	38	34	34	32	ophos mcs*;a8442
04BD0278	32	39	39	3B	62	62	37	61	39	31	37	61	3B	65	33	37	299; bb7a917a; e37
04BD0288	66	39	64	30	38	3B	61	37	35	34	63	61	34	3B	32	36	f9d08;a754ca4;26
04BD0298	62	35	64	31	38	63	38	39	63	32	32	61	33	38	34	3B	b5d18c;9c22a384;
04BD02A8	31	66	30	35	64	34	36	63	3B	62	63	38	31	66	66	33	1f05d46c;bc81ff3
04BD02B8	39	3B	61	64	38	36	66	35	37	38	3B	34	30	32	62	65	9; ad86f578; 402be
04BD02C8	66	64	34	3B	73	6F	70	68	6F	73	20	77	65	62	20	69	fd4;sophos web i
04BD02D8	6E	74	65	6C	6C	69	67	65	6E	63	65	2A	3B	62	39	33	ntelligence*; b93
04BD02E8	30	33	62	34	33	3B	38	37	61	64	39	31	63	32	38	38	03b43;87ad91c2;8
04BD02F8	31	65	32	65	39	66	62	3B	73	6F	70	68	6F	73	70	61	1e2e9fb;sophospa
04BD0308	74	63	68	2A	3B	32	32	30	34	64	35	64	31	3B	38	33	tch*;2204d5d1;83
04BD0318	31	32	39	65	34	34	38	38	33	31	32	39	65	34	34	3B	129e44;83129e44;
04BD0328	64	34	62	66	61	62	37	38	3B	32	33	62	30	37	63	61	d4bfab78;23b07ca
04BD0338	30	3B	62	37	38	66	39	62	34	65	3B	39	38	37	31	36	0;b78f9b4e;98716
04BD0348	33	65	39	3B	34	35	61	31	63	31	39	37	3B	64	65	33	3e9;45a1c197;de3
04BD0358	64	61	62	63	37	38	66	63	34	66	36	34	32	32	3B	31	dabc7;fc4f6422;1
04BD0368	34	64	61	64	31	61	ЗB	35	33	61	32	33	35	33	62	3B	4dad1a;53a2353b;
				34	63	66	64	34	38	62	65	30	39	32	34	38	b154cfd4;be09248
04BD0378	62	31	35														
	31	3B 65	35 31 34	38 38	65	35	38	38	63 61	36	3B 62	35	39	65	38	34	1;18e588c6;59e84 ee4;5ec9a5bb

Figure 26

Grief appends the following extension to the file name of the encrypted files:

Address	Hex	ASCII
04BD0048	2E 70 61 79 30 72 67 72 69 65 66 00 00 00 00 00	.payOrgrief

Figure 27

The ransom note file name is also decrypted using RC4:

Address	He	< .															ASCII
04BD0048	2E	69	77	61	6E	74	32	73	75	72	76	69	76	65	2E	68	.iwant2survive.h
04BD0058	74	GD	6C	00	00	00	00	00	00	00	00	00	00	00	00	00	tml

Figure 28

An RSA public key that is Base64-encoded is decrypted by the process:



Address	He	ć.															ASCII
04BD0048	4D	49	49	42	49	6A	41	4E	42	67	6B	71	68	68	69	47	MIIBIjANBgkghkiG
04BD0058	39	77	30	42	41	51	45	46	41	41	4F	43	41	51	38	41	9w0BAQEFAAOCAQ8A
04BD0068	4D	49	49	42	43	67	4B	43	41	51	45	41	72	43	43	30	MIIBCGKCAQEArCCO
04BD0078	76	4C	48	70	6A	35	57	39	46	49	53	62	72	68	79	6A	vLHpj5W9FISbrhyj
04BD0088	OD	0A	55	77	65	33	34	62	56	30	46	7A	57	50	7A	57	Uwe34bV0FzWPzW
04BD0098	62	73	31	58	4D	4D	61	49	31	32	34	6C	30	2F	70	76	bs1XMMaI12410/pv
048D00A8	45	58	2B	34	74	39	48	4E	43	33	52	72	49	69	6B	32	EX+4t9HNC3RrIik2
04BD00B8	6E	65	76	6D	50	2B	6F	54	74	79	66	66	59	62	52	71	nevmP+oTtyffYbRq
04BD00C8	53	4B	OD	0A	32	46	4C	6F	66	35	43	64	54	4A	67	71	SK.,2FLof5CdTJgq
048D00D8	4C	6C	38	36	73	58	30	7A	2F	7A	4E	58	4A	69	30	2B	L186sX0z/zNXJi0+
04BD00E8	6A	31	37	65	36	67	66	38	63	59	4F	52	54	4D	65	75	j17e6gf8cYORTMeu
04BD00F8	GD	47	4E	36	2B	48	30	65	41	79	2B	58	50	54	53	45	mGN6+H0eAy+XPTSE
04BD0108	53	45	57	31	OD	0A	43	41	53	4D	53	55	4C	65	32	65	SEW1CASMSULe2e
04BD0118	6C	45	31	4B	47		63	45	4F	34	47	55	6D	2B	6B	47	lE1KG4cE04GUm+kG
04BD0128	76	31	33	47		59		74	61		59	47	36	54	46	53	v13GpYVtaGYG6TFS
04BD0138	50	48	44	2B	4E	39	49	4C	41	77	64	6C	43	46	30	39	PHD+N9ILAwd1CF09
04BD0148	4B	52	2B	51	30	56	OD	0A	70	56	67	44	GD	4E	56	69	KR+QOVpVgDmNVi
04BD0158	6C	4A	39	4D	73	31	63	30	72	70	56	35	4B	71	44	63	1J9Ms1c0rpV5KqDc
04BD0168	33	41	48	31	55	2F	66	69	30	77	58	77	45	69	6F	53	3AH1U/fiOwXwEioS
04BD0178	2F	31	72	42	65	6B	70	30	48	54	63	45	5A	38	48	44	/1rBekpOHTCEZ8HD
04BD0188	32	44	49	6A	66	56	57	71	OD	0A	6C	4E	52	30	4C	61	2DIjfVWqlNROLa
04BD0198	4B	42	59	5A	36	58	69	4A	61	70	36	64	43	2B	43	52	KBYZ6XiJap6dC+CR
048D01A8	4F	73	44	53	76	65	31	6A	38	62	69	47	42	74	69	6A	OsDSve1j8biGBtij
04BD01B8	50	73	76	30	44	72	7A	36	77	63	78	2F	31	59	59	31	Psv0Drz6wcx/1YY1
04BD01C8	42	2F	65	67	51	69	6C	4B	77	31	0D	0A	79	51	49	44	B/egQilKw1yQID
04BD01D8	41	51	41	42	OD	0A	00	00	00	00	00	00	00	00	00	00	AQAB

Figure 29

The content of the ransom note is also revealed:

Address	He	¢															ASCII
04BD0048	3C	68	74	6D	6C	3E	ЗC	68	65	61	64	3E	3C	73	74	79	<html><head><sty< th=""></sty<></head></html>
04BD0058	6C	65	20	74	79	70	65	3D	22	74	65	78	74	2F	63	73	le type="text/cs
04BD0068	73	22	3E	40	66	6F	6E	74	2D	66	61	63	65	7B	66	6F	s">@font-face{fo
04BD0078	6E	74	2D	66	61	6D	69	6C	79	3A	20	27	54	6F	6D	6F	nt-family: 'Tomo
04BD0088	72	72	6F	77	27	3B	20	66	6F	6E	74	2D	73	74	79	6C	rrow'; font-styl
04BD0098	65	3A		6E			6D			3B						2D	e: normal; font-
04BD00A8	77	65	69	67	68	74	ЗA	20	34	30	30	3B	20	66	6F	6E	weight: 400; fon
04BD00B8	74	2D	64	69	73	70	6C	61	79	ЗA	20	73	77	61	70	3B	t-display: swap;
04BD00C8	20	73	72	63												3A	<pre>src: url(https:</pre>
04BD00D8	2F	2F	66							73							//fonts.gstatic.
04BD00E8			GD							6F							
04BD00F8	35	2F	57	42	4C	GD	72	45	54	4E	62	46	74	5A	43	65	5/WBLmrETNbFtZCe
04BD0108		71	-	52			65			77							
04BD0118	66	32	29	20			72			74							
04BD0128	32		29	3B						6F							
04BD0138	67	65	-	20													ge: U+0100-024F,
04BD0148	20	55		30													U+0259, U+1E00-
04BD0158	31	45	46	46	2C	20	55	2B	32	30	32	30	2C	20	55	2B	1EFF, U+2020, U+
04BD0168	32	30	41	30	2D	32	30	41	42	2C	20	55	2B	32	30	41	20A0-20AB, U+20A

Figure 30

The LegalNoticeCaption and LegalNoticeText registry values will be modified to contain the client's name, a password, and the Dark web link that needs to be accessed in order to communicate with the threat actor. We've redacted the company name, however, we've confirmed that it was listed on the Grief's page:

Address	He	<u>(</u>													ASC	III		
04BD0048															C			
04BD0058 04BD0068		75	20	61	72	65	20	CC	75	65	64	2E	00	79	ou	are	fueleo	У

Figure 31



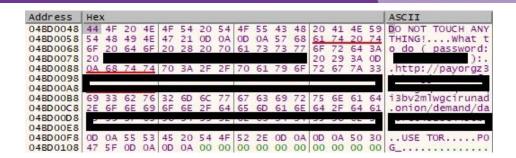


Figure 32

The process also decrypts the Windows Defender Registry Key and the DisableAntiSpyware registry key, which will be utilized to turn off Microsoft Defender Antivirus:

Address	He	< .															ASCII
04E76728	53	00	4F	00	46	00	54	00	57	00	41	00	52	00	45	00	S.O.F.T.W.A.R.E.
04E76738	5C	00	50	00	6F	00	6C	00	69	00	63	00	69	00	65	00	\.P.o.1.i.c.i.e.
04E76748	73	00	5C	00	4D	00	69	00	63	00	72	00	6F	00	73	00	s.\.M.i.c.r.o.s.
04E76758	6F	00	66	00	74	00	5C	00	57	00	69	00	GE	00	64	00	o.f.t.\.W.i.n.d.
04E76768	6F	00	77	00	73	00	20	00	44	00	65	00	66	00	65	00	o.w.sD.e.f.e.
04E76778	6E	00	64	00	65	00	72	00	00	00	00	00	00	00	00	00	n.d.e.r

Figure 33

Address	He	¢															ASCII
04D00048	44	00	69	00	73	00	61	00	62	00	6C	00	65	00	41	00	D.i.s.a.b.l.e.A.
04D00058	6E	00	74	00	69	00	53	00	70	00	79	00	77	00	61	00	n.t.i.S.p.y.w.a.
04D00068	72	00	65	00	00	00	00	00	00	00	00	00	00	00	00	00	r.e

Figure 34

A list of commands that will be used to delete the Volume Shadow Copies is decrypted by the ransomware:

Address	He	ĸ															ASCII
04E76728	44	00	65	00	6C	00	65	00	74	00	65	00	20	00	53	00	D.e.1.e.t.eS
04E76738	68	00	61	00	64	00	6F	00	77	00	73	00	20	00	2F	00	h.a.d.o.w.s/
04E76748	41	00	6C	00	6C	00	20	00	2F	00	51	00	75	00	69	00	A.1.1/.Q.u.i
04E76758	65	00	74	00	00	00	00	00	00	00	00	00	00	00	00	00	e.t



Address	He	ĸ															ASCII		
04E76728	64	65	6C	65	74	65	20	73	68	61	64	6F	77	73	20	61	delete	shadows	a
04E76738	6C	6C	OD	0A	65	78	69	74	OD	0A	00	00	00	00	00	00	11exi	t	



Grief decrypts even more data using RC4, however, we've included the other less relevant strings in the appendix for completeness.

