

A Deep Dive into The Grief Ransomware's Capabilities

Prepared by: LIFARS, LLC
Date: 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 ".pay0rgrief" 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":

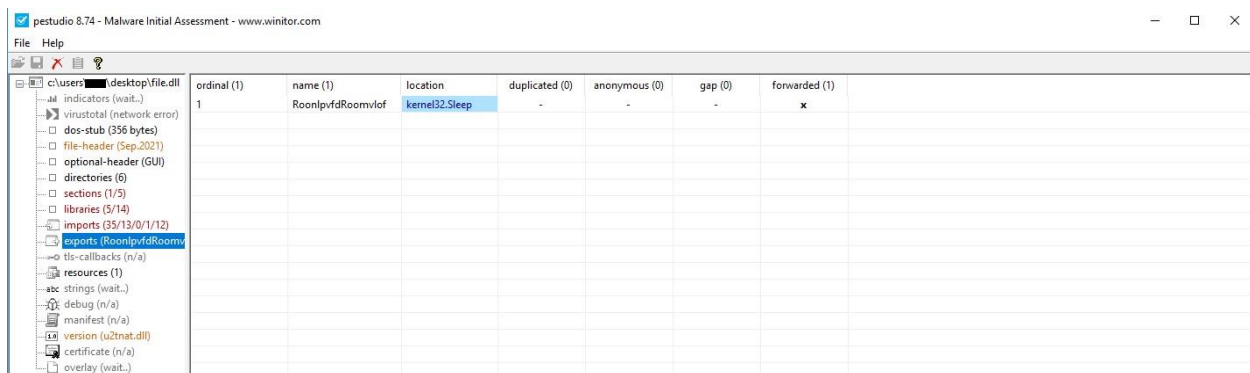


Figure 1

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

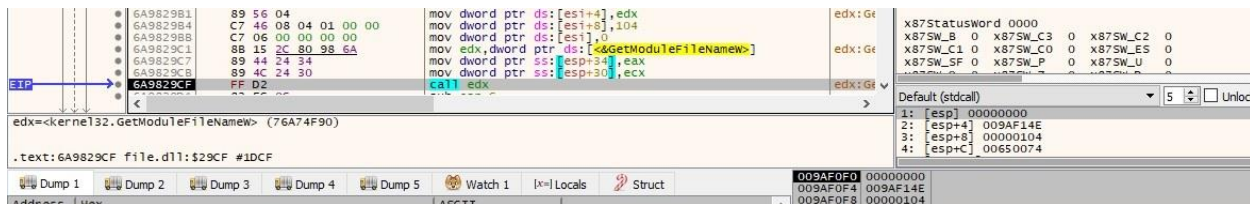


Figure 2

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

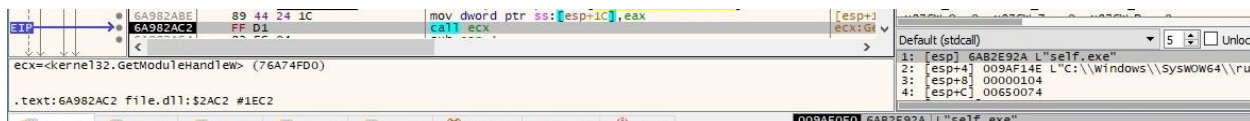


Figure 3

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

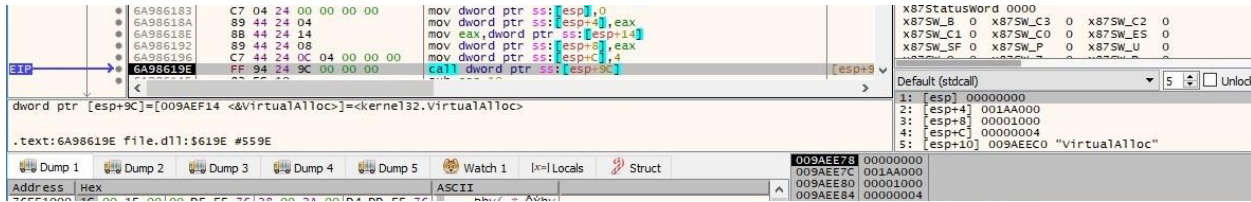


Figure 4

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:

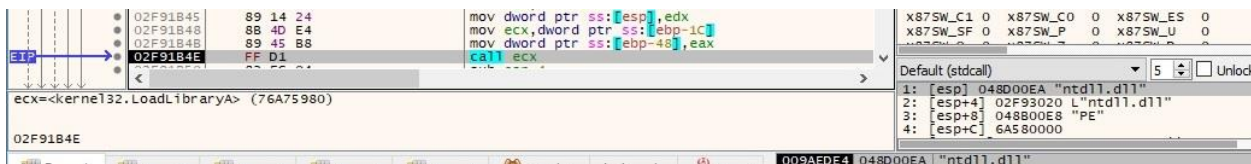


Figure 5

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

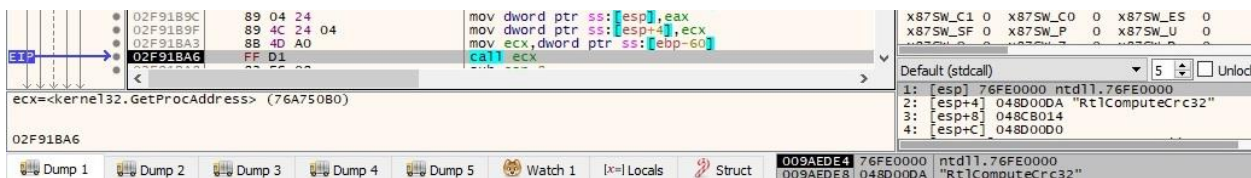


Figure 6

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

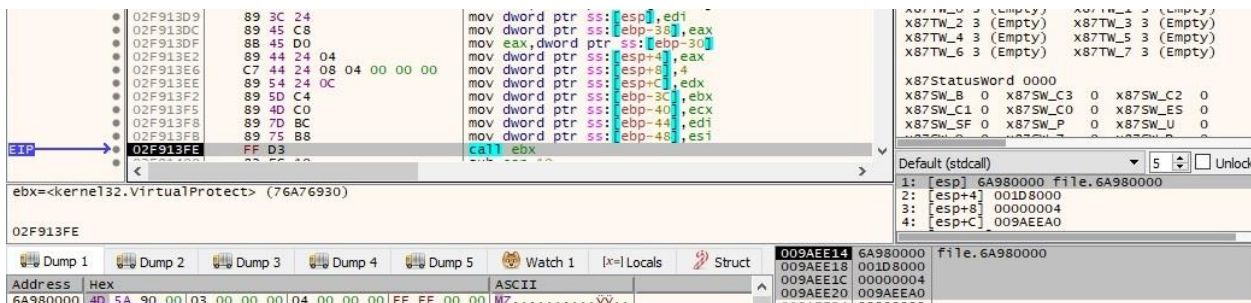


Figure 7

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**):

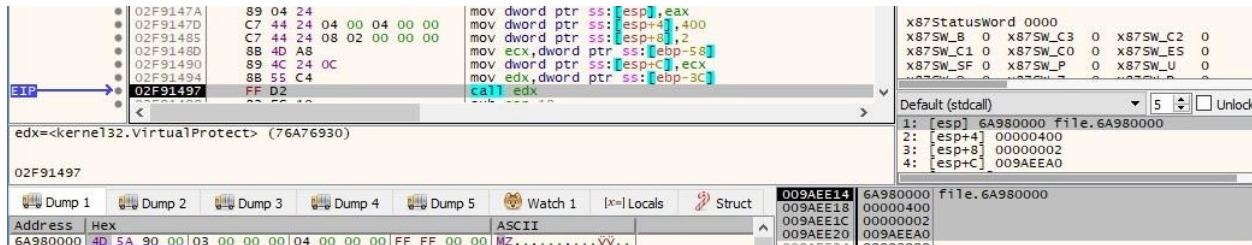


Figure 8

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

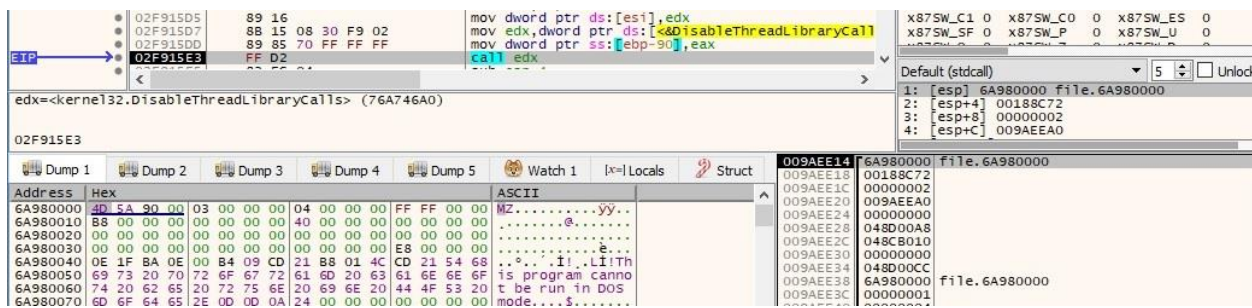


Figure 9

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:

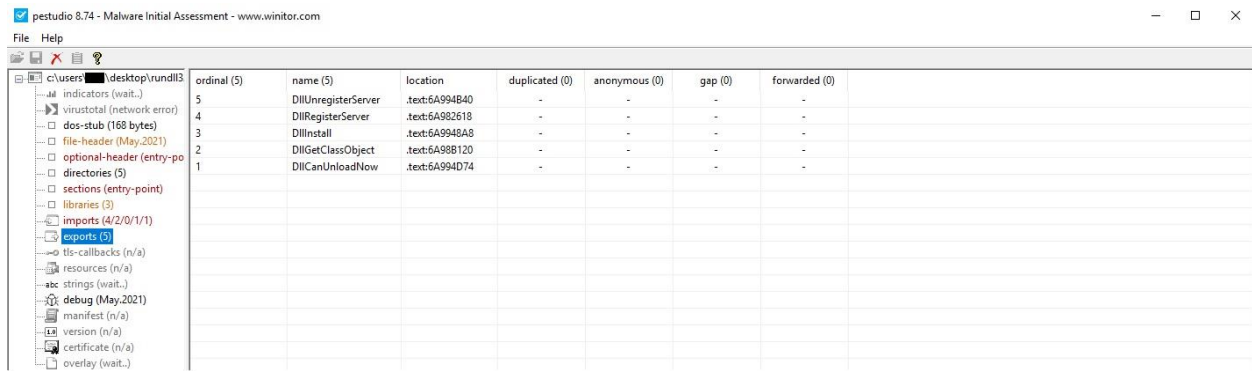


Figure 10

An important hint which suggests that the file is encrypted/obfuscated is the lack of imports: GetCommandLineW, lstrcpw, CommandLineToArgvW, and RtlComputeCrc32. 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):