The ransom note called ".iwant2survive.html" is displayed in figure 37:

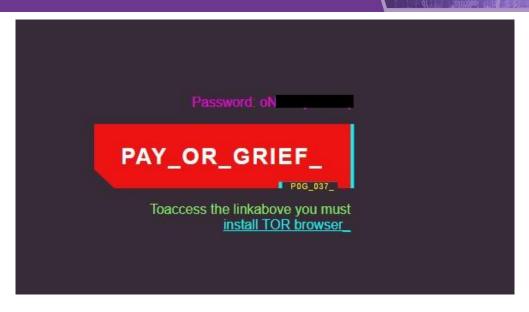


Figure 37

ExpandEnvironmentStringsA is utilized to expand an environment-variable string and replace it with the value defined for the current user:

6A7B5B7B FF 74 24 3C 6A7B5B7F FF 74 24 3C 6A7B5B83 FF 33	push dword ptr ss:[esp+3C] push dword ptr ss:[esp+3C] push dword ptr ds:[ebx]	[eb)	x875W_C1 0 x875W_C0 0 x875W_E5 0 x875W_SF 0 x875W_P 1 x875W_U 0
eax= <kernel32.expandenvironmentstringsa> (76A74920)</kernel32.expandenvironmentstringsa>	call eax	eax: v	Default (stdcall) ▼ 5 ↓ Unlock 1: [esp] 04BD29A0 "%ProgramData%\\Microsoft\\Wi 2: [esp+4] 04BD0048
.text:6A7B5B85 dump_patched.dll:\$5B85 #5B85			2: [csp+4] 040002800 4: [csp+C] 00CEFA14
Ump 1 Ump 2 Ump 3 Ump 4 Um Dump 4	o 5 🧐 Watch 1 🛛 🕼 🖉 Struct	OOCEF8E4 04BD	
Address Hex 048029A0 25 50 72 67 72 61 60 44 61 74 61 25 5C 40 048029A0 63 72 67 73 67 66 74 5C 57 69 66 64 67 77 7 048029B0 57 45 52 5C 52 67 70 67 69 66 64 67 77 7 048029C0 57 45 52 5C 52 67 70 67 72 63 66 74 57 69 66 76 77 71 048029C0 57 45 52 52 52 65 70 67 72 74 51 75 63 65 76 63 74 51 75 63 76 77 73 74 51 75 63	S 5C crosoft\Windows\	OOCEF8E8 0000 OOCEF8EC 00CE OOCEF8E0 0000 OOCEF8E4 OOCEF8E4 OOCEF8E4 OOCEF8E4 OOCEF8E4 OOCEF8E4 OOCEF8E4 OOCEF8E4 OOCEF8E4 OOCEF8E4 OOCEF8E4 OOCEF8E4 OOCEF8E4 OOCEF8E4 OOCEF8E4 OOCEF8E4 OOCEF8E4 OOCEF8E4 OOCEF8E4 OOCEF8E4	FA14 00008

Figure 38

The malicious process extracts the NetBIOS name of the local computer via a function call to GetComputerNameW:

	• 6A7C11AF • 6A7C11B0		24 04	p	oush edx oush dword p	tr ss:[esp+	4]			x875W_U 0
	6A7C11B4	FF DO		-	all eax	a	i	eax: v	Default (stdcall) 1: [esp] 04BD48F8	▼ 5 🕏 🗆 Unlock
eax= <kernel32. .text:6A7C11B4</kernel32. 									2: [esp+4] 00CEF8CC 3: [esp+8] 04BD48F8 4: [esp+C] 00000080	
Dump 1	Dump 2	Dump 3	Dump 4	Dump 5	🛞 Watch 1	[x=] Locals	Struct	00CEF8BC 048 00CEF8C0 000		

Figure 39

The binary acquires a handle to a key container within a CSP (cryptographic service provider) using the CryptAcquireContextW API ($0x18 = PROV_RSA_AES$, $0xF0000000 = CRYPT_VERIFYCONTEXT$):



● GA7C66EF ● GA7C66F4 ● GA7C66F6 ● GA7C66F7 ● GA7C66F7 ● GA7C66F8 ■ GA7C66F8 ■ GA7C66F8	68 00 00 00 F0 6A 18 55 55 52 FF D0	push F0000000 push 18 push ebp push ebp push ebp call eax	eax)	Default (stdcall)	0 x87SW_ES 0
eax= <advapi32.cryptacquirec< td=""><td></td><td></td><td></td><td>1: [esp] 00CEF8B8 2: [esp+4] 0000000 3: [esp+8] 0000000 4: [esp+C] 0000018</td><td></td></advapi32.cryptacquirec<>				1: [esp] 00CEF8B8 2: [esp+4] 0000000 3: [esp+8] 0000000 4: [esp+C] 0000018	
Her Dump 1 Dump 2 Her Address Hex 00CEF5888 00 00 00 00 00 00 00 00 00	Dump 3 U Dump 4 U	ASCII	00CEF894 00C 00CEF895 000 00CEF89C 000 00CEF8A0 000 00CEF8A4 F00	00000 00000 00018	

Figure 40

The CryptCreateHash function is used to create a handle to a CSP hash object (0x8003 = **CALG_MD5**):

.text:6A7C		ed.dll:\$16733 #16733		x=] Locals 🎾 Struct	00CEF894 0 00CEF898 0 00CEF89C 0	0008003		
.text:6A7C	6733 dump_patche	ed.dll:\$16733 #16733			000000000000000000000000000000000000000	001638		
IP eax= <advap< th=""><th>647C6732 647C6733 (132.CryptCreated</th><th>56 FF D0 4ash> (73A5FB50)</th><th>push esi call eax</th><th></th><th>eax</th><th>Default (s 1: [esp 2: [esp 3: [esp</th><th></th><th>s 🛟 🗌 Unloc ext></th></advap<>	647C6732 647C6733 (132.CryptCreated	56 FF D0 4ash> (73A5FB50)	push esi call eax		eax	Default (s 1: [esp 2: [esp 3: [esp		s 🛟 🗌 Unloc ext>
	 6A7C672C 6A7C672D 6A7C672E 	52 55 55 FF 74 24 40	push edx push ebp push ebp push dword ptr	ss:[esp+40]		x875W_	B 0 x87SW_C3 0 x87SW_C C1 0 x87SW_C0 0 x87SW_E SF 0 x87SW_P 1 x87SW_U 2 x87SW_P 1 x87SW_U	5 0

Figure 41

The binary computes the MD5 hash of the computer name by calling the CryptHashData routine:

	x				ASCII	1		OOCEE	8A0 000	00000E
Dump 1	Ump 2	Ump 3	Ump 4	Ump 5	🥘 Watch 1	[x=] Locals	🎾 Struct	OOCEF		BD2F88 "DESKTOP-
eax= <advapi< th=""><th>32.CryptHas</th><th></th><th>5FC90) 16760 #16760</th><th>6</th><th></th><th></th><th></th><th></th><th></th><th>1: [esp] 030C88C8 <&CPCreateHash> 2: [esp+4] 048D2F88 " 3: [esp+8] 0000000F 4: [esp+C] 00000000</th></advapi<>	32.CryptHas		5FC90) 16760 #16760	6						1: [esp] 030C88C8 <&CPCreateHash> 2: [esp+4] 048D2F88 " 3: [esp+8] 0000000F 4: [esp+C] 00000000
	GA7C67	50 FF D			all eax				eax: v	Default (stdcall) 🔻 5 🗘 🗌 Unlod
	 6A7C67 6A7C67 6A7C67 6A7C67 6A7C67 	SA FF 7 SE 53 SF 55	4 24 34	5	oush o oush dword p oush ebx oush ebp	tr ss: <mark>[</mark> esp+	34		[es] ebx:	X8/5W_B 0 X8/5W_C3 0 X8/5W_C2 0 x875W_C1 0 X875W_C0 0 X875W_E5 0 X875W_SF 0 X875W_P 1 X875W_U 0

Figure 42

The hash size (16 bytes) is extracted by calling the CryptGetHashParam API ($0x4 = HP_HASHSIZE$):

ETP A GATCG79E EF DO Call eav	Dump 1	Dump 2	19 22 110 100 100	📖 Dump 5 🛛 🛞 Watch 1	[x=] Locals	Struct	00CEF898 000	0000004	
310 Call eax Call eax Construction Call eax Call eax <t< th=""><th>text:6A7C6</th><th>79F dump_patche</th><th>d.dll:\$16/9F #16/9</th><th>1</th><th></th><th></th><th></th><th></th><th></th></t<>	text:6A7C6	79F dump_patche	d.dll:\$16/9F #16/9	1					
6 A7C6798 GA 00 push 0 x875tatUsWord 0020 6 A7C679A 52 push edx x875W_C1 0 x875W_C2 0 6 GA7C679B 53 push ebx x875W_C1 0 x875W_C2 0 x875W_C2 0 6 GA7C679E 53 push ebx x875W_C5 0 x875W_D 0 x875W_D 0 6 GA7C679E 6 A 04 push 4 x855W_D 0 x875W_D 0 x875W_D 0	eax= <advapi< th=""><th>6A7C679B 6A7C679C 6A7C679E 6A7C679E 6A7C679E €</th><th>53 6A 04 55 FF D0 Param> (73A5FAB0)</th><th>push edx push ebx push 4 push ebp call eax</th><th>*</th><th></th><th></th><th>x875w_C1 0 x875w_C0 0 x875w_SF 0 x875w_P 1 Default (stdcall) 1: [esp] 030C88C8 <4CPCrea 2: [esp+4] 0000004 3: [esp+8] 00CER88C</th><th>x875W_ES 0 x875W_U 0 ▼ 5 🚖 🗌 L</th></advapi<>	6A7C679B 6A7C679C 6A7C679E 6A7C679E 6A7C679E €	53 6A 04 55 FF D0 Param> (73A5FAB0)	push edx push ebx push 4 push ebp call eax	*			x875w_C1 0 x875w_C0 0 x875w_SF 0 x875w_P 1 Default (stdcall) 1: [esp] 030C88C8 <4CPCrea 2: [esp+4] 0000004 3: [esp+8] 00CER88C	x875W_ES 0 x875W_U 0 ▼ 5 🚖 🗌 L



The hash value is retrieved using the same API ($0x2 = HP_HASHVAL$):



								i I		01 0100000
EIP	6A7C67D6 6A7C67D8 6A7C67D0 6A7C67D0 6A7C67D0 6A7C67D0 6A7C67E1	6A 00 8D 54 50 6A 02 55 FF D3	24 18		push 0 lea edx,dword push edx push eax push 2 push ebp call ebx	ptr ss: [e	sp+18 <mark>)</mark>		ebx: v	Default (stdcall) 🔻 5 🔹 🗌 Unlock
ebx= <advapi32< th=""><th>1 dump_patche</th><th>ed.dll:\$16</th><th>7E1 #167E1</th><th></th><th><i>6</i>5</th><th></th><th>Struct</th><th>DOCEF</th><th>894 030</th><th>1: [esp] 030C88C8 <&CPCreateHash> 2: [esp+4] 00000002 3: [esp+4] 000EF88C 4: [esp+C] 00CEF88C</th></advapi32<>	1 dump_patche	ed.dll:\$16	7E1 #167E1		<i>6</i> 5		Struct	DOCEF	894 030	1: [esp] 030C88C8 <&CPCreateHash> 2: [esp+4] 00000002 3: [esp+4] 000EF88C 4: [esp+C] 00CEF88C
Address Hex	Dump 2	Dump 3	Dump 4	Dump 5	Watch 1	[x=] Locals	& Struct	00CEF	898 000 89C 04B	000002 3D2B98
04BD2B98 00 0	00 00 00 00 00	00 00 00	00 00 00	00 00 00 00				00CEF	8A0 000 8A4 000	EF8BC
04BD2B98 00 0 Address Hex					ASCII			OUCEF	000	
04BD2B98 CC	47 82 6A 90 CO) 3F 88 85	64 B5 18	C7 4F 24 AD	1G.].A?dµ	.ços.				