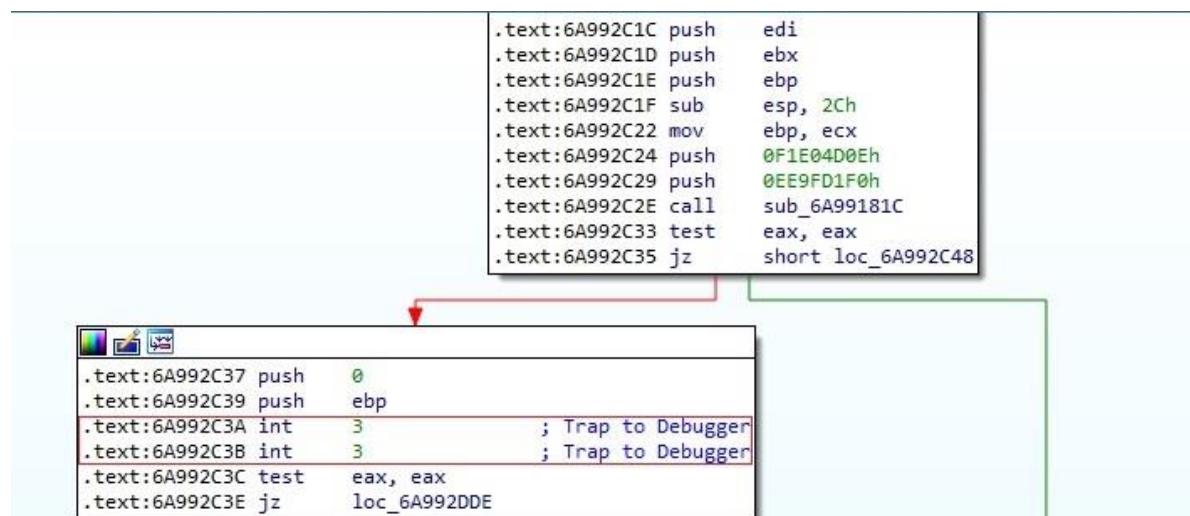


```
.text:6A982618 ; Exported entry 4. DllRegisterServer
.text:6A982618
.text:6A982618
.text:6A982618
.text:6A982618 public DllRegisterServer
.text:6A982618 DllRegisterServer proc near
.text:6A982618
.text:6A982618 arg_0= dword ptr 4
.text:6A982618 arg_4= byte ptr 8
.text:6A982618
.text:6A982618 lea     eax, [esp+arg_4]
.text:6A98261C push   offset aMicrosoft ; "Microsoft"
.text:6A982621 push   eax
.text:6A982622 call   ds:lstrcpw
.text:6A982628 call   ds:GetCommandLineW
.text:6A98262E mov     edx, eax
.text:6A982630 lea     eax, [esp+4]
.text:6A982633 push   eax
.text:6A982634 push   edx
.text:6A982635 call   ds:CommandLineToArgvW
.text:6A98263B mov     ebx, eax
.text:6A98263D mov     eax, [esp+4]
.text:6A982640 mov     edx, 7FFFFFFFh
.text:6A982645 dec     eax
.text:6A982646 mov     [esp+4], eax
.text:6A982649 mov     ecx, [ebx+eax*4]
.text:6A98264C call   sub_6A98D300
.text:6A982651 mov     edx, [esp+4]
.text:6A982654 add     eax, eax
.text:6A982656 push   eax ; Length
.text:6A982657 push   dword ptr [ebx+edx*4] ; Buffer
.text:6A98265A push   0 ; InitialCrc
.text:6A98265C call   ds:RtlComputeCrc32
.text:6A982662 add     eax, 1EC6086Bh
.text:6A982667 mov     [esp+arg_0], eax
.text:6A98266B call   sub_6A99A550
.text:6A982670 add     [esp+arg_0], eax
.text:6A982674 jmp     [esp+arg_0]
.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:



```
.text:6A992C1C push edi
.text:6A992C1D push ebx
.text:6A992C1E push ebp
.text:6A992C1F sub esp, 2Ch
.text:6A992C22 mov ebp, ecx
.text:6A992C24 push 0F1E04D0Eh
.text:6A992C29 push 0EE9FD1F0h
.text:6A992C2E call sub_6A99181C
.text:6A992C33 test eax, eax
.text:6A992C35 jz short loc_6A992C48

.text:6A992C37 push 0
.text:6A992C39 push ebp
.text:6A992C3A int 3 ; Trap to Debugger
.text:6A992C3B int 3 ; Trap to Debugger
.text:6A992C3C test eax, eax
.text:6A992C3E jz loc_6A992DDE
```

Figure 12

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



```
EAAS18C8 68 40 1C A5 6A push dump_patched.6AA51C40
EAAS18CD 6A 01          push 1
EAAS18CF FF D1          call ecx

ecx=<ntdll.RtlAddVectoredExceptionHandler> (77007360)
.text:6AA518CF dump_patched.d11:$118CF #118CF

Default (stdcall)
1: [esp] 00000001
2: [esp+4] 6AA51C40 dump_patched.6AA51C40
3: [esp+8] 6AA518B8 dump_patched.6AA518B8
4: [esp+C] 0004FC54
```

Figure 13

The exception handler displayed in figure 14 expects an exception code as an argument. Whether the exception code is `0xC0000005` (**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".

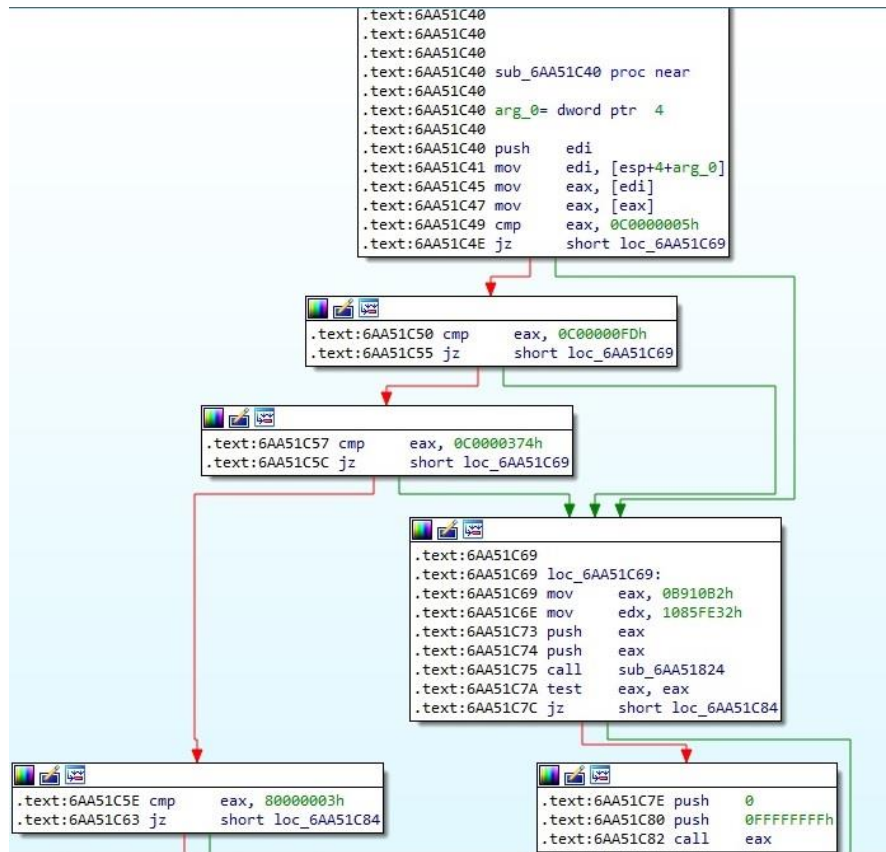


Figure 14

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



Figure 15

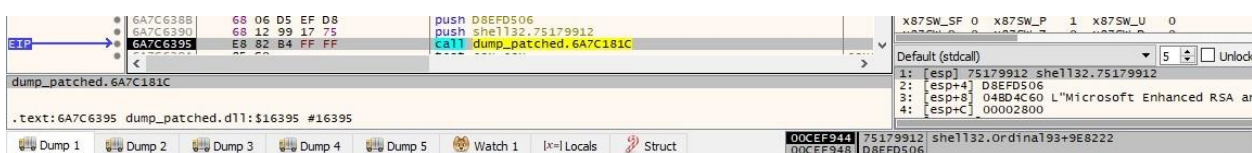


Figure 16

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:

```

6AA4EC1C 53          push ebx
6AA4EC1D 55          push ebp
6AA4EC1E 8B D9       mov ebx,ecx
6AA4EC20 8B 0D 38 10 A6 6A  mov ecx,dword ptr ds:[6AA61038]
6AA4EC26 8B C2       mov eax,edx
6AA4EC28 81 F9 66 4D A8 D0  cmp ecx,D0A84D66
6AA4EC2E 75 08       jne dump_patched.6AA4EC38
6AA4EC30 33 C9       xor ecx,ecx
6AA4EC32 89 0D 38 10 A6 6A  mov dword ptr ds:[6AA61038],ecx
6AA4EC38 81 FB 69 6F 75 7C  cmp ebx,7C756F69
6AA4EC3E 75 10       jne dump_patched.6AA4EC50
6AA4EC40 8B 15 3C 10 A6 6A  mov edx,dword ptr ds:[<&RtlCreateHeap>]
6AA4EC46 81 FA 31 15 7E 51  cmp edx,517E1531
6AA4EC4C 75 30       jne dump_patched.6AA4EC7E
6AA4EC4E EB 4B       jmp dump_patched.6AA4EC98
6AA4EC50 81 FB 86 17 AD 25  cmp ebx,25AD1786
6AA4EC56 75 10       jne dump_patched.6AA4EC68
6AA4EC58 8B 15 40 10 A6 6A  mov edx,dword ptr ds:[<&RtlAllocateHeap>]
6AA4EC5E 81 FA 3D 42 5C 8B  cmp edx,8B5C423D
6AA4EC64 75 18       jne dump_patched.6AA4EC7E
6AA4EC66 EB 33       jmp dump_patched.6AA4EC98
6AA4EC68 81 FB C4 F2 B2 4E  cmp ebx,4EB2F2C4
6AA4EC6E 75 28       jne dump_patched.6AA4EC9B
6AA4EC70 8B 15 44 10 A6 6A  mov edx,dword ptr ds:[<&RtlComputeCrc32>]
6AA4EC76 81 FA D5 5E 30 A4  cmp edx,A4305ED5
6AA4EC7C 74 1D       je dump_patched.6AA4EC9B
6AA4EC7E 8B C2       mov eax,edx
6AA4EC80 EB 1F       jmp dump_patched.6AA4ECA1
6AA4EC82 33 ED       xor ebp,ebp
6AA4EC84 33 D2       xor edx,edx
6AA4EC86 3B 5C 0A 08  cmp ebx,dword ptr ds:[edx+ecx+8]
6AA4EC8A 74 18       je dump_patched.6AA4ECA4
6AA4EC8C 45          inc ebp
6AA4EC8D 83 C2 1C    add edx,1C
6AA4EC90 83 FD 10    cmp ebp,10
6AA4EC93 7C F1      jl dump_patched.6AA4EC86
6AA4EC95 8B 89 C0 01 00 00  mov ecx,dword ptr ds:[ecx+1C0]
6AA4EC9B 85 C9      test ecx,ecx
6AA4EC9D 75 E3      jne dump_patched.6AA4EC82
6AA4EC9F 33 C0      xor eax,eax
6AA4ECA1 5D         pop ebp
6AA4ECA2 5B         pop ebx
6AA4ECA3 C3         ret
  
```

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:

```

6AA51C69 B8 B2 10 89 00  mov eax,B910B2
6AA51C6E BA 32 FE 85 10  mov edx,1085FE32
6AA51C73 50             push eax
6AA51C74 50             push eax
6AA51C75 E8 AA FB FF FF  call dump_patched.6AA51824
6AA51C7A 85 C0         test eax,eax
6AA51C7C 74 06       je dump_patched.6AA51C84
6AA51C7E 6A 00       push 0
6AA51C80 6A FF       push FFFFFFFF
6AA51C82 FF D0       call eax
  
```

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:

```

namespace data-manipulation/encryption/rc4
author moritz.raabe@mandiant.com
scope function
att&ck Defense Evasion:Obfuscated Files or Information [T1027]
mbc Cryptography:Encrypt Data:RC4 [C0027.009], Cryptography:Encryption Key:RC4 KSA [C0028.002]
examples 34404A3FB9804977C6A886CB991F8130:0x403D40, C805528F6844D/CAF5793C025B56F67D:0x4067AE, 9324D1A8AE37A36AE560C37448C9705A:0x404950, 782A48821D88060ADF0F7EF3E8759FEE30DA049E942DAAD18C
54F8AE0E9EB511:0x405C42, 73CE04892E5F39EC82B00C02FC04C70F:0x40646E
function @ 0x6A996428
or:
and:
subscope:
and: = initialize 5
characteristic: Tight loop @ 0x6A99647C
or:
number: 0x100 @ 0x6A996490
and: = initialize 5
characteristic: Tight loop @ 0x6A99649E
or:
number: 0x100 @ 0x6A9964BC
or:
count(mnemonic(CALLS)): 2 or more @ 0x6A99649E, 0x6A9964A3, 0x6A9964AF, 0x6A9964D9, and 6 more...
or: = modulo key length
mnemonic: IDIV @ 0x6A996480
namespace data-manipulation/rc4
author moritz.raabe@mandiant.com
scope function
att&ck Defense Evasion:Obfuscated Files or Information [T1027]
mbc Cryptography:Encrypt Data:RC4 [C0027.009], Cryptography:Generate Pseudo-random Sequence:RC4 PRGA [C0021.004]
examples 34404A3FB9804977C6A886CB991F8130:0x403D80, 34404A3FB9804977C6A886CB991F8130:0x403E50, 9324D1A8AE37A36AE560C37448C9705A:0x4049F0, 73CE04892E5F39EC82B00C02FC04C70F:0x40646E
function @ 0x6A996428
and:
count(characteristic(NXOR)): 1 @ 0x6A996507
or:
count(mnemonic(CALLS)): 4 or more @ 0x6A99649E, 0x6A9964A3, 0x6A9964AF, 0x6A9964D9, and 6 more...
count(basicblock): 4 or more @ 0x6A996428, 0x6A99643E, 0x6A996446, 0x6A996454, and 14 more...
match: contain loop @ 0x6A996428
or:
characteristic: loop @ 0x6A996428
characteristic: Tight loop @ 0x6A99647C, 0x6A99649E
optional:
or:
number: 0x100 @ 0x6A996490, 0x6A9964BC

```

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**:

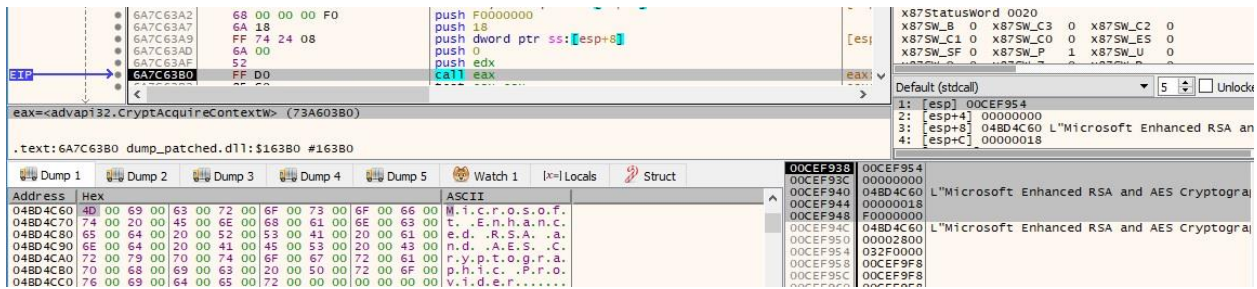


Figure 21

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	Hex	ASCII
04BD0048	73 76 73 68 6F 2A 2E 65 78 65 38 73 63 68 72 65	svsho*.exe;schre
04BD0058	2A 2E 62 61 74 38 56 30 31 2E 6C 6F 2A 38 56 30	*.bat;v01.lo*;v0
04BD0068	31 2E 63 68 2A 38 56 30 31 72 65 73 2A 2E 6A 72	1.ch*;v01res*;jr
04BD0078	73 38 52 61 63 57 6D 69 2A 2E 73 64 66 38 57 65	s;Racwmi*.sdf;we
04BD0088	62 2A 56 30 31 2E 64 61 74 38 64 65 66 61 75 6C	b*v01.dat;defaul
04BD0098	74 2E 72 64 70 38 4E 54 55 53 45 52 2E 44 41 2A	t.rdp;NTUSER.DA*
04BD00A8	3B 2A 2E 6C 6E 6B 38 2A 2E 69 63 6F 3B 2A 2E 69	;*.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
04BD00D8	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
04BD0108	3B 2A 2E 53 59 53 3B 2A 2E 48 4C 46 3B 2A 2E 4C	;*.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	Hex	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 38 56 69 72 74 75 61 6C 53 74 6F 72 65	hes;VirtualStore

Figure 23

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

Address	Hex	ASCII
04BD0048	25 50 72 6F 67 72 61 6D 44 61 74 61 25 5C 4D 69	%ProgramData%\Mi
04BD0058	63 72 6F 73 6F 66 74 5C 57 69 6E 64 6F 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	38 25 77 69 6E 64 69 72 25 38 25 74 65 6D 70 25	;%windir%;%temp%
04BD0088	38 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 38 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	73 5C 41 6C 6C 20 55 73 65 72 73 5C 41 70 70 6C	s\All Users\Appl
04BD00D8	69 63 61 74 69 6F 6E 20 44 61 74 61 5C 41 70 70	ication Data\Ap
04BD00E8	6C 69 63 61 74 69 6F 6E 20 44 61 74 61 61 5C 38 25	lication Data;%
04BD00F8	48 4F 4D 45 44 52 49 56 45 25 5C 55 73 65 72 73	HOMEDRIVE%\Users
04BD0108	5C 41 6C 6C 20 55 73 65 72 73 5C 41 70 70 6C 69	\All Users\Appl
04BD0118	63 61 74 69 6F 6E 20 44 61 74 61 5C 41 70 70 6C	ication Data\Appl
04BD0128	69 63 61 74 69 6F 6E 20 44 61 74 61 5C 38 25 53	ication Data;%S
04BD0138	79 73 74 65 6D 44 72 69 76 65 25 5C 44 6F 63 75	ystemDrive%\Docu
04BD0148	6D 65 6E 74 73 20 61 6E 64 20 53 65 74 74 69 6E	ments and Settin
04BD0158	67 73 5C 41 6C 6C 20 55 73 65 72 73 5C 41 70 70	gs\All Users\Ap
04BD0168	6C 69 63 61 74 69 6F 6E 20 44 61 74 61 61 5C 41 70	lication Data\Ap
04BD0178	70 6C 69 63 61 74 69 6F 6E 20 44 61 74 61 5C 38	plication Data;
04BD0188	25 53 79 73 74 65 6D 44 72 69 76 65 25 5C 55 73	%SystemDrive%\us
04BD0198	65 72 73 5C 41 6C 6C 20 55 73 65 72 73 5C 41 70	ers\All Users\Ap
04BD01A8	70 6C 69 63 61 74 69 6F 6E 20 44 61 74 61 5C 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	Hex	ASCII
04BD0358	3B 34 39 65 65 37 32 38 62 38 6D 73 6F 6C 61 70	;49ee728b;mso1ap
04BD0368	24 2A 38 6D 73 73 71 6C 24 2A 38 64 65 33 61 33	%;mssql\$*;de3a3
04BD0378	35 35 62 38 66 34 34 35 62 31 39 33 38 32 39 37	55b;f445b193;297
04BD0388	33 31 34 39 34 38 39 34 63 32 39 36 31 37 38 34	31494;94c29617;4
04BD0398	61 31 34 33 66 30 38 34 34 32 38 62 39 62 34 38	a143f0;4428b9b4;