Figure 44

From our analysis and the OSINT, one of the parameters that Grief is supposed to run with is "-<First 6 chars from the hash value>". Based on this observation, the parameter changes from one host to another.

The binary retrieves the command line string for the current process:

<pre>eax=<kernel32.getcommandlinew> (76A74BC0) .text:6A7C12FC dump_patched.dll:\$112FC #112FC</kernel32.getcommandlinew></pre>			1: [esp] 00000000 2: [esp+4] 00CEFB1C 3: [esp+6] 00CEFB18 4: [esp+C] 00CEF918
GA7C12FC FF D0	call eax	eax: v	Default (stdcall) 🔻 5 🖨 🗌 Unlock
GA7C12E7 BA F0 D1 9F EE GA7C12E7 GA7C12F1 S0 GA7C12F1 S0 GA7C12F2 S0 GA7C12F3 S2 GA7C12F8 S2 GA7C12F8 S S GA7C12FA 7 4 G	<pre>mov eax,ESPD1F0 mov eax,ESPD1F0 mov eax,SDC84A66 push eax push eax kall dump_patched.6A7C1824 test eax,eax je dump_patched.6A7C1302</pre>	eax: edx: eax: eax: eax:	A0/IN_0 5 (EmpLy) A0/IN_/ 5 (EmpLy) x875tatusWord 0020 x875W_6 0 x875W_6 0 020 x875W_6 0 x875W_6 0 x875W_6 0 x875W_6 0 x875W_6 0 x875W_6 0 x875W_5 0 x875W_9 1 x875W_0 0 x875W_5 0 x875W_9 1 x875W_0 0

Figure 45

CommandLineToArgvW is utilized to parse the command line string and return an array of pointers to the cmd line arguments:

	 6A7C1290 6A7C1290 	57		p	oush edx oush edi			edi:	x875W_SF 0 x875W_P 1 x875W_U 0	
EIP	→ <u>6A7C129</u>				all eax			eax: v	Default (stdcall) 1: [esp] 030B21A6 L"\"C:\\Windows\\SysW	
	2.CommandLin 29E dump_pat								2: [esp+4] 002EF8D8 3: [esp+8] 0000000 4: [esp+6] 000EF8DC	
Dump 1	Ump 2	Ump 3	Ump 4	Ump 5	🧶 Watch 1	[x=] Locals	Struct	00CEF8D0 030 00CEF8D4 000	0B21A6 L"\"C:\\Windows\\SysWOW64\\rundll32 CEF8D8	2.exe\"

Figure 46

The malicious process retrieves the path of the executable of the current process via a function call to GetModuleFileNameW:

	 6A7B609 6A7B609 6A7B609 6A7B609 	8 FF 7			push ecx push dword p push 0	tr ds:[ebx	+C])	(875W_C1 0 x875W_C0 (875W_SF 0 x875W_P	
EIP	GA7B609				call eax	d d		•	eax	D	efault (stdcall)	👻 💈 🗖 Unloci
	132.GetModule									2	: [esp] 00000000 : [esp+4] 032FA970 : [esp+8] 00000200 : [esp+C] 00000000	
Dump 1	Ump 2	Dump 3	Dump 4	Dump 5	🛞 Watch 1	[x=] Locals	2 Str	uct	00CEF8E4 0 00CEF8E8 0			
Adda and Ltr					LACCTT	1			OOCEESEC 0	000002	00	

Figure 47

The ransomware also computes the MD5 hash of the string "1<Computer Name extracted earlier>":



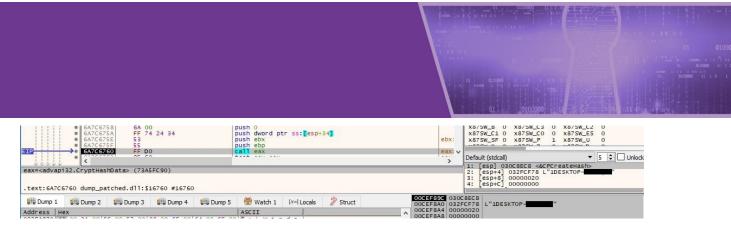


Figure 48

The file creates an event object using the NtCreateEvent routine (0x1F0003 = **EVENT_ALL_ACCESS**):

Jump 1 📲 Dump 2 📲 Dump 3 📲 Dump 4 📲 Dump 5 🧐 Watch 1 💷 Locals 🥖 Struct 🛛 OOCEF860	53 00CEF894 6C 001F0003 70 00CEF898
• 6A7C4F63 68 03 00 1F 00 push 1F00003 • 6A7C4F63 55 push ebp • 6A7C4F63 FF D6 call esi • call esi call esi call esi • call esi call esi call esi • call esi call esi call esi	x875W_SF 0 x875W_P 1 x875W_U 0 befault (stdcal) 5 ↓ Unlock 1: [esp] 00CEF894 2: [esp+4] 001F0003 3: [esp+4] 002F898 4: [esp+4] 000F898 4: [esp+4] 0000000 1: [esp+6] 0000000

Figure 49

The process creates a mutant object by calling the NtCreateMutant function (0x1F0001 = **MUTEX_ALL_ACCESS**):

Address Hex		ASCII	^ 00CE	F870 00CE	F890
Dump 1 Dump 2	💭 Dump 3 🛛 💭 Dump 4 💭 Dump	5 🛞 Watch 1 🛛 🕅 🕅 🕅 🕅 🕅	Struct OOCE	F868 00CE F86C 001F	0001
esi= <ntdll.ntcreatemutan .text:6A7C4AB1 dump_patcl</ntdll.ntcreatemutan 				>	Default (stocall)
6A7C4AA8 6A7C4AA8 6A7C4AA8 6A7C4AB0 6A7C4AB0 6A7C4AB1	6A 00 50 68 01 00 1F 00 52 FF D6	push 0 push eax push 1F0001 push edx call est		esi: v	x875W_B 0 x875W_C2 0 x875W_C2 0 x875W_C1 0 x875W_C0 0 x875W_E5 0 x875W_SF 0 x875W_P 1 x875W_U 0 Default (stdcall)

Figure 50

The malware decodes the Base64-encoded RSA public key using the CryptStringToBinaryA function ($0x1 = CRYPT_STRING_BASE64$):

Image: Construction Construction Construction Construction Construction Construction Construction Construction <t< th=""><th></th><th> 6A7C5D32 6A7C5D33 6A7C5D34 6A7C5D38 6A7C5D38 6A7C5D38 6A7C5D38 6A7C5D38 6A7C5D38 6A7C5D38 6A7C5D38 </th><th>51 80 44 24 18 50 52 56 51 53</th><th>push eax push edx push esi push ecx push ebx</th><th>ord ptr ss:[e</th><th>sp+18<mark>]</mark></th><th></th><th></th><th>x87 x87 x87 x87 x87 x87</th><th><pre>rr6 3FFF80000000000000000 ST6 Empty 1.000000 rr7 3FFF800700750F22363 ST7 Empty 1.1071487 'TagNord FFFF 'Tw_0 3 (Empty) x87Tw_1 3 (Empty) Tw_3 2 (Empty) x87Tw_3 3 (Empty) Tw_4 3 (Empty) x87Tw_5 3 (Empty)</pre></th></t<>		 6A7C5D32 6A7C5D33 6A7C5D34 6A7C5D38 6A7C5D38 6A7C5D38 6A7C5D38 6A7C5D38 6A7C5D38 6A7C5D38 6A7C5D38 	51 80 44 24 18 50 52 56 51 53	push eax push edx push esi push ecx push ebx	ord ptr ss:[e	sp+18 <mark>]</mark>			x87 x87 x87 x87 x87 x87	<pre>rr6 3FFF80000000000000000 ST6 Empty 1.000000 rr7 3FFF800700750F22363 ST7 Empty 1.1071487 'TagNord FFFF 'Tw_0 3 (Empty) x87Tw_1 3 (Empty) Tw_3 2 (Empty) x87Tw_3 3 (Empty) Tw_4 3 (Empty) x87Tw_5 3 (Empty)</pre>
bep-crypt32.Cr	EIP	GA7C5D3D	FF DS	call ebp					Defa	
Address Hex Acting Acting <th>.text:6A70</th> <th>C5D3D dump_patch</th> <th>d.dll:\$15D3D #15D3E</th> <th>, ,</th> <th></th> <th>(Å)</th> <th></th> <th>00CEF67C</th> <th>2: 3: 4:</th> <th>[esp+4] 0000000 [esp+8] 0000001 [esp+C] 048D48F8</th>	.text:6A70	C5D3D dump_patch	d.dll:\$15D3D #15D3E	, ,		(Å)		00CEF67C	2: 3: 4:	[esp+4] 0000000 [esp+8] 0000001 [esp+C] 048D48F8
D32FC460 D0 D4 D4 D4 D4 D4 D6 D4 D4 <thd4< th=""> D4 D4 <t< th=""><th></th><th></th><th>Dump 3 Ully Dump 4</th><th></th><th>1 [x=] Locals</th><th>2 Struct</th><th></th><th>00CEF680</th><th>00000000</th><th></th></t<></thd4<>			Dump 3 Ully Dump 4		1 [x=] Locals	2 Struct		00CEF680	00000000	
327C588 (0 4 2 4 2 4 5 7 8 4 2 5 8 4 5 7 8 4 3 4 5 7 8 4 5 7 8 4 3 6 1 7 2 4 3 8 10 MIEGORAGEACO 327C588 (0 5 7 6 5 3 5 7 6 5 3 5 7 6 5 3 5 7 6 5 3 5 7 6 5 3 5 7 7 6 5 3 6 7 7 8 7 8 7 7 8 7	032FC868	4D 49 49 42 49 6/	41 4E 42 67 6B 71	68 68 69 47 MIIBIIAN	BgkqhkiG		<u>^</u>	00CEF688	048D48F8	3
3327C988 34 44 32155 27 45 69 53 34412/100/www.clos 000000000000000000000000000000000000)32FC898)32FC848)32FC848)32FC808)32FC808)32FC808)32FC808)32FC988)32FC918)32FC918)32FC938)32FC938)32FC938)32FC988)2FC988)2FC988	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	72 66 79 6A VLHDJŠWD 71 92 6A 77 192 7A 75 19	TJSDrhyj VOFZWPZW 12410/pv JSRrI1k2 tyffYDRq fScdTJqq /zNXJ10+ tyNKTSE SMSUL22c VORTMeu AyrXMTSE SMSUL22c J4GUm4KG AGYGSTFS Mwd1cFp J4GUm4KG AGYGSTFS Mwd1cFp J4GUm4KG J4GUT TCEZ8MD INROLA BbiGBtij wcx(JYYL			00CEF694 00CEF69C 00CEF69C 00CEF6A2 00CEF6A4 00CEF6A4 00CEF6A8 00CEF6A8 00CEF688 00CEF688 00CEF688 00CEF680 00CEF6C0 00CEF6C0 00CEF604 00CEF604 00CEF604	0000000 00000126 04804878 00000126 000000126 00000026 000000126 000000126 00000016 6478881 0000000 0480460 0000000 0480460 0000000 0000000 0000000 0000000 000000	return to dump_patched.6478EB16 from dum d"WIIBIJANBgkqhk169w0BAQEFAAOcAQ8AWIIBCgH return to dump_patched.647C5F24 from dump