04BD03A8	63 31 62 31 66 30 66 62 38 34 64 66 39 63 32 37	clb1f0fb;4df9c27
04BD03B8	36 38 65 33 64 34 36 38 39 32 38 33 34 38 39 66	6;e3d46892;3489f
04BD03C8	39 31 38 35 31 37 38 64 64 35 39 38 66 66 33 36	91;5178dd59;ff36
04BD03D8	34 38 39 31 38 36 64 39 30 61 36 34 39 38 34 36	4891;6d90a649;46
04BD03E8	35 61 34 64 61 37 38 66 31 32 65 33 64 30 39 38	5a4da7;f12e3d09;
04BD03F8	33 37 66 63 39 31 31 61 38 31 37 32 62 31 66 61	37fc911a;172b1fa
04BD0408	66 38 63 64 66 39 37 61 38 62 38 32 31 38 31 63	f;cdf97a8b;2181c
04BD0418	31 35 65 38 33 35 39 32 37 63 30 66 38 36 38 38	15e;35927c0f;688
04BD0428	33 33 64 63 31 38 66 32 61 65 37 39 63 38 63 61	33dc1;f2ae79c;ca
04BD0438	66 66 31 30 62 33 38 65 66 64 64 32 37 34 38 66	ff10b3;efdd274;f
04BD0448	61 34 39 33 65 31 33 38 72 65 70 6F 72 74 73 65	a493e13;reportse
04BD0458	72 76 65 72 24 2A 38 62 32 61 31 37 62 31 38 38	rver\$*;b2a17b18;
04BD0468	32 30 36 63 63 30 62 35 38 66 35 38 32 66 34 66	206cc0b5;f582f4f
04BD0478	33 38 61 33 32 32 31 65 62 31 38 36 61 39 39 35	3;a3221eb1;6a995
04BD0488	66 63 64 38 36 37 66 39 39 38 36 65 38 33 36 30	fdc;67f9986e;360
04BD0498	62 39 37 39 39 38 33 62 39 66 31 62 33 65 38 61	b9799;3b9f1b3e;a
04BD04A8	36 37 37 32 63 39 36 38 35 30 35 30 31 36 34 62	6772c96;5050164b
04BD04B8	38 37 64 30 39 63 35 36 32 38 37 31 34 64 64 32	;7d09c562;714dd2
04BD04C8	65 61 38 36 38 38 32 64 37 34 34 38 62 32 64 61	ea;6882d744;b2da
04BD04D8	35 63 34 30 38 62 65 38 37 66 37 33 38 38 66 63	5c40;be87f738;fc
04BD04E8	34 66 36 34 32 32 38 63 37 62 37 61 39 38 36 36	4f6422;c7b7a9;66
04BD04F8	61 38 65 65 61 64 38 34 36 62 36 30 37 63 32 38	a8eead;46b607c2;
04BD0508	64 62 31 61 63 37 62 62 38 66 33 63 30 34 35 65	db1ac7bb;f3c045e
04BD0518	34 38 66 64 30 37 34 33 39 38 73 71 6C 61 67 65	4;fd07439;sqlage
04BD0528	6E 74 24 2A 38 35 62 64 33 31 61 34 61 38 38 36	nt\$*;5bd31a4a;86

Figure 25

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

Address	Hex	ASCII
048D0048	33 65 39 34 35 35 65 34 38 33 34 32 32 30 63 33	3e9455e4;34220c3
048D0058	33 38 61 39 31 39 39 39 39 63 38 61 39 31 39 39	3;a919999c;a9199
048D0068	39 39 63 38 63 36 63 66 64 61 61 64 38 39 61 31	99c;c6cfdaad;9a1
048D0078	32 31 32 32 39 38 35 35 32 62 39 64 64 62 38 33	21229;552b9ddb;3
048D0088	35 39 62 35 64 63 34 38 64 36 37 64 31 65 36 30	59b5dc4;d67d1e60
048D0098	38 66 64 37 65 31 61 62 30 38 36 39 65 37 62 63	;fd7e1ab0;69e7bc
048D00A8	61 35 38 65 33 38 31 61 34 35 39 38 36 38 35 30	a5;e381a459;6850
048D00B8	37 31 38 35 38 35 38 30 39 66 36 66 37 38 32 30	7185;5809f6f7;20
048D00C8	33 62 62 38 39 63 38 64 64 31 30 39 31 34 34 38	3bb89c;dd109144;
048D00D8	61 39 32 33 66 65 37 61 38 35 35 66 30 61 36 30	a923fe7a;55f0a60
048D00E8	34 38 35 39 64 32 64 62 62 66 38 32 37 34 36 32	4;59d2dbbf;27462
048D00F8	66 66 66 38 38 63 62 34 31 35 34 63 38 39 61 34	fff;8cb4154c;9a4
048D0108	66 37 66 34 33 38 65 30 36 37 64 62 33 30 38 66	f7f43;e067db30;f
048D0118	63 39 35 62 61 39 64 38 63 63 35 66 35 62 66 31	c95ba9d;cc5f5bf1
048D0128	38 34 36 37 32 35 35 65 34 38 37 32 64 39 62 35	;467255e4;72d9b5
048D0138	39 63 38 62 34 66 61 36 63 66 38 31 37 32 62 31	9c;b4fa6cf;172b1
048D0148	66 61 66 38 35 65 65 65 63 37 31 35 38 36 63 35	faf;5eeec715;6c5
048D0158	62 31 39 32 37 38 39 35 63 37 36 32 34 36 38 66	b1927;95c76246;f
048D0168	35 38 32 66 34 66 33 38 61 33 32 32 31 65 62 31	582f4f3;a3221eb1
048D0178	38 39 33 61 37 66 32 32 31 38 66 62 37 38 63 35	;93a7f221;fb78c5
048D0188	33 38 63 38 62 64 39 66 34 64 38 61 65 36 34 63	3;c8bd9f4d;ae64c
048D0198	36 62 33 38 62 31 31 37 65 66 63 38 33 36 30 62	6b3;b117efc;360b
048D01A8	39 37 39 39 38 33 62 39 66 31 62 33 65 38 61 36	9799;3b9f1b3e;a6
048D01B8	37 37 32 63 39 36 38 62 65 38 37 66 37 33 38 38	772c96;be87f738;
048D01C8	63 39 39 31 34 37 61 36 38 64 37 37 36 31 66 64	c99147a6;d7761fd
048D01D8	64 38 61 33 37 62 61 66 33 37 38 61 37 62 30 39	d;a37baf37;a7b09
048D01E8	61 65 65 38 66 35 64 63 35 31 64 35 38 66 39 32	ae;f5dc51d5;f92
048D01F8	36 32 38 61 30 38 35 32 66 32 62 38 31 31 38 31	628a0;52f2b811;1
048D0208	65 61 61 37 36 37 32 38 65 36 33 66 39 30 30 34	aaa672;e63f9004
048D0218	38 73 6f 70 68 6f 73 20 63 6c 69 65 6E 74 20 66	;sophos client f
048D0228	69 72 65 77 61 6c 6c 2a 38 38 39 36 64 36 39 61	irewall*;896d69a
048D0238	37 38 36 30 36 64 34 64 39 39 38 65 30 33 35 37	7;606d4d99;e0357
048D0248	36 35 38 38 39 33 63 34 31 37 34 34 38 61 63 31	658;93c41744;ac1
048D0258	34 64 63 39 30 38 64 61 65 62 64 61 33 62 38 73	4dc90;daebda3b;s
048D0268	6f 70 68 6f 73 20 6d 63 73 2a 38 61 38 34 34 32	ophos mcs*;a8442
048D0278	32 39 39 38 62 62 37 61 39 31 37 61 38 65 33 37	299;bb7a917a;e37
048D0288	66 39 64 30 38 38 61 37 35 34 63 61 34 38 32 36	f9d08;a754ca4;26
048D0298	62 35 64 31 38 63 38 39 63 32 32 61 33 38 34 38	b5d18c;9c22a384;
048D02A8	31 66 30 35 64 34 36 63 38 62 63 38 31 66 66 33	1f05d46c;bc81ff3
048D02B8	39 38 61 64 38 36 66 35 37 38 38 34 30 32 62 65	9;ad86f578;402be
048D02C8	66 64 34 38 73 6f 70 68 6f 73 20 77 65 62 20 69	fd4;sophos web 1
048D02D8	6E 74 65 6C 6C 69 67 65 6E 63 65 2A 38 62 39 33	ntelligence*;b93
048D02E8	30 33 62 34 33 38 38 37 61 64 39 31 63 32 38 38	03b43;87ad91c2;8
048D02F8	31 65 32 65 39 66 62 38 73 6f 70 68 6f 73 70 61	1e2e9fb;sophospa
048D0308	74 63 68 2A 38 32 32 30 34 64 35 64 31 38 38 33	tch*;2204d5d1;83
048D0318	31 32 39 65 34 34 38 38 33 31 32 39 65 34 34 38	129e44;83129e44;
048D0328	64 34 62 66 61 62 37 38 38 32 33 62 30 37 63 61	d4bfb78;23b07ca
048D0338	30 38 62 37 38 66 39 62 34 65 38 39 38 37 31 36	0;b78f9b4e;98716
048D0348	33 65 39 38 34 35 61 31 63 31 39 37 38 64 65 33	3e9;45a1c197;de3
048D0358	64 61 62 63 37 38 66 63 34 66 36 34 32 32 38 31	dabc7;fc4f6422;1
048D0368	34 64 61 64 31 61 38 35 33 61 32 33 35 33 62 38	4dad1a;53a2353b;
048D0378	62 31 35 34 63 66 64 34 38 62 65 30 39 32 34 38	b154cfd4;be09248
048D0388	31 38 31 38 65 35 38 38 63 36 38 35 39 65 38 34	1;18e5886c;59e84
048D0398	65 65 34 38 35 65 63 39 61 35 62 62 00 00 00 00	ee4;5ec9a5bb....

Figure 26

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

Address	Hex	ASCII
048D0048	2E 70 61 79 30 72 67 72 69 65 66 00 00 00 00 00	.pay0rgrief.....

Figure 27

The ransom note file name is also decrypted using RC4:

Address	Hex	ASCII
048D0048	2E 69 77 61 6E 74 32 73 75 72 76 69 76 65 2E 68	.iwant2survive.h
048D0058	74 6D 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	Hex	ASCII
04BD0048	4D 49 49 42 49 6A 41 4E 42 67 68 71 68 68 69 47	MIIBIjANBgkqhkiG
04BD0058	39 77 30 42 41 51 45 46 41 41 4F 43 41 51 38 41	9w0BAQEFAAOCAQBA
04BD0068	40 49 49 42 43 67 48 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	0D 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
04BD00A8	45 58 28 34 74 39 48 4E 43 33 52 72 49 69 68 32	EX+4t9HNC3RrIik2
04BD00B8	6E 65 76 6D 50 28 6F 54 74 79 66 66 59 62 52 71	nevmp+oTtyffYbRq
04BD00C8	53 48 0D 0A 32 46 4C 6F 66 35 43 64 54 4A 67 71	SK..2FLoF5CdTJgq
04BD00D8	4C 6C 38 36 73 58 30 7A 2F 7A 4E 58 4A 69 30 28	L186sX0z/zNXJi0+
04BD00E8	6A 31 37 65 36 67 66 38 63 59 4F 52 54 4D 65 75	j17e6gf8cYORTMeu
04BD00F8	6D 47 4E 36 28 48 30 65 41 79 28 58 50 54 53 45	mGN6+H0eAy+XPTSE
04BD0108	53 45 57 31 0D 0A 43 41 53 4D 53 55 4C 65 32 65	SEW1..CASMSULe2e
04BD0118	6C 45 31 48 47 34 63 45 4F 34 47 55 6D 28 68 47	1E1KG4cEO4Gum+kG
04BD0128	76 31 33 47 70 59 56 74 61 47 59 47 36 54 46 53	v13GpYVtaGYG6TFS
04BD0138	50 48 44 28 4E 39 49 4C 41 77 64 6C 43 46 30 39	PHD+N91Awd1CF09
04BD0148	48 52 28 51 30 56 0D 0A 70 56 67 44 6D 4E 56 69	KR+Q0V..pvgDmNVi
04BD0158	6C 4A 39 40 73 31 63 30 72 70 56 35 48 71 44 63	1J9Ms1c0rpv5KqDc
04BD0168	33 41 48 31 55 2F 66 69 30 77 58 77 45 69 6F 53	3AH1U/fi0wXwEi0S
04BD0178	2F 31 72 42 65 6B 70 30 48 54 63 45 5A 38 48 44	/1rBekp0HTCEZ8HD
04BD0188	32 44 49 6A 66 56 57 71 0D 0A 6C 4E 52 30 4C 61	2D1jfVwq..1NR0La
04BD0198	48 42 59 5A 36 58 69 4A 61 70 36 64 43 28 43 52	KBYZ6X1Jap6dC+CR
04BD01A8	4F 73 44 53 76 65 31 6A 38 62 69 47 42 74 69 6A	OSD5ve1j8biGBTij
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 48 77 31 0D 0A 79 51 49 44	B/egQilKw1..yQID
04BD01D8	41 51 41 42 0D 0A 00 00 00 00 00 00 00 00 00 00	AQAB.....

Figure 29

The content of the ransom note is also revealed:

Address	Hex	ASCII
04BD0048	3C 68 74 6D 6C 3E 3C 68 65 61 64 3E 3C 73 74 79	<html><head><sty
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 78 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 38 20 66 6F 6E 74 2D 73 74 79 6C	rrow'; font-styl
04BD0098	65 3A 20 6E 6F 72 6D 61 6C 38 20 66 6F 6E 74 2D	e: normal; font-
04BD00A8	77 65 69 67 68 74 3A 20 34 30 30 38 20 66 6F 6E	weight: 400; fon
04BD00B8	74 2D 64 69 73 70 6C 61 79 3A 20 73 77 61 70 3B	t-display: swap;
04BD00C8	20 73 72 63 3A 20 75 72 6C 28 68 74 74 70 73 3A	src: url(https:
04BD00D8	2F 2F 66 6F 6E 74 73 2E 67 73 74 61 74 69 63 2E	//fonts.gstatic.
04BD00E8	63 6F 6D 2F 73 2F 74 6F 6D 6F 72 72 6F 77 2F 76	com/s/tomorrow/v
04BD00F8	35 2F 57 42 4C 6D 72 45 54 4E 62 46 74 5A 43 65	5/WBLmrETNbFtZCe
04BD0108	47 71 67 52 58 63 65 32 44 77 4C 51 2E 77 6F 66	GqgRxc2DwLQ.wof
04BD0118	66 32 29 20 66 6F 72 6D 61 74 28 27 77 6F 66 66	f2) format('woff
04BD0128	32 27 29 3B 20 75 6E 69 63 6F 64 65 2D 72 61 6E	2'); unicode-ran
04BD0138	67 65 3A 20 55 28 30 31 30 30 2D 30 32 34 46 2C	ge: U+0100-024F,
04BD0148	20 55 28 30 32 35 39 2C 20 55 28 31 45 30 30 2D	U+0259, U+1E00-
04BD0158	31 45 46 46 2C 20 55 28 32 30 32 30 2C 20 55 28	1EFF, U+2020, U+
04BD0168	32 30 41 30 2D 32 30 41 42 2C 20 55 28 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	Hex	ASCII
04BD0048	43	C
04BD0058	[REDACTED]	[REDACTED]
04BD0068	6F 75 20 61 72 65 20 66 75 [REDACTED] 65 64 2E 00 00	ou are fu[REDACTED]ed...