your digital world, secured

Address	He	¢															ASCII
04BD48F8	30	82	01	22	30	OD	06	09	2A	86	48	86	F7	OD	01	01	0"0*.H.÷
04BD4908														82			
04BD4918	00	AC	20	B4	BC	B1	E9	8F	95	BD	14	84	9B	AE	10	A3	.¬ ¼±é½®.f
																	S. · áµt.5. ÍfìÖs.h
04BD4938																	
																	".ib%cþ;;r}ö.F¤.
																	ØReL.*:±}3ÿ
																	3W&->.^pê.üqLÇ
04BD4978	AE	98	63	7A	F8	7D	1E	03	2F	97	3D	34	84	48	45	B5	◎.czø}/.=4.HEµ
04BD4988	08	04	8C	49	42	DE	D9	E9	44	D4	A1	B 8	70	43	B 8	19	IBÞÚÉDÖ; pC .
																	I%.kõÜjXVÖ. n#
																	<pre>Ç.ã} °OvP.ÓÒ.ù</pre>
																	¥XÖbL*W40.y*
																	UU.ős÷âó.ð.*.ÿZ
																	Azjt.7.gAAØ2#}Uª
																	.Ôt-¢.a@éĐ¾.
04BD49F8																	
04BD 4A08	EB	CF	AC	10	C7	FD	58	63	50	7F	7A	04	22	94	AC	35	ëϬ.ÇýXcP.z.".¬5
04BD4A18	C9	02	03	01	00	01	00	00	00	00	00	00	00	00	00	00	É

Figure 52

Grief decodes a structure of the **X509_PUBLIC_KEY_INFO** type by calling the CryptDecodeObject API (0x10001 = **PKCS_7_ASN_ENCODING** | **X509_ASN_ENCODING**, 0x8 = **X509_PUBLIC_KEY_INFO**):

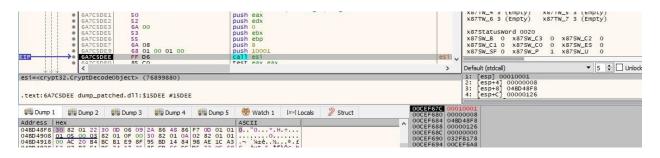
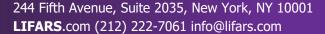


Figure 53

Address	He	ĸ															ASCII
032FB178	90	B1	2F	03	02	00	00	00	A8	B1	2F	03	OE	01	00	00	.±/ ±/
032FB188	BO	B1	2F	03	00	00	00	00	31	2E	32	2E	38	34	30	2E	°±/1.2.840.
032FB198	31	31	33	35	34	39	2E	31	2E	31	2E	31	00	00	00	00	113549.1.1.1
032FB1A8	05	00	00	00	00	00	00	00	30	82	01	0A	02	82	01	01	
032FB1B8	00	AC	20	B 4	BC	B1	E9	8F	95	BD	14	84	9B	AE	10	A3	'¼±é½@.£
032FB1C8				E1												68	
032FB1D8				5D													.vâ]?¦ñ.û.}.Đ·F⁼
032FB1E8	22	93	69	DE												8A	
032FB1F8				7F										7D			ØReL.*:±}3ÿ
032FB208				2D													3W&->.^Þê.üqLÇ
032FB218				7A													
032FB228				49													IBÞÙÉDÔ; pC .
032FB238				6B													I¼.kõÜjXVÖ. n. #
032FB248				7D												15	
032FB258																	¥XÔbL*W40.y*
032FB268				01													
032FB278				74													
032FB288				2D													
032FB298				2B													+ÞÖ? m.312@
032FB2A8				10													ēI¬.ÇýXcP.z.".¬5
032FB2B8	C9	02	03	01	00	01	00	00	00	00	00	00	00	00	00	00	É

your digital world, secured



CryptImportPublicKeyInfo is utilized to convert and import the RSA public key information into the provider (0x10001 = **PKCS_7_ASN_ENCODING** | **X509_ASN_ENCODING**):

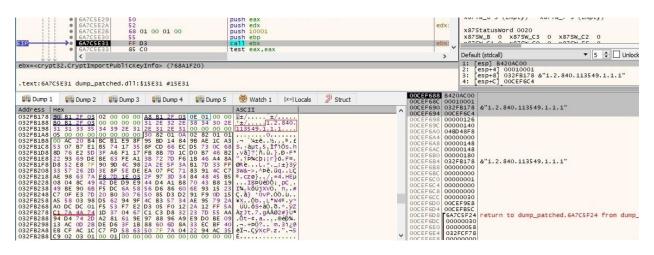


Figure 55

The RSA public key is in the ASN.1 format, and a great explanation of this format is presented at [4]. The public key is used to encrypt the generated AES file encryption keys. We were not able to reach the point where the malware encrypts the files due to the lack of the initial parameters.

Grief also implements the Heaven's Gate technique, which is fully described at [5]. Shortly, the process running as a 32-bit binary switches to the 64-bit environment and executes some instructions there. As we can see in figure 56, the binary pushes 0x33 (the segment selector) on the stack and calls the next line. The retf instruction is a "far return" and specifies the address where the execution returns and the segment. The code that starts after the retf instruction should be interpreted as 64-bit and debugged accordingly (for example, using WinDbg because x64dbg or the IDA Pro debugger can't be used to perform the switch).

.text:6A997D34		
.text:6A997D34		
.text:6A997D34	; Attr	ibutes: bp-based frame
.text:6A997D34		
.text:6A997D34	sub_6A	997D34 proc far
.text:6A997D34		
.text:6A997D34	var_10	= dword ptr -10h
.text:6A997D34		
.text:6A997D34	push	ebp
.text:6A997D35	mov	ebp, esp
.text:6A997D37	push	esi
.text:6A997D38	push	esi
.text:6A997D39	push	33h ; '3'
.text:6A997D3B	call	\$+5
.text:6A997D40	add	[esp+10h+var_10], 5
.text:6A997D44	retf	
.text:6A997D44	SUB 6A	997D34 endp ; sp-analysis faile
.text:6A997D44		

Figure 56



Deletion of Volume Shadow Copies using vssadmin and Diskshadow. Disable Microsoft Defender Antivirus

The ransomware initializes the COM library for use by the calling thread using the CoInitializeEx API ($0x2 = COINIT_APARTMENTTHREADED$):

	6AA488C 6AA488C	5 6A 00		pu	ish 2 ish 0 11 eax						(875W_U 0
EIP	→ 6AA4880				all eax			×	Defa	ault (stdcall) [esp] 00000000	▼ 5 🗘 🗌 Unlock
		izeEx> (76B0							2:	[esp+4] 00000002 [esp+8] 00000000	
Dump 1	Ump 2	🚚 Dump 3	💭 Dump 4	Dump 5	🛞 Watch 1	[x=] Locals	Struct	00EDF81C 0000 00EDF820 0000			

Figure 57

The binary calls the CoCreateInstance function in order to create a Group Policy Object with the CLSID {EA502722-A23D-11D1-A7D3-0000F87571E3} (0x1 = **CLSCTX_INPROC_SERVER**):

 GAA488F1 GAA488F2 GAA488F4 GA GAA488F4 GA GAA488F6 SD SS GAA488FC SI 	push ec push 1 push 0 1ea eco push ec	x,dword ptr ds:[eax+80] dx x,dword ptr ds:[eax+90] cx		x87StatusWord 0020 x87SW_B 0 x87SW_C3 x87SW_C1 0 x87SW_C0	0 x875W_C2 0 0 x875W_E2 0 0 x875W_E5 0 1 x875W_U 0
ebp= <combase.cocreateinstance> (768 .text:6AA488FD dump_patched.dll:\$88</combase.cocreateinstance>		DP	>	Default (stdcall) 1: [esp] 00EDF8F8 2: [esp+4] 00000000 3: [esp+8] 00000001 4: [esp+C] 00EDF8E8	▼ 5 🗘 Unlock
💭 Dump 1 💭 Dump 2 💭 Dump 3	💷 Dump 4 🛛 💷 Dump 5 🛛 🎯 V	Natch 1 🛛 🛛 🖉 Struct		0000000	
Address Hex 00EDF8F8 22 27 50 EA 3D A2 D1 11 A7 00EDF808 EC EE 77 74 00 00 00 00 00	ASCI: D3 00 00 F8 75 71 E3 "'Pê: EE EE 76 90 40 74 74 (iðwt		00EDF818 00EDF81C 00EDF820 00EDF820 0	OEDF8E8	

Figure 58

The OpenLocalMachineGPO method is used to open the default GPO for the computer and load the registry information (0x1 =**GPO_OPEN_LOAD_REGISTRY**):

310 → 6AA45510 FF 52 14 call dword ptr ds:[edX+14] > Default (stdcall) ▼ S □ Unlod dword ptr [edx+14]=[gpedit.68CD1880]=gpedit.68D11A80 2: [esp+4] 00000001 3: [esp+6] 00000000 4: [esp+6] 000000000 4: [esp+6] 00000000 4: [esp+6] 000000000 4: [esp+6] 00000000 4: [esp+6] 00000000 4: [esp+6] 00000000 4: [esp+6] 00000000 4: [esp+6] 000000000 4: [esp+6] 000000000 4: [esp+6] 00000000 4: [esp+6] 000000000 4: [esp+6] 00000000 4: [esp+6] 000000000 4: [esp+6] 000000000 4: [esp+6] 00000000 4: [esp+6] 000000000 4:		 6AA48908 6AA48900 6AA48900 6AA48900 	50 50 88 10		pu	ish 1 ish eax iv edx,dword	ptr ds:[ea	ax]			875W_C1 0 x875W_C0 0 875W_SF 0 x875W_P 1	
	dword ptr	<pre></pre>	dit.6BCD18E	30]=gpedit.6		11 dword pt	r ds:[edx+.	[4]	>	De 1: 2: 3:	[esp] 03583678 [esp+4] 00000001 [esp+8] 00000000	🗘 🗌 Unlock



GetRegistryKey is utilized to retrieve a handle to the root of the registry key for the computer section (0x2 = **GPO_SECTION_MACHINE**):



GA448923 50 GA448924 GA 02	push eax push 2	0,1 0,00,000 1,1 0,00 1,1 0,00 1,0 0,00 1,0 0,00 1,0 0,00 1,0 0,00 1,0 0,00 1,0 0,00 1,0 0,0 0	0 6 - 11 0 0
GAA49926 88 0A GAA49928 52 GAA49928 52 GAA49928 FF 51 3C C dword ptr [ccx+3C]=[gpedit.68CD18D8]=gpedit.6	mov ecx,dword ptr ds:[edx] push edx call dword ptr ds:[ecx+3C]	x875w_SF 0 x875w_F 1 x875w_U 0 → Default (stdcall) ▼ 5 € Unlock 1: [esp1 03583678 2: [esp14] 0000002	
.text:6AA48929 dump_patched.dll:\$8929 #8929	Imp 5 🛞 Watch 1 [x=] Locals 🦻 Struct	3: [esp+8] 00EDF864 4: [esp+C] 00000000 00EDF816 0358678 00EDF816 00000002	
Address Hex	ASCTT	00EDF820 00EDF864	

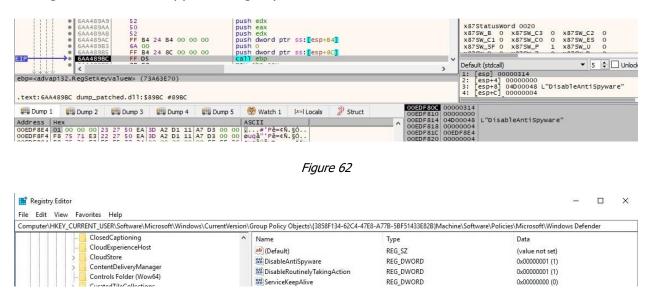
Figure 60

The malicious binary opens the "SOFTWARE\Policies\Microsoft\Windows Defender" registry key (0x3 = **KEY_QUERY_VALUE** | **KEY_SET_VALUE**):

EIP	GAA48959 GAA4895A GAA4895C GAA4895E GAA4895E GAA48962 GAA48966	50 6A 03 6A 00 FF 74 FF 74 FF D2		pi pi pi pi	ush eax ush 3 ush 0 ush dword ptr ush dword ptr all edx				~	x875w_LS 0 x875w_LC2 0 x875w_LC1 0 x875w_LC3 0 x875w_LE5 0 x875w_SF 0 x875w_LC3 0 x875w_LE5 0 x875w_SF 0 x875w_LC3 0 x875w_LE5 0 x875w_SF 0 x875w_LC3 0 x875w_LE5 0 variation 0 0 x875w_LC3 0 x875w_LC3 0 Default (stdcall) variation state state 0 0
	32.Reg0penKey									<pre>1: [esp] 00000318 2: [esp+4] 04E76728 L"SOFTWARE\\Policies\\Mi 3: [esp+8] 0000000 4: [esp+C] 00000003</pre>
.text:6AA48	966 dump_patc	hed.d11:\$8	966 #8966	Dump 5	👹 Watch 1	[x=] Locals	Struct	OOED	FS10 000	

Figure 61

The process turns off Microsoft Defender, as well as 3rd-party antivirus software and apps by setting the "DisableAntiSpyware" registry value to 1:





The Save method is used to save the specified registry policy settings to disk and update the revision number. The parameter called pGuidExtension is set to the GUID {35378eac-683f-11d2-a89a-00c04fbbcfa2} and pGuid is set to {3D271CFC-2BC6-4AC2-B633-3BDFF5BDAB2A}:



									01.	0100000	
312	 GAA489FB GAA489FC GAA489FE GAA489FF GAA4802 GAA48A03 GAA48A03 GAA48A07 GAA48A08 GAA48A08 GAA48A08 	50 33 C9 41 80 50 F0 52 88 75 00 51 51 55 55 FF 56 1C		oush eax sor ecx,ecx lea edx,dword ptr (bush edx nov esi,dword ptr s bush ecx bush ecx bush ebp all dword ptr ds:	ss: [ebp]			x87TW_6 3 x87Status x87SW_B x87SW_C1	(Empty)	0 x87SW_ES 1 x87SW_U	pty) pty) 0 0
dword ptr [<pre>@ < [esi+1C]=[gpedit</pre>	.6BCD18B8]=gpedit d.dll:\$8A09 #8A09	1.10		Sector 2		>	3: [esp+8		• 5	🗢 🗌 Unlock
Dump 1		Dump 3 🛛 💭 Dump 4	Ump 5	🛞 Watch 1 🛛 🕅 🖉	ocals 🖉 Struct	00ED	F810 0350 F814 0000 F818 0000	00001			
00EDF908 A0 00EDF918 F0	C 8E 37 35 3F 68 C 1C 27 3D C6 2B	D2 11 A8 9A 00 C C2 4A B6 33 3B D	0 4F BB CF A	2 ¬.75?hÒÀO»Ĭ¢ ù.'=∉+ÀJ¶3;BÕ½«*		OOED	F81C 00E	0F918			

Figure 64

The GPO object created earlier is deleted using the Delete method:

GAA48A60 50 GAA48A61 8B 00	push eax mov_eax,dword_ptr_ds:[eax]	x875W_SF 0 x875W_P 1 x875W_U 0
GAA48A63 FF 50 08	<pre>call dword ptr ds:[eax+8]</pre>	> Default (stdcall) 5
dword ptr [eax+8]=[gpedit.6BCD18A4]=gpedi .text:6AA48A63 dump_patched.dl]:\$8A63 #8A		2: [esp] 038530 2: [esp+8] 00000000 3: [esp+8] 00EDF82C &"PE" 4: [esp+C]_00C000E8 "PE"

Figure 65

Grief enumerates the executable files located in the System32 directory using the FindFirstFileExW routine (0x1 = FindExInfoBasic, $0x2 = FIND_FIRST_EX_LARGE_FETCH$):

Address H				00 56 50 00	ASCII UA.O.ÉØi	81			558 00E 55C 000	
Dump 1	Ump 2	Dump 3	Ump 4	Uump 5	🛞 Watch 1	[x=] Locals	Struct	OOEDFS	554 000	
	GAASGA8 GAASGA8 GAASGA8 GAASGA9 GAASGA9 GAASGA9 GAASGA9 GAASGA9 GAASGA9 GAASGA9 GAASGA9 GAASGA9 GAASGA9 GAASGA9 GAASGA9 GAASGA9 GAASGA9 GAASGA9 GAASGA8 GAASGA9 GAASG	52 F 52 0 8D 57 52 F 52 F 52 52 F 52 52 F 52 52 52 F 52 52 52 52 52 52 52 52 52 52 52 52 52 5	10 76ACDEA0)	p p p p c	ush ebx ush edx ush edx ea edx,dword ush edx ush eax ush dword pt all ecx	-	-		>	x8//W_65 (tmpty) x8//W_/ 3 (tmpty) x8/7statusWord 0020 x8/7sW_EB 0 x8/7sW_C3 0 x8/7sW_C2 0 x8/7sW_C1 0 x8/7sW_C3 0 x8/7sW_E5 0 x8/7sW_57 0 x8/7sW_P 1 x8/7sW_U 0 x8/7sW_57 0 x8/7sW_P 1 x8/7sW_U 0 Default (stdcall) v [5 \$] Unlo Default (stdcall) [5 \$] Unlo 1: [esp+4] 00000001 3: [esp+4] 00000001 3: [esp+6] 00EP594 4: [esp+C]_00000000

Figure 66

The process computes a "hash" (4-byte value) of each executable name using a custom function:



	🗾 🖆 🖼		1
	.text:6AA56978		
	.text:6AA56978		
	.text:6AA56978		
	.text:6AA56978 sub 6AA	56978 proc near	
	.text:6AA56978		
	.text:6AA56978 arg_0=	dword ptr 4	
	.text:6AA56978	an a	
	.text:6AA56978 push	ebx	
	.text:6AA56979 push	ebp	
	.text:6AA5697A mov	ebx, edx	
	.text:6AA5697C mov	edx, [esp+8+arg_0]	
	.text:6AA56980 xor	eax, eax	
	.text:6AA56982 dec	eax	
	.text:6AA56983 dec	edx	
	.text:6AA56984 jmp	short loc_6AA56999	
		20	
	V 1		7
	🗾 🚄 🖼		
	.text:6AA56999		
		A56999:	
	.text:6AA56999		
	.text:6AA56999 .text:6AA56999 loc_6A		5
	.text:6AA56999 .text:6AA56999 loc_6A/ .text:6AA56999 cmp	edx, 0FFFFFFFFh	5
	.text:6AA56999 .text:6AA56999 loc_6A/ .text:6AA56999 cmp	edx, 0FFFFFFFFh	
	.text:6AA56999 .text:6AA56999 loc_6A/ .text:6AA56999 cmp	edx, 0FFFFFFFFh	
■ ┏着 🖼 .text:6A556986	.text:6AA56999 .text:6AA56999 loc_6A/ .text:6AA56999 cmp	edx, 0FFFFFFFh short loc_6AA56986	•
.text:6AA56986	.text:6AA56999 .text:6AA56999 loc_6A/ .text:6AA56999 cmp .text:6AA5699C jnz	edx, 0FFFFFFh short loc_6AA56986	not eax
	.text:6AA56999 .text:6AA56999 loc_6A .text:6AA56999 cmp .text:6AA5699C jnz	edx, 0FFFFFFh short loc_6AA56986 .text:6AA5699E .text:6AA5699A	not eax pop ebp
.text:6AA56986 .text:6AA56986 loc_6A	.text:6AA56999 .text:6AA56999 loc_6A/ .text:6AA56999 cmp .text:6AA5699C jnz	edx, @FFFFFFh short loc_6AA56986 .text:6AA5699E .text:6AA56940 .text:6AA569A1	not eax pop ebp pop ebx
.text:6AA56986 .text:6AA56986 loc_6A .text:6AA56986 movzx .text:6AA56989 dec	.text:6AA56999 .text:6AA56999 loc_6A .text:6AA56990 gmp .text:6AA5699C jnz	edx, 0FFFFFFh short loc_6AA56986 .text:6AA5699E .text:6AA56940 .text:6AA569A1 .text:6AA569A2	not eax pop ebp pop ebx retn 4
.text:6AA56986 .text:6AA56986 loc_6A .text:6AA56986 movzx	.text:6AA56999 .text:6AA56999 loc_6AJ .text:6AA56999 cmp .text:6AA5699C jnz	edx, 0FFFFFFh short loc_6AA56986 .text:6AA5699E .text:6AA56940 .text:6AA569A1 .text:6AA569A2	not eax pop ebp pop ebx retn 4
<pre>.text:6AA56986 .text:6AA56986 loc_6A .text:6AA56986 movzx .text:6AA56989 dec .text:6AA5698A xor .text:6AA5698C inc</pre>	.text:6AA56999 .text:6AA56999 loc_6A/ .text:6AA56999 cmp .text:6AA5699C jnz .text:6AA5699C jnz .text:6AA5699C jnz .text:6AA5699C jnz .text:6AA5699C jnz .text:6AA5699C jnz .text:6AA5699C jnz .text:6AA5699P jnz .text:6AA569P jnz .text:6AA5	edx, 0FFFFFFh short loc_6AA56986 .text:6AA5699E .text:6AA569A0 .text:6AA569A1 .text:6AA569A2 .text:6AA569A2	not eax pop ebp pop ebx
.text:6AA56986 .text:6AA56986 loc_6A .text:6AA56986 movzx .text:6AA56980 dec .text:6AA56980 xor .text:6AA5698C inc .text:6AA5698D and	.text:6AA56999 .text:6AA56999 loc_6A .text:6AA56999 cmp .text:6AA5699C jnz .text:6AA5699C jnz	edx, 0FFFFFFh short loc_6AA56986 .text:6AA5699E .text:6AA569A0 .text:6AA569A1 .text:6AA569A2 .text:6AA569A2	not eax pop ebp pop ebx retn 4
<pre>.text:6AA56986 .text:6AA56986 loc_6A .text:6AA56986 movzx .text:6AA56989 dec .text:6AA5698A xor .text:6AA5698C inc</pre>	.text:6AA56999 .text:6AA56999 loc_6A/ .text:6AA56999 cmp .text:6AA5699C jnz .text:6AA5699C jnz .text:6AA5699C jnz .text:6AA5699C jnz .text:6AA5699C jnz .text:6AA5699C jnz .text:6AA5699C jnz .text:6AA5699P jnz .text:6AA569P jnz .text:6AA5	edx, 0FFFFFFh short loc_6AA56986 .text:6AA5699E .text:6AA569A0 .text:6AA569A1 .text:6AA569A2 .text:6AA569A2	not eax pop ebp pop ebx retn 4