Figure 31

Address	Hex	ASCII
048D0048	44 4F 20 4E 4F 54 20 54 4F 55 43 48 20 41 4E 59	DO NOT TOUCH ANY
048D0058	54 48 49 4E 47 21 0D 0A 0D 0A 57 68 61 74 20 74	THING!....What t
048D0068	6F 20 64 6F 20 28 20 70 61 73 73 77 6F 72 64 3A	o do (password:
048D0078	20 [REDACTED])::
048D0088	0A 68 74 74 70 3A 2F 2F 70 61 79 6F 72 67 7A 33	.http://payorgz3
048D0098	[REDACTED]	[REDACTED]
048D00A8	[REDACTED]	[REDACTED]
048D00B8	69 33 62 76 32 6D 6C 77 67 63 69 72 75 6E 61 64	i3bv2mlwgcirunad
048D00C8	2E 6F 6E 69 6F 6E 2F 64 65 6D 61 6E 64 2F 64 61	.onion/demand/da
048D00D8	[REDACTED]	[REDACTED]
048D00E8	[REDACTED]	[REDACTED]
048D00F8	0D 0A 55 53 45 20 54 4F 52 2E 0D 0A 0D 0A 50 30	..USE TOR....PO
048D0108	47 5F 0D 0A 0D 0A 00 00 00 00 00 00 00 00 00 00	G.....

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	Hex	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.l.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 6E 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.s. .D.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	Hex	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	Hex	ASCII
04E76728	44 00 65 00 6C 00 65 00 74 00 65 00 20 00 53 00	D.e.l.e.t.e. .S.
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.l.l. ./..Q.u.i.
04E76758	65 00 74 00 00 00 00 00 00 00 00 00 00 00 00 00	e.t.....

Figure 35

Address	Hex	ASCII
04E76728	64 65 6C 65 74 65 20 73 68 61 64 6F 77 73 20 61	delete shadows a
04E76738	6C 6C 0D 0A 65 78 69 74 0D 0A 00 00 00 00 00 00	ll..exit.....

Figure 36

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:

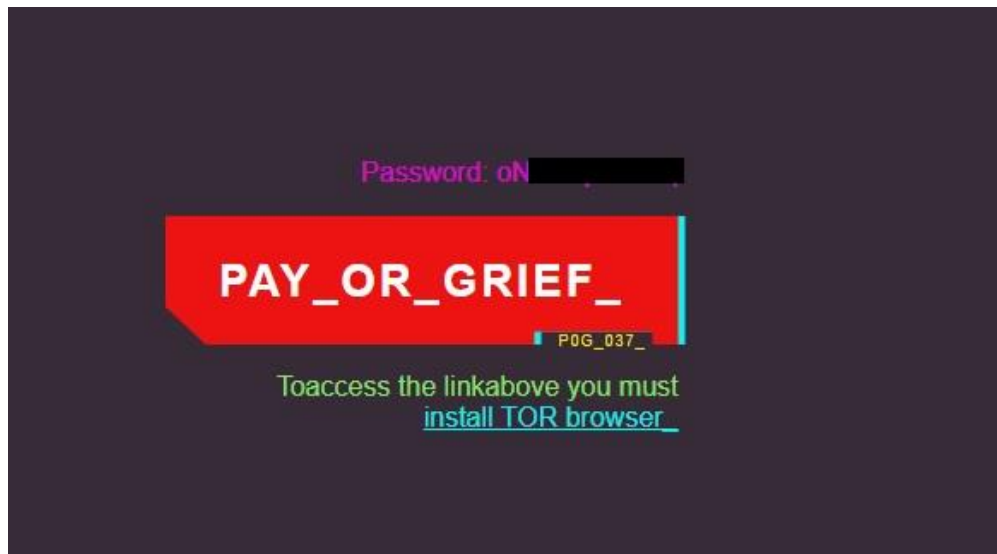


Figure 37

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

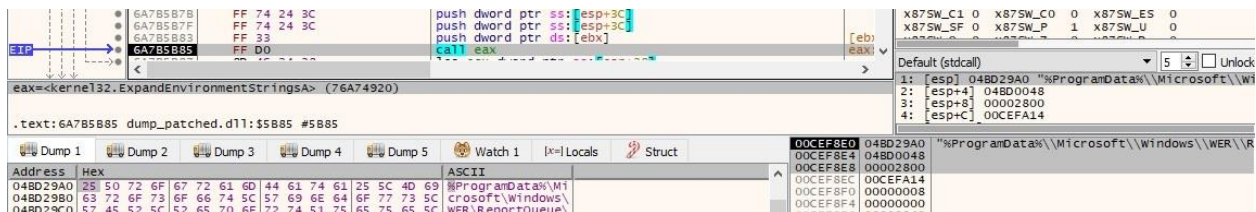


Figure 38

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



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**):

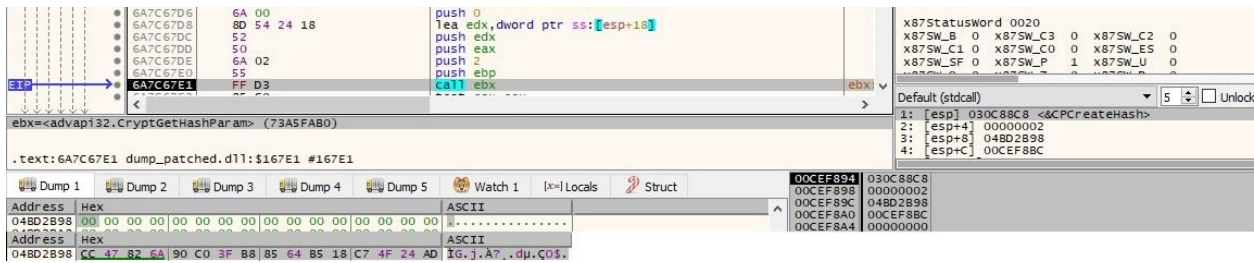


Figure 44

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:



Figure 45

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



Figure 46

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

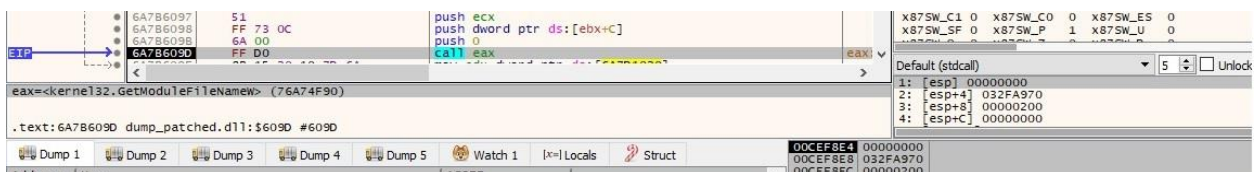


Figure 47

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

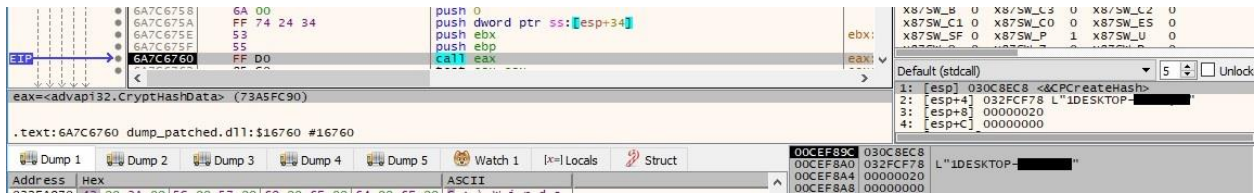


Figure 48

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

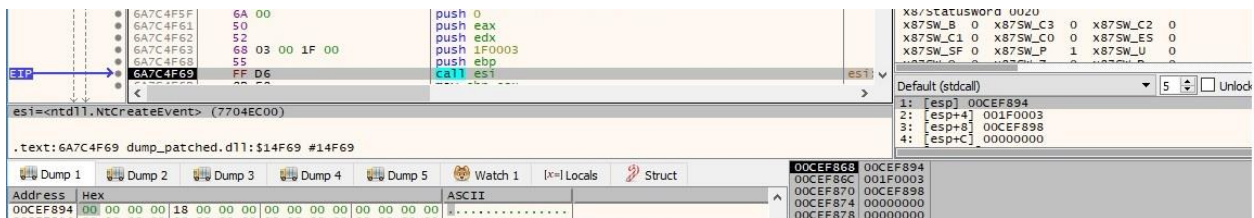


Figure 49

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

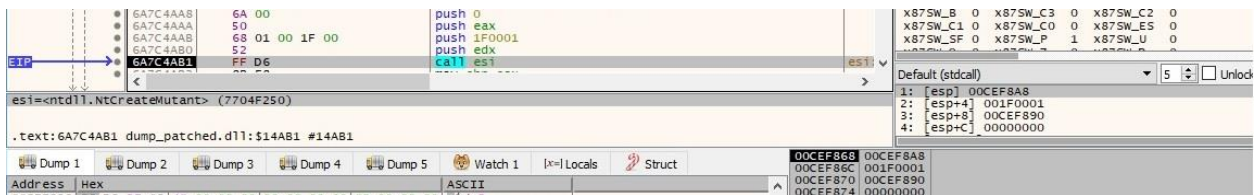


Figure 50

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

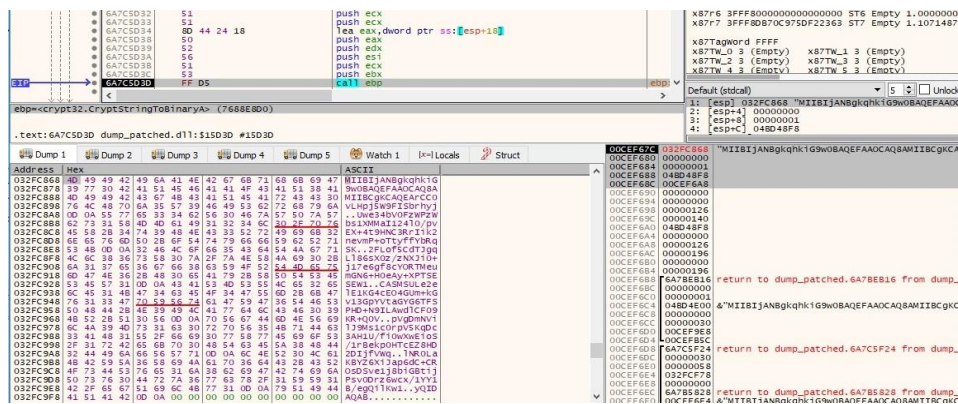


Figure 51

Address	Hex	ASCII
04BD48F8	30 82 01 22 30 0D 06 09 2A 86 48 86 F7 0D 01 01	0.. "0...*.H.÷...
04BD4908	01 05 00 03 82 01 0F 00 30 82 01 0A 02 82 01 010.....
04BD4918	00 AC 20 B4 BC B1 E9 8F 95 BD 14 84 98 AE 1C A3	.. %±é.%.%..@.f
04BD4928	53 07 B7 E1 B5 74 17 35 8F CD 66 EC D5 73 0C 68	S. .ápt.5.ifiòs.h
04BD4938	8D 76 E2 5D 3F A6 F1 17 FB 8B 7D 1C D0 B7 46 B2	.vâ] ?;ñ.ù.}.D.F="
04BD4948	22 93 69 DE BE 63 FE A1 38 72 7D F6 1B 46 A4 8A	".ip%çpi;r}ô.F#.
04BD4958	D8 52 E8 7F 90 9D 4C 98 2A 2E 5F 3A B1 7D 33 FF	ØRè...L.*.:±}3ÿ
04BD4968	33 57 26 2D 3E 8F 5E DE EA 07 FC 71 83 91 4C C7	3W&->.Àpè.ùq..LÇ
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 B8 70 43 88 19	...IBPÙèDôj.pC..
04BD4998	49 BE 90 68 F5 DC 6A 58 56 D6 86 60 6E 93 15 23	I%.kôújxvö. n. #
04BD49A8	C7 0F E3 7D 20 80 30 76 50 85 D3 D2 91 F9 0D 15	Ç.ã} °ovP.óó.ù..
04BD49B8	A5 58 03 98 D5 62 94 9F 4C B3 57 34 AE 95 79 2A	%X..òb..L*W4@.y*
04BD49C8	A0 DC DC 01 F5 53 F7 E2 D3 05 F0 12 2A 12 FF 5A	ÜÜ.òs-áo.ð.*.ÿZ
04BD49D8	C1 7A 4A 74 1D 37 04 67 C1 C3 D8 32 23 7D 55 AA	ÀzJt.7.gÀA02#}U#
04BD49E8	94 D4 74 2D A2 81 61 9E 97 88 96 A9 E9 D0 BE 09	.òt-c.a....@èD%.
04BD49F8	13 AC 0D 2B DE D6 3F 1B 88 60 6D 8A 33 EC BF 40	..+p0?..m.3i¿@
04BD4A08	EB CF AC 1C C7 FD 58 63 50 7F 7A 04 22 94 AC 35	èI-.çYXçP.z."-5
04BD4A18	C9 02 03 01 00 01 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	Hex	ASCII
032FB178	90 B1 2F 03 02 00 00 00 A8 B1 2F 03 0E 01 00 00	±/..... ±/.....
032FB188	80 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 01 010.....
032FB1B8	00 AC 20 B4 BC B1 E9 8F 95 BD 14 84 98 AE 1C A3	.. %±é.%.%..@.f
032FB1C8	53 07 B7 E1 B5 74 17 35 8F CD 66 EC D5 73 0C 68	S. .ápt.5.ifiòs.h
032FB1D8	8D 76 E2 5D 3F A6 F1 17 FB 8B 7D 1C D0 B7 46 B2	.vâ] ?;ñ.ù.}.D.F="
032FB1E8	22 93 69 DE BE 63 FE A1 38 72 7D F6 1B 46 A4 8A	".ip%çpi;r}ô.F#.
032FB1F8	D8 52 E8 7F 90 9D 4C 98 2A 2E 5F 3A B1 7D 33 FF	ØRè...L.*.:±}3ÿ
032FB208	33 57 26 2D 3E 8F 5E DE EA 07 FC 71 83 91 4C C7	3W&->.Àpè.ùq..LÇ
032FB218	AE 98 63 7A F8 7D 1E 03 2F 97 3D 34 84 48 45 B5	@.czø}.../=4.HEµ
032FB228	08 04 8C 49 42 DE D9 E9 44 D4 A1 B8 70 43 88 19	...IBPÙèDôj.pC..
032FB238	49 BE 90 68 F5 DC 6A 58 56 D6 86 60 6E 93 15 23	I%.kôújxvö. n. #