Figure 67

The hash value is XOR-ed with 0x84794EF2, and then compared with 0x668B9032 (hard-coded value). Whether the two values aren't equal, the malware continues the enumeration by calling the FindNextFileW API:

L	 6AA5 686 6AA5 686 	B FF 36			ush ebp ush dword pt	tr ds:[esi]				x875W_SF 0 x875W_P	1 x87SW_U	0
EIP	GAA5 6B6				all eax			>		Default (stdcall) 1: [esp] 03338AC8	▼ 5	🗘 🗌 Unlock
	132.FindNext 6B6D dump_pa								23	1: [esp] 03338848 2: [esp+4] 00EDF594 3: [esp+8] 00EDF560 4: [esp+C]_00EDF5C0	L"appidtel.exe"	
💭 Dump 1	Dump 2	Dump 3	Dump 4	Dump 5	🥘 Watch 1	[x=] Locals	Struct	00EDF564 03 00EDF568 00				

Figure 68

The binary is looking for vssadmin.exe. It disables file system redirection for the current thread using Wow64DisableWow64FsRedirection:

EIP	6AA517F4 6AA517F5	FF DO	call eax		~	lt (stdcall)	▼ 5 €	Unlock
		Wow64FsRedirection			2: [esp+8] 6AA592E3	L"C:\\WINDOWS\\syst dump_patched.6AA592 L"C:\\WINDOWS\\syst	E3
-	 		· · · · · · · · · · · · · · · · ·	00EDF770 00	DEDF7A4			

Figure 69

The ransomware deletes all Volume Shadow Copies using vssadmin (0x08000000 = **CREATE_NO_WINDOW**):



GAAS92F8 57 GAAS92F9 FF 74 24 4C GAAS92F9 FF 74 24 4C GAAS9303 50 GAAS9303 50 GAAS9304 53 GAAS9305 GA 01 GAAS9305 FF 74 24 20 GAAS9306 FF 74 24 20 GAAS9307 FF 75 75 75 75 75 75 75 75 75 75 75 75 75	<pre>push edi push dword ptr ss: esp+4Cl xor eax,eax push eax push dword ptr ss:[esp+20] push eap call eax </pre>	x87TW_0 3 (Empty) x87TW_1 3 (Empty) x87TW_2 3 (Empty) x87TW_3 3 (Empty) x87TW_4 3 (Empty) x87TW_5 3 (Empty) x87TW_6 3 (Empty) x87TW_7 3 (Empty) x87TW_6 3 (Empty) x87TW_7 3 (Empty) x87TW_6 3 (Empty) x87TW_7 3 (Empty) x87StatusWord 0020 x87SW_C1 0 x87SW_C2 0 x87SW_5 0 x87SW_C0 0 x87SW_E5 0 x87SW_5 0 x87SW_7 1 x87SW_0 Default (stdcall) v
edx= <kernel32.createprocessw> (76A74510) .text:6AA5930E dump_patched.dll:\$1930E #1930E ## Dump 1 ## Dump 2 ## Dump 4 ## Address Hex 00EPF744 <u>A858 E7 04 F8 93 A5 68 00 00 00 00 00 00 00 00 00 00 00 00 00</u></kernel32.createprocessw>	Dump5 Watch 1 Impose 00EDF758 04E75988 L"C:\\ AscII ∧ 00EDF75C 0000000 00EDF76C 0000000 0 00 00 X; Li ¥] 00EDF76C 0000000 00EDF76C 0000000 0 00 00 C+1.D 00EDF76C 0000000 0000000 00EDF76C 00000000 0 00 00 C+1.D 00EDF76C 00000000 0000000 0000000	<pre>XII [esp] 042758A8 L"C:\\WINDOWS\\system32\\vs 2: [esp+4] 0475988 L"C:\\WINDOWS\\system32\\ 3: [esp+4] 0475988 L"C:\\WINDOWS\\system32\\ 4: [esp+6] 0000000 4: [esp+6] 0000000 WINDOWS\\system32\\vssadmin.exe" WINDOWS\\system32\\vssadmin.exe Delete Shadows /All /Quiet"</pre>

Figure 70

The process restores file system redirection for the current thread via a function call to Wow64RevertWow64FsRedirection:

● 6AA51814 ● 6AA51817	FF 75 00	push dword ptr ss:[ebp]		
Land CAASISI	FF DU	Cdl edx	>	Default (stdcall) 🔻 5 🖨 🗌 Unlock
<pre>eax=<kernel32.wow64revert .text:6aa51817="" dump_patch<="" pre=""></kernel32.wow64revert></pre>				<pre>1: [esp] 00EDF3F0 2: [esp+4] 04E758A8 L"C:\\WINDOWS\\system32\\ 3: [esp+8] 6AA59328 dump_patched.6AA59328 4: [esp+C] 04E75988 L"C:\\WINDOWS\\system32\\</pre>
the second se			00EDF770 00E	DF3F0

Figure 71

The process of enumerating the executable files from the System32 folder is repeated one more time, and the XOR-ed result is compared with 0x96164682 (hard-coded value). Based on our analysis, the targeted file is Diskshadow.exe.

CryptGenRandom is utilized to generate 4 random bytes 3 times:

Address He	N.				ASCIT			00EDF790	OOEE	DF7A8	
Ump 1	Dump 2	Ump 3	Ump 4	🚛 Dump 5	🥘 Watch 1	[x=] Locals	Struct	00EDF788 00EDF78C	0000	00004	
eax= <advapi< td=""><td>32.CryptGen</td><td>Random> (73/</td><td></td><td></td><td>dii eax</td><td></td><td></td><td>3</td><td>></td><td>Default (stdcall) 1: [esp] 03345768 <&CP/ 2: [esp+4] 0000004 3: [esp+8] 00EDF7A8 4: [esp+C] 03345768 <&d</td><td>▼ 5 ÷ Unlo AcquireContext> CPAcquireContext></td></advapi<>	32.CryptGen	Random> (73/			dii eax			3	>	Default (stdcall) 1: [esp] 03345768 <&CP/ 2: [esp+4] 0000004 3: [esp+8] 00EDF7A8 4: [esp+C] 03345768 <&d	▼ 5 ÷ Unlo AcquireContext> CPAcquireContext>
	● 6AA561E ● 6AA561E ● 6AA561E ● 6AA561E	4 57 5 55		p	ush esi ush edi ush ebp all eax					x87SW_C1 0 x87SW_C0 x87SW_SF 0 x87SW_P	0 x87SW_ES 0 1 x87SW_U 0

Figure 72

The binary creates an empty temporary file with a prefix string generated based on the random bytes:

text: 6AA5		ed.dll:\$10DCF #10DC		00EDF7CC 0	04E75930 L"C:\\PROGRA~3\\"
text: 6AA5	DCF dump_patch	ed.dll:\$10DCF #10DC	F		
IP edi= <kerne< th=""><th>GAASODCA GAASODCE GAASODCE CAASODCE CAASODCE</th><th>FF 74 24 2C 56 FF D7 NameW> (76ACE0A0)</th><th>push dword ptr ss:[esp+2C] push esi call edi</th><th>></th><th>x875w.SF 0 x875w.P 1 x875w.U 0 Default (stdcal) ▼ 5 ↓ Unloc 1: [Esp] 04575930 L"C:\\PROGRA-3\\" 2: esp+4] 0457560 "30" 3: [Esp+4] 04575940 6500 "30" 3" esp+8] 0000000 4: [Esp+6] 04570408 8" 6" 6" 1"</th></kerne<>	GAASODCA GAASODCE GAASODCE CAASODCE CAASODCE	FF 74 24 2C 56 FF D7 NameW> (76ACE0A0)	push dword ptr ss:[esp+2C] push esi call edi	>	x875w.SF 0 x875w.P 1 x875w.U 0 Default (stdcal) ▼ 5 ↓ Unloc 1: [Esp] 04575930 L"C:\\PROGRA-3\\" 2: esp+4] 0457560 "30" 3: [Esp+4] 04575940 6500 "30" 3" esp+8] 0000000 4: [Esp+6] 04570408 8" 6" 6" 1"
	 6AA5 0DC5 6AA5 0DC8 	FF 34 24 6A 00	push dword ptr ss:[esp] push 0		x87SW_B 0 x87SW_C3 0 x87SW_C2 0 x87SW_C1 0 x87SW_C0 0 x87SW_ES 0





The malware retrieves the short path form of the specified path by calling the GetShortPathNameW routine:

.text:6AA5		🖉 Dump 3 🛛 🖉 🖉 D	Dump 4 🛛 Dump 5	🛞 Watch 1	[x=] Locals	2 Struct	00EDF7D 00EDF7D	4 04E	7D9E0
text: 6AA5				00	1. 1. 1	(3)	00EDE7D	0 04E	7D408 L"C:\\PROGRA~3\\3Q2C8C.tmp"
	132.GetShortPat 0E7B dump_patch	-							1: [esp] 04E7D408 L"C:\\PROGRA~3\\3Q2C8C.tm 2: [esp+4] 04E7D9E0 3: [esp+8] 00000100 4: [esp+C]_04E7D408 L"C:\\PROGRA~3\\3Q2C8C.
IP	GAA50E7B	FF DO	C	all eax				>	Default (stdcall) 🔻 5 🖨 🗌 Unlo
	 GAASOE74 GAASOE75 GAASOE77 	52 FF 33 FF 74 24 08	pi	ush edx ush dword pt ush dword pt	r ds:[ebx]	81			x87SW_C1 0 x87SW_C0 0 x87SW_ES 0 x87SW_SF 0 x87SW_P 1 x87SW_U 0



Grief ransomware opens the newly created file using CreateFileW (0xC0000000 = GENERIC_READ | GENERIC_WRITE, 0x5 = TRUNCATE_EXISTING, 0x80 = FILE_ATTRIBUTE_NORMAL):

	GAA5262C GAA5263C GAA5262C GAA5262C GAA5262C GAA5262C GAA5262C GAA5262C GAA5262C GAA5265C GAA526C GAA	ew> (76ACDD	EO)	, p p p p p p p p p	ush eax ush dword pt ush ebp ush eax ush ebx ush esi ush dword pt all edx			 >	×8 ×8 ×8 ×8	[esp+8] 00000	020 5W_C3 (0 5W_C0 (0 5W_P 1 	x875₩_ES x875₩_U	0 0 0
Ump 1	📖 Dump 2	Ump 3	💭 Dump 4	Dump 5	🛞 Watch 1	[x=] Locals	🖉 Struct	 00EDF7C4	000000		A~3\\3Q	2C8C.tmp"	
Address He	ex				ASCII			00EDF7C8 0 00EDF7CC 0					
	00 3A 00 5C							OOEDF7D0					
	L 00 7E 00 33 3 00 43 00 2E							00EDF7D4					
								00EDF7D8 0					

Figure 75

The malware calls the SetFileTime function in order to prevent file operations using the file handle from modifying the last access time and the last write time (dwLowDateTime and dwHighDateTime are set to 0xFFFFFFF):

Ump 1 Dump 2	🗒 Dump 3 🛛 Dump 4	Uump 5 😸 Watch 1	[x=] Locals 🐉 Struct	00EDF7CC 000 00EDF7D0 000 00EDF7D4 00E 00EDF7D8 00E	000000 EDF7E4
GAA52684 GAA52685 GAA52685 GAA52688 GAA52688 GAA52688 GAA52688 GAA52688 GAA52684 GAA52684 GAA52684 GAA52684 GAA52684 Compate		push ebp push ebp push 0 push dword ptr call eax	ds:[ebx]	×	x875w_B 0 x875w_C3 0 x875w_C2 0 x875w_C1 0 x875w_C0 0 x875w_E5 0 x875w_5F 0 x875w_P 1 x875w_U 0 Default (stdcall)

Figure 76

The file is populated with the following content:



										01 m	0100000 Q1	0100000				11 0 0
EIP	GAAS 295 GAAS 295 GAAS 295 GAAS 296 GAAS 296 GAAS 296 GAAS 296	C 50 D FF 74 1 FF 74 5 FF 33 7 FF D2	24 08 24 1C		push 0 push eax push dword pt push dword pt push dword pt call edx	tr ss:[esp+ tr ss:[esp+ tr ds:[ebx]	5) LC		~	x8 x8 x8 Def	7 SW_B 7 SW_C 1 7 SW_SF 2 SW_SF ault (stdc		3 0 0 0	x875W_ x875W_	ES O] Unlock
	132.WriteFil 2967 dump_pa									1: 2: 3: 4:	[esp+4 [esp+8	000002FC 04E76728 0000001A 00EDF80C		ete sha	dows all	\r\nex
Dump 1	Dump 2	Ump 3	Ump 4	Dump 5	🛞 Watch 1	[x=] Locals	Struct	00EDF7	8 000 FC 04E	002F	C 8 "del	ete shadow	s all	\r\nexi	t\r\n"	
Address He		4 65 20 73	68 61 64 6F	77 73 20 6	ASCII 1 delete shad	dows a		^ 00EDF8 00EDF8 00EDF8	00 000 04 00E	0001 DF80	A C			de di.	18 - 18 ⁶	

Figure 77

The ransomware deletes all Volume Shadow Copies by creating a Diskshadow process and then running the "delete shadows all" command:

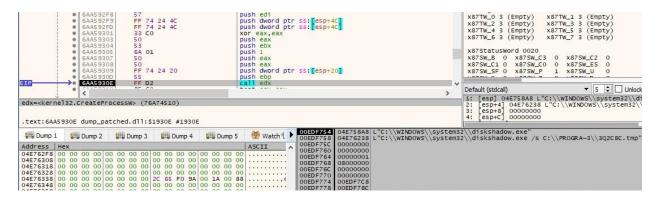


Figure 78

The malicious file retrieves the share names available on the local computer using the NetShareEnum API:

esi= <srvcli< th=""><th></th><th>52 51 59 6A FF 55 55 53 53 53 53</th><th>10)</th><th>p p p p</th><th>ush eax ush edx ush ecx ush FFFFFFF ush ebp ush ebx ush ebx all esi</th><th></th><th>o.* o.</th><th> ></th><th>x875t x875w x875w x875w x875w x875w Default 1: [e: 2: [e: 3: [e:</th><th>atusW _B 0 _C1 0 _SF 0 [stdcall] [stdcall] [sp] 00 [sp+4] [sp+8]</th><th>ord 0020 x875W_C3 x875W_C0 x875W_P</th><th>0</th><th>w_/ 3 (Emp x875W_C2 x875W_E5 x875W_U</th><th>0</th></srvcli<>		52 51 59 6A FF 55 55 53 53 53 53	10)	p p p p	ush eax ush edx ush ecx ush FFFFFFF ush ebp ush ebx ush ebx all esi		o. * o.	 >	x875t x875w x875w x875w x875w x875w Default 1: [e: 2: [e: 3: [e:	atusW _B 0 _C1 0 _SF 0 [stdcall] [stdcall] [sp] 00 [sp+4] [sp+8]	ord 0020 x875W_C3 x875W_C0 x875W_P	0	w_/ 3 (Emp x875W_C2 x875W_E5 x875W_U	0
Ump 1	Ump 2	🚛 Dump 3	📖 Dump 4	🚛 Dump 5	👹 Watch 1	[x=] Locals	2 Struct	00EDF808 00						
Address He	ex				ASCII			00EDF810 00						
OOEDF8A8 FE	FF FF FF F	00 00 00 00	20 34 C3 75	B8 CE 32 03	þ <mark>ÿÿÿ</mark> üøi. 4Å .@zt	u 12. 		00EDF814 FF 00EDF818 00 00EDF81C 00 00EDF820 00	EDF8A8 EDF928					

Figure 79



REFERENCES

- 1. <u>https://www.crowdstrike.com/blog/doppelpaymer-ransomware-and-dridex-2/</u>
- 2. https://cyber-anubis.github.io/malware%20analysis/dridex/
- 3. https://github.com/mandiant/capa
- 4. <u>https://stackoverflow.com/questions/18039401/how-can-i-transform-between-the-two-styles-of-public-key-format-one-begin-rsa</u>
- 5. <u>https://blog.malwarebytes.com/threat-analysis/2018/01/a-coin-miner-with-a-heavens-gate/</u>



APPENDIX

The other strings decrypted using the RC4 algorithm are shown in the following pictures:

Address	He	¢															ASCII
04BD0048	39	63	34	66	61	34	61	3B	66	31	39	37	39	39	31	64	9c4fa4a;f197991d
04BD0058	3B	63	35	62	66	64	66	35	34	3B	32	63	36	35	65	33	;c5bfdf54;2c65e3
04BD0068	63	31	3B	33	39	38	31	61	30	64	34	3B	35	66	63	38	c1;3981a0d4;5fc8
04BD0078	34	39	65	37	3B	61	65	35	61	32	32	62	34	3B	36	32	49e7; ae5 a22b4; 62
04BD0088	37	34	66	61	36	34	3B	66	36	32	35	32	36	62	39	3B	74fa64;f62526b9;
04BD0098	64	63	34	30	61	64	62	61	3B	34	31	30	37	61	61	37	dc40adba; 4107aa7
04BD00A8	36	3B	62	66	64	66	35	32	39	65	3B	33	30	36	64	35	6; bfdf529e; 306d5
04BD00B8	31	61	30	3B	66	31	63	63	32	33	65	34	3B	31	38	66	1a0;f1cc23e4;18f
04BD00C8	33	62	39	39	31	3B	34	62	38	63	66	62	61	3B	36	63	3b991;4b8cfba;6c
04BD00D8	63	65	66	36	62	62	3B	32	35	61	66	64	34	33	3B	31	cef6bb;25afd43;1
04BD00E8	61	33	35	39	30	32	61	3B	36	64	34	37	38	61	61	66	a35902a;6d478aaf
04BD00F8	3B	34	39	64	39	31	36	32	35	3B	34	36	39	63	38	66	;49d91625;469c8f
04BD0108	65	30	3B	31	66	38	39	36	62	64	62	3B	32	65	33	30	e0;1f896bdb;2e30
04BD0118	64	37	34	35	3B	37	35	62	32	64	39	37	31	3B	39	37	d745;75b2d971;97
04BD0128	62	37	35	31	62	33	00	00	00	00	00	00	00	00	00	00	b751b3

Figure 80

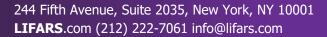
Address	He	¢															ASCII
04BD0048	36	64	34	37	38	61	61	66	3B	65	62	30	31	32	62	30	6d478aaf;eb012b0
04BD0058	30	3B	36	36	32	62	33	61	36	37	3B	38	33	65	64	39	0;662b3a67;83ed9
04BD0068	38	61	33	3B	39	31	65	35	33	66	38	66	3B	34	62	37	8a3;91e53f8f;4b7
04BD0078	61	66	36	66	36	3B	63	65	38	63	35	36	62	37	3B	39	af6f6;ce8c56b7;9
04BD0088	63	34	66	61	34						37				64		c4fa4a;f197991d;
04BD0098	63	35	62	66		66			3B								c5bfdf54;2c65e3c
04BD00A8	31	3B	33	39	38	31	61	30		34		35		63			
04BD00B8	39	65	37	3B			66				61						
04BD00C8	63	38			30												
048D00D8	63	34	38	64	64	34	35	3B			32			61			c48dd45;8a27aa19
04BD00E8	3B	37	35	61	39	33	33		36				31		64	34	
04BD00F8	37	65	3B	35		30	33	34	61				63		33	66	7e;5a034a5a;c13f
04BD0108	39	_	63		3B						37						90ca;269fc73;75e
04BD0118	31	32	65	39	65	3B	32	32	38	32	61	64		38			12e9e;2282ad08;4
04BD0128	61	38	32	65	64	30	61	3B			64			65			a82ed0a;c6d2ce67
04BD0138	3B	34	37	39	35	36	31	65	3B	33	64		63	62	61	35	;479561e;3d8cba5
04BD0148	3B	63	61	62	33	32	62	65	37		39			37	33	31	;cab32be7;9e0731
04BD0158	39				34											37	94;c43897e8;17f7
04BD0168	66	36	65	63	3B	63	32	64	34	32	33	62	33	3B	62	30	f6ec;c2d423b3;b0

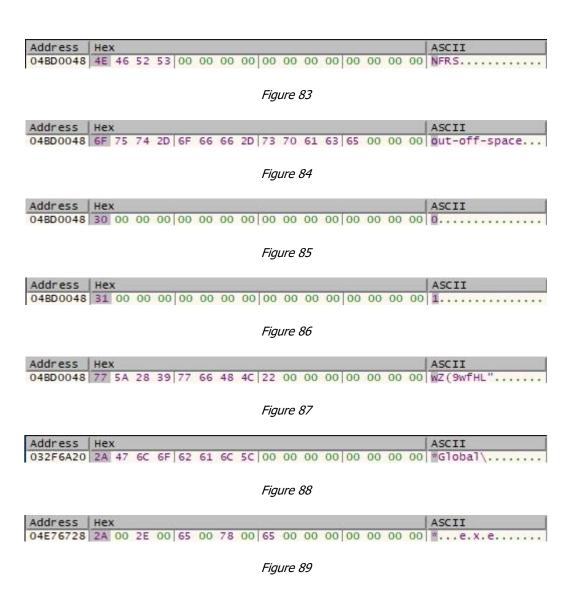
Figure 81

Address	He	¢															ASCII
04BD0048	31	61	32	31	32	34	63	30	3B	32	30	66	33	61	61	61	1a2124c0;20f3aaa
04BD0058	35	3B	34	35	36	62	31	30	39	66	3B	38	39	35	61	62	5;456b109f;895ab
04BD0068	64	37	33	3B	34	39	64	39	31	36	32	35	3B	36	36	32	d73;49d91625;662
04BD0078	62	33	61	36	37	3B	38	33	65	64	39	38	61	33	3B	32	b3a67;83ed98a3;2
04BD0088	64	33	34	36	31	64	30	3B	38	62	66	38	64	61	31	36	d3461d0;8bf8da16
04BD0098	3B	65	30	39	33	37	31	34	35	3B	32	66	62	61	33	37	;e0937145;2fba37
04BD00A8	30	36	3B	34	36	39	33	66	30	31	34	3B	32	63	36	35	06;4693f014;2c65
04BD00B8	65	33	63	31	3B	33	39	38	31	61	30	64	34	3B	34	34	e3c1;3981a0d4;44
04BD00C8	35	37	61	64	31	36	38	66	35	38	64	65	30	36	30	3B	57ad16;f58de060;
04BD00D8	63	35	39	39	31	65	39	62	3B	62	64	61	66	66	30	63	c5991e9b;bdaff0c
04BD00E8	3B	31	64	36	33	37	33	36	62	3B	39	35	66	36	63	64	;1d63736b;95f6cd
04BD00F8	37	35	3B	62	33	32	37	39	63	34	35	3B	38	65	30	61	75;b3279c45;8e0a
04BD0108	36	32	66	31	3B	34	61	36	65	34	32	62	36	3B	65	31	62f1;4a6e42b6;e1
04BD0118	64	31	32	34	66	61	3B	33	33	63	38	30	65	33	3B	36	d124fa; 33c80e3; 6
04BD0128	64	36	64	34	62	61	35	3B	35	33	63	63	61	39	65	30	d6d4ba5;53cca9e0
04BD0138	3B	33	64	65	34	66	39	34	39	3B	65	36	39	37	37	35	;3de4f949;e69775
04BD0148	35	32	3B	62	62	62	62	61	35	65	36	3B	32	36	39	66	52;bbbba5e6;269f
04BD0158	63	37	33	3B	31	37	66	37	66	36	65	63	3B	61	65	39	c73;17f7f6ec;ae9
04BD0168	30	37	62	30	34	3B	32	37	30	34	35	32	66	62	3B	65	07b04;270452fb;e

Figure 82

your digital world, secured





List of files and file's extensions to be skipped:

 svsho*.exe;schre*.bat;V01.lo*;V01.ch*;V01res*.jrs;RacWmi*.sdf;Web*V01.dat;default.r dp;NTUSER.DA*;*.lnk;*.ico;*.ini;*.msi;*.chm;*.sys;*.hlf;*.lng;*.inf;*.ttf;*.cmd;*.LNK;* .ICO;*.INI;*.MSI;*.CHM;*.SYS;*.HLF;*.LNG;*.INF;*.TTF;*.CMD

List of directories to be skipped:

• System Volume Information; \$RECYCLE.BIN; \$Recycle.Bin; WebCache; Caches; VirtualStore



List of environment-variable strings:

%ProgramData%\\Microsoft\\Windows\\WER\\ReportQueue\\;%windir%;%temp%;%A
PPDATA%\\Local\\VirtualStore\\;%HOMEDRIVE%\\Documents and Settings\\All
Users\\Application Data\\Application Data\\;%HOMEDRIVE%\\Users\\All
Users\\Application Data\\Application Data\\;%SystemDrive%\\Documents and
Settings\\All Users\\Application Data\\Application Data\\;%SystemDrive%\\Users\\All
Users\\Application Data\\Application Data\\;%SystemDrive%\\Users\\All
Users\\Application Data\\Application Data\\;%SystemDrive%\\Users\\All
Users\\Application Data\\Application Data\\;%SystemDrive%\\Users\\All
Users\\Application Data\\

List of services to be stopped:

- msolap\$*;mssql\$*;sqlagent\$*;reportserver\$*
- sophos client firewall*;sophos mcs*;sophos web intelligence*;sophospatch*

Extension of encrypted files:

• .pay0rgrief

Grief's Dark web site and the impacted client:

- "CROMOLOGY SERVICES ... ZOLPAN, you are fu**ed." (Redacted)
- "DO NOT TOUCH ANYTHING!\r\n\r\nWhat to do (password: oN********):\r\nhttp[:]//payorgz3j6hs2gj66nk6omfw65atgmqwzxqbbxnqi3bv2mlwgcirunad[.]onion/ demand/da597c8432bc4458b9475627fd55eded\r\n\r\nUSE TOR.\r\n\r\nP0G_\r\n\r\n" (Redacted)



API hashing table

		•	
A169D93E	ExitProcess	7408F6CF	RegDeleteValueA
B7303F40	GetCurrentDirectoryW	CB74E56B	RegSetValueExA
F1E04D0E	CreateDirectoryW	3440E30C	RegQueryValueExA
14134842	CreateThread	3FA0503A	RegSetValueExW
D8FC22B5	CreateProcessW	C094565D	RegQueryValueExW
C4B669CF	CreateFileMappingW	8D388F19	RegEnumValueA
1D4786C2	QueryDosDeviceW	2D504FC7	RegCloseKey
2CE276DD	MapViewOfFile		RegOpenKeyExW
	-	B1978170	01 3
BD63F85D	UnmapViewOfFile	49A2BC02	RegEnumKeyW
589C7CD4	GetFileType	2478983B	RegCreateKeyExW
2596A7DB	CreateFileW	2C39743C	CryptReleaseContext
D7509C5D	GetVolumeNameForVolumeMountPointW	826FDC1D	CryptGetHashParam
8EB1B560	DeviceIoControl	429ACFE2	CryptHashData
22C3F66E	ExpandEnvironmentStringsA	5B40E61E	CryptCreateHash
78120C03	GetModuleFileNameW	D8EFD506	CryptAcquireContextW
ECED49A4	FileTimeToSystemTime	8E1D8F12	CryptDestroyHash
BC8CDE49	SystemTimeToFileTime	53F5694D	CryptGenRandom
A88A7EA6	GetShortPathNameW	D4E43A30	CryptEncrypt
F5656839	GetLogicalDrives	DE78F152	CryptExportKey
86089CF3	GetDriveTypeW	6F75B3F1	
			CryptGenKey
D9DE4146	SetThreadPriority	69836B71	CryptDestroyKey
E66CC345	GetDiskFreeSpaceExW	8A2AACA0	SetSecurityInfo
65C66CA1	SetFileAttributesW	7EBEE13C	GetSecurityDescriptorSacl
AE320B72	MoveFileW	7F0B03AE	ConvertStringSecurityDescriptorToSecurityDescriptorW
F68850CB	MultiByteToWideChar	8B6FA607	ControlService
1EF9AB7B	WideCharToMultiByte	3373DF6A	OpenServiceW
AF2A8DE9	GetVersionExW	2EE029FE	StartServiceCtrlDispatcherW
F246E304	GetSystemInfo	F40C812D	CloseServiceHandle
BA71B979	LocalFree	F66A15F1	OpenSCManagerW
AA297AF9	IsWow64Process	9675A67D	ChangeServiceConfigW
C0ED06A6	GetSystemWow64DirectoryW	5CDDF47	StartServiceW
F61D52F9	GetSystemDirectoryW		
		3F1483A7	QueryServiceConfigW
459F8107	GetEnvironmentStringsW	42132256	QueryServiceStatus
9224D8AB	GetTempFileNameW	26652D0D	EnumServicesStatusExW
8F5E891D	GetWindowsDirectoryW	E8C5D221	SetServiceStatus
CA2E3F55	GetComputerNameW	518E8878	RegisterServiceCtrlHandlerW
5DCB4A66	GetCommandLineW	29DBE130	GetUserNameW
D5D107B9	IsBadReadPtr	787BAFBC	GetSidSubAuthority
5321A741	GetThreadId	F2EC9F3E	GetSidSubAuthorityCount
1F442F52	GetProcessId	922CE64F	GetTokenInformation
99CD5D11	GetCurrentProcessId	7DBF48E7	OpenProcessToken
72A2E993	SearchPathW	B514674F	FreeSid
4FCE620F	Wow64DisableWow64FsRedirection	9C01B84F	ConvertSidToStringSidA
84A5D7E5	Wow64RevertWow64FsRedirection		EqualSid
		D46EE9FF	
569C7845	GetLastError	434A3624	AllocateAndInitializeSid
A5F904F1	SetFileTime	967918CF	RegSetKeyValueW
6BBEA486	SetFilePointer	62DE91AE	CreateProcessAsUserW
23820F97	GetFileSize	65E4543B	NetUserEnum
8D254D22	ReadFile	0BCF31C0	NetUserSetInfo
489018E0	WriteFile	53FF883	NetShareEnum
7E44617A	FlushFileBuffers	4266BEEF	NetApiBufferFree
A160FFA8	SetEndOfFile	F5EE9951	NetShareDel
BBD6B3B8	GetFileTime	A633633A	CryptStringToBinaryA
2236F20A	GetFileAttributesExW	C380FA58	CryptDecodeObject
4348FE4D	RemoveDirectoryW	2F9F0714	CryptImportPublicKeyInfo
E1369068	DeleteFileW	DD2C7E1F	WTSEnumerateSessionsW
3E4FB2EF	GetHandleInformation	7809AAC1	WTSQueryUserToken
	QueryFullProcessImageNameW		
784487EE		71A22286	WTSFreeMemory
7D5DB015	GetProcessTimes	0B910B2	ZwClose
98B31D0F	GetExitCodeProcess	4D62C13	RtlExitUserThread
9F5CDB	LocalSize	8631D459	GetClassNameW
565B4A16	GetSystemTime		
5C52B868	FindClose		
7380D608	FindFirstFileExW		
58AD2EB	FindNextFileW		
5982AEC6	SetLastError		
BBB8F37F	LoadLibraryA		
5E116D7D	FreeLibrary GetProcAddress		
4D05510D	GELETUCAUULESS		