032FB248	C7 0F E3 7D 20 80 30 76 50 85 D3 D2 91 F9 0D 15	Ç.ã} °ovP.óó.ù..
032FB258	A5 58 03 98 D5 62 94 9F 4C B3 57 34 AE 95 79 2A	%X..òb..L*W4@.y*
032FB268	A0 DC DC 01 F5 53 F7 E2 D3 05 F0 12 2A 12 FF 5A	ÜÜ.òs-áo.ð.*.ÿZ
032FB278	C1 7A 4A 74 1D 37 04 67 C1 C3 D8 32 23 7D 55 AA	ÀzJt.7.gÀA02#}U#
032FB288	94 D4 74 2D A2 81 61 9E 97 88 96 A9 E9 D0 BE 09	.òt-c.a....@èD%.
032FB298	13 AC 0D 2B DE D6 3F 1B 88 60 6D 8A 33 EC BF 40	..+p0?..m.3i¿@
032FB2A8	EB CF AC 1C C7 FD 58 63 50 7F 7A 04 22 94 AC 35	èI-.çYXçP.z."-5
032FB2B8	C9 02 03 01 00 01 00 00 00 00 00 00 00 00 00	É.....

Figure 54

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

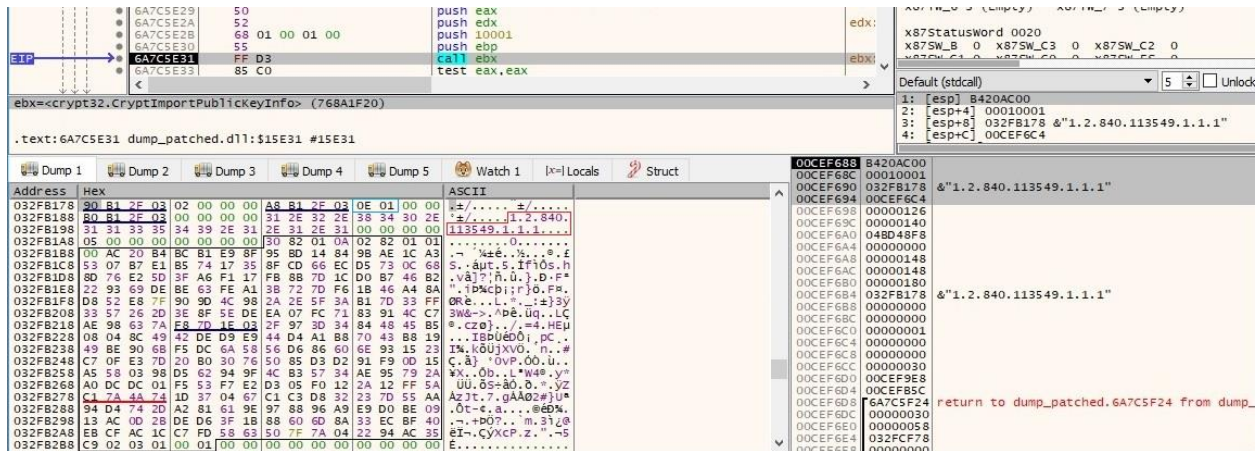


Figure 55

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).



Figure 56

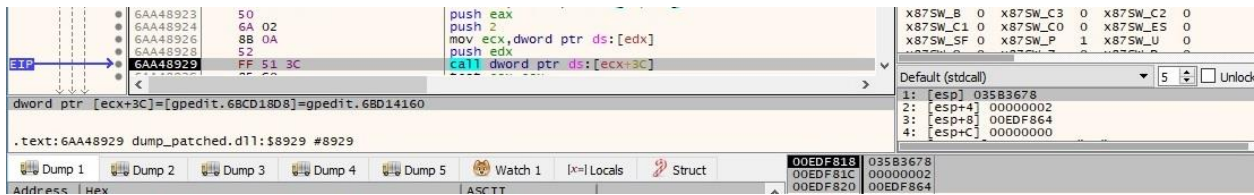


Figure 60

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

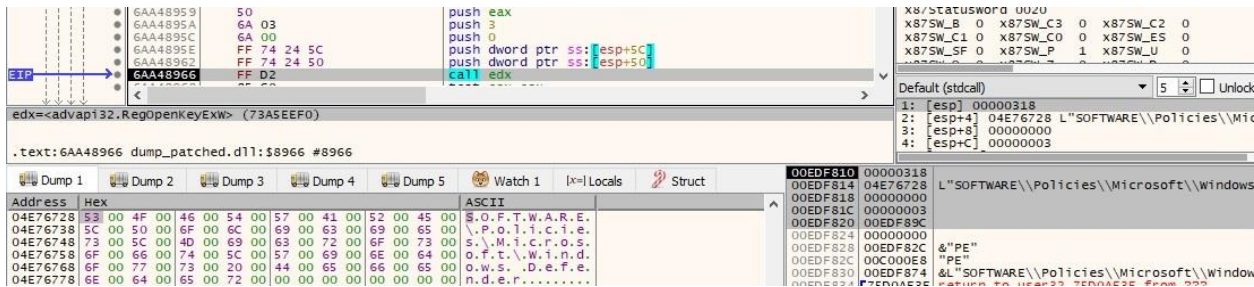


Figure 61

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

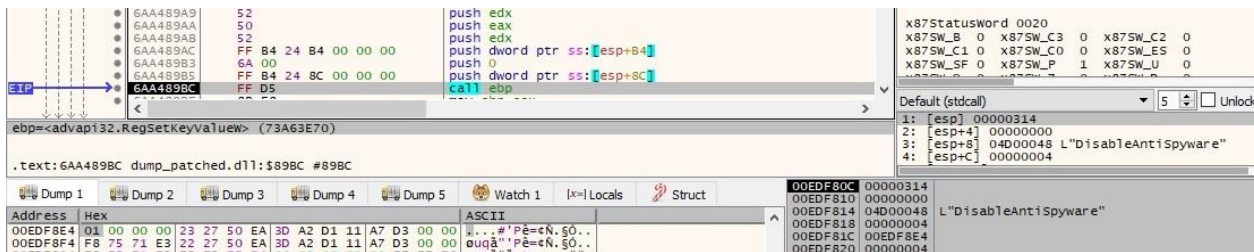


Figure 62



Figure 63

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-00c04fbbcf2} and pGuid is set to {3D271CFC-2BC6-4AC2-B633-3BDFF5BDAB2A}:

Figure 64

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

Figure 65

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

Figure 66

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

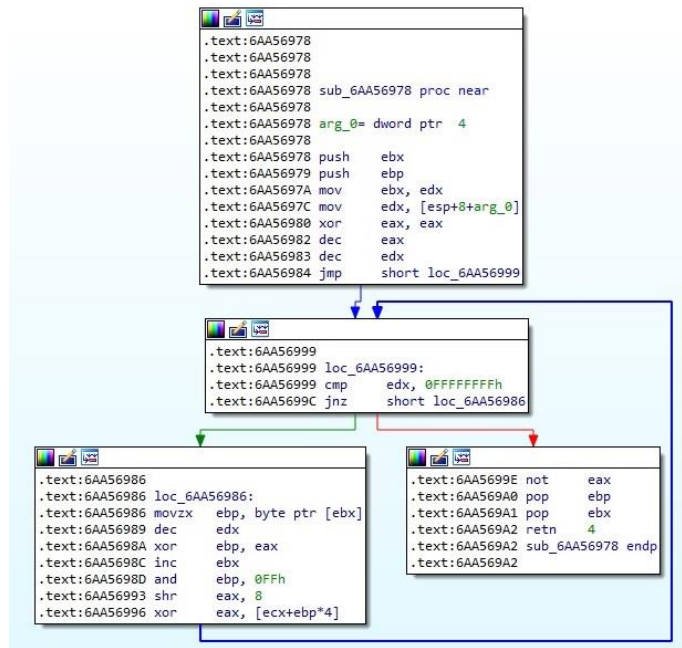


Figure 67

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:

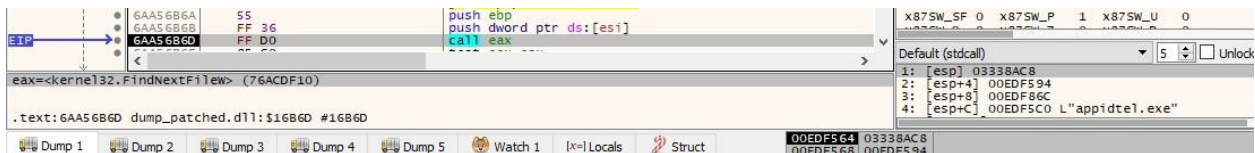


Figure 68

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

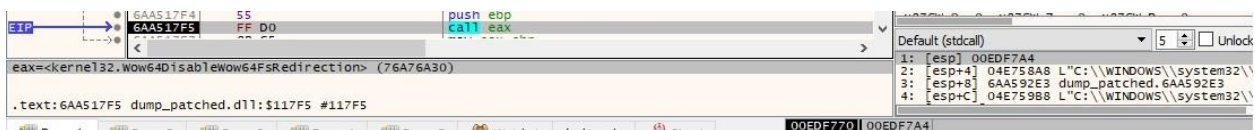


Figure 69

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

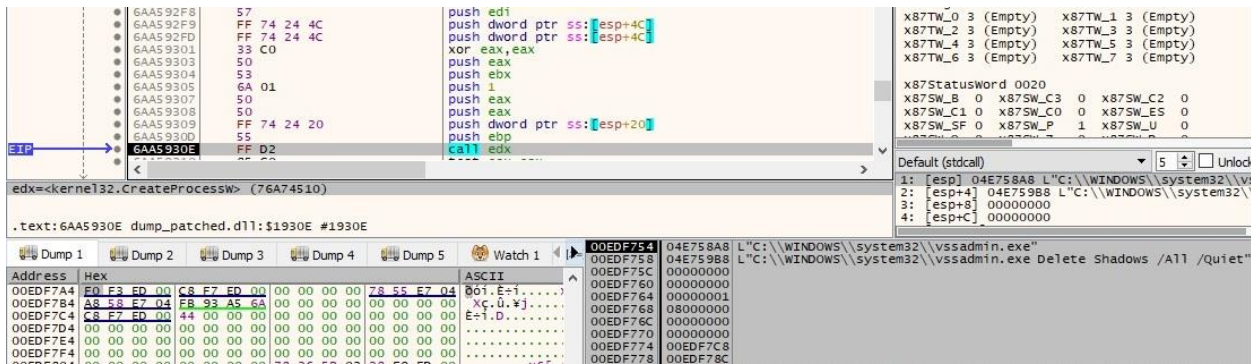


Figure 70

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



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:

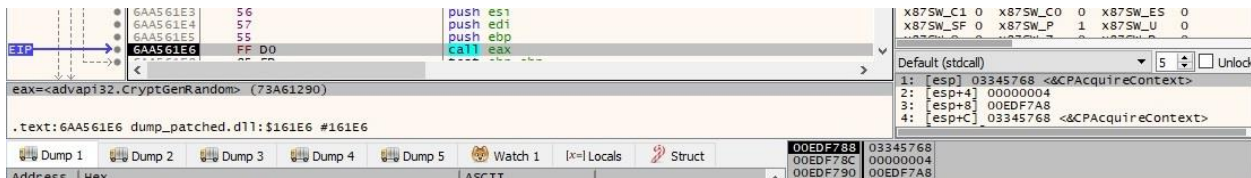


Figure 72

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

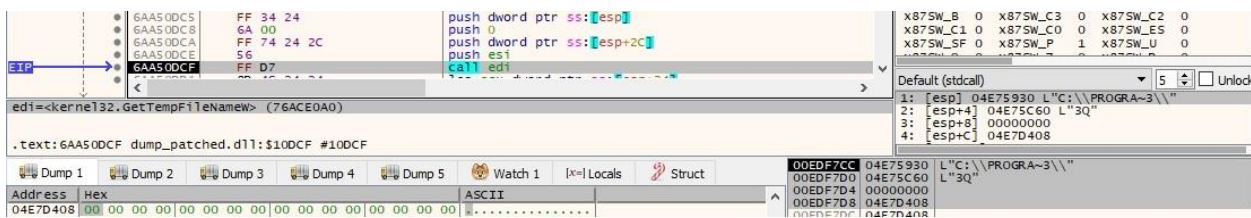


Figure 73

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

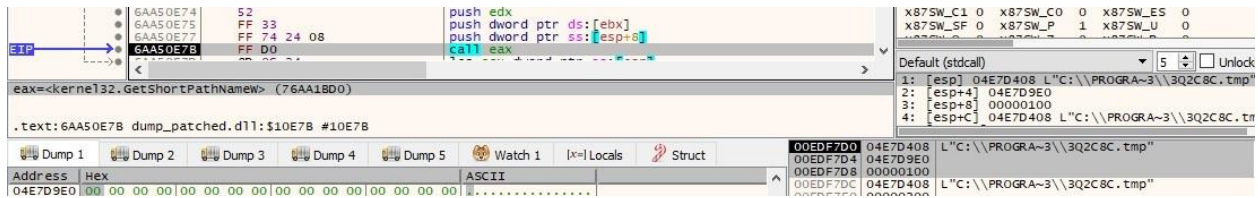


Figure 74

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

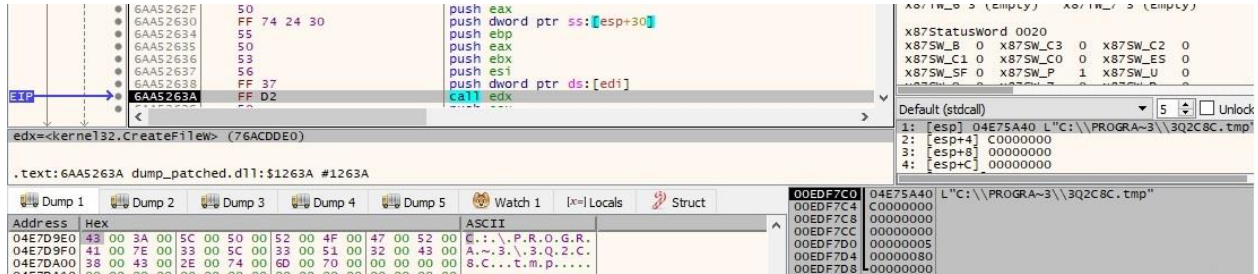


Figure 75

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 0xFFFFFFFF):

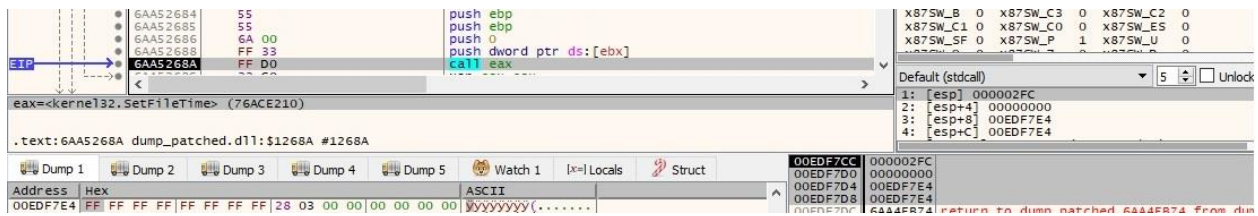


Figure 76

The file is populated with the following content:

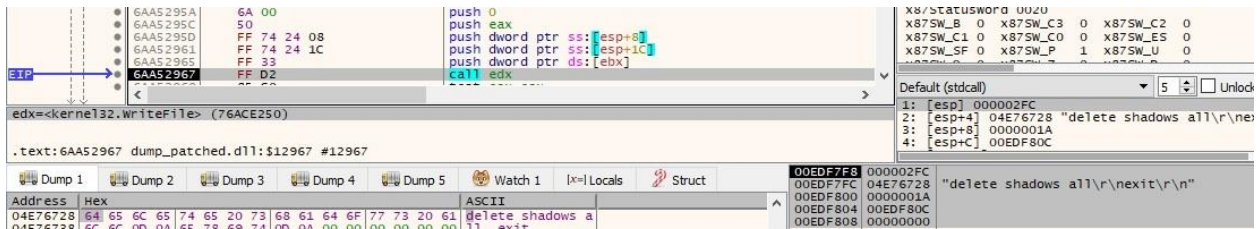


Figure 77

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

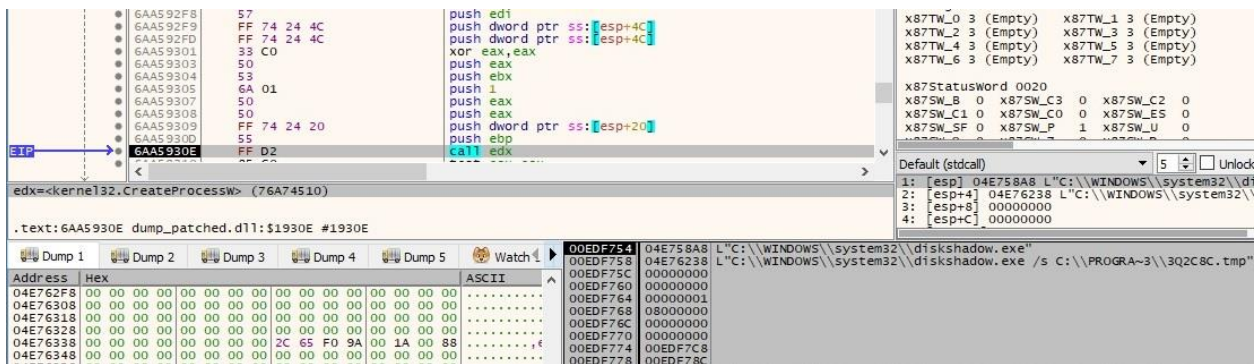


Figure 78

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

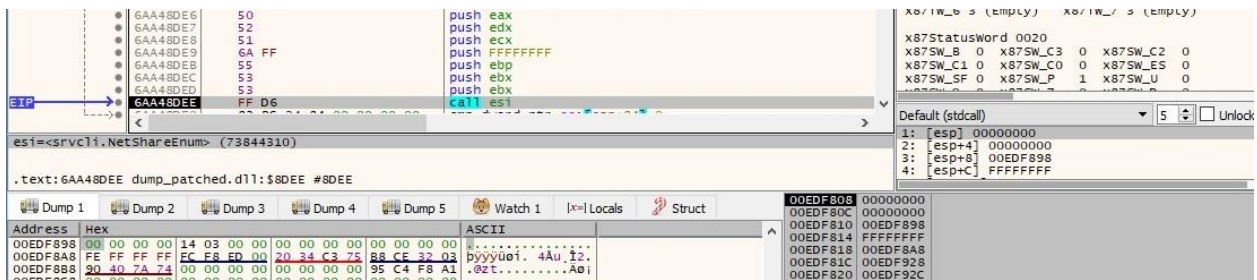


Figure 79

REFERENCES

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

APPENDIX

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

Address	Hex	ASCII
04BD0048	39 63 34 66 61 34 61 38 66 31 39 37 39 39 31 64	9c4fa4a;f197991d
04BD0058	38 63 35 62 66 64 66 35 34 38 32 63 36 35 65 33	;c5bfd54;2c65e3
04BD0068	63 31 38 33 39 38 31 61 30 64 34 38 35 66 63 38	c1;3981a0d4;5fc8
04BD0078	34 39 65 37 38 61 65 35 61 32 32 62 34 38 36 32	49e7;ae5a22b4;62
04BD0088	37 34 66 61 36 34 38 66 36 32 35 32 36 62 39 38	74fa64;f62526b9;
04BD0098	64 63 34 30 61 64 62 61 38 34 31 30 37 61 61 37	dc40adba;4107aa7
04BD00A8	36 38 62 66 64 66 35 32 39 65 38 33 30 36 64 35	6;bfd529e;306d5
04BD00B8	31 61 30 38 66 31 63 63 32 33 65 34 38 31 38 66	1a0;f1cc23e4;18f
04BD00C8	33 62 39 39 31 38 34 62 38 63 66 62 61 38 36 63	3b991;4b8cfba;6c
04BD00D8	63 65 66 36 62 62 38 32 35 61 66 64 34 33 38 31	cef6bb;25afd43;1
04BD00E8	61 33 35 39 30 32 61 38 36 64 34 37 38 61 61 66	a35902a;6d478aaf
04BD00F8	38 34 39 64 39 31 36 32 35 38 34 36 39 63 38 66	;49d91625;469c8f
04BD0108	65 30 38 31 66 38 39 36 62 64 62 38 32 65 33 30	e0;1f896bdb;2e30
04BD0118	64 37 34 35 38 37 35 62 32 64 39 37 31 38 39 37	d745;75b2d971;97
04BD0128	62 37 35 31 62 33 00 00 00 00 00 00 00 00 00	b751b3.....

Figure 80

Address	Hex	ASCII
04BD0048	36 64 34 37 38 61 61 66 38 65 62 30 31 32 62 30	6d478aaf;eb012b0
04BD0058	30 38 36 36 32 62 33 61 36 37 38 38 33 65 64 39	0;662b3a67;83ed9
04BD0068	38 61 33 38 39 31 65 35 33 66 38 66 38 34 62 37	8a3;91e53f8f;4b7
04BD0078	61 66 36 66 36 38 63 65 38 63 35 36 62 37 38 39	af6f6;ce8c56b7;9
04BD0088	63 34 66 61 34 61 38 66 31 39 37 39 39 31 64 38	c4fa4a;f197991d;
04BD0098	63 35 62 66 64 66 35 34 38 32 63 36 35 65 33 63	c5bfd54;2c65e3c
04BD00A8	31 38 33 39 38 31 61 30 64 34 38 35 66 63 38 34	1;3981a0d4;5fc84
04BD00B8	39 65 37 38 32 30 66 33 61 61 61 35 38 34 36 39	9e7;20f3aaa5;469
04BD00C8	63 38 66 65 30 38 64 38 64 61 39 32 37 36 38 65	c8fe0;d8da9276;e
04BD00D8	63 34 38 64 64 34 35 38 38 61 32 37 61 61 31 39	c48dd45;8a27aa19
04BD00E8	38 37 35 61 39 33 33 62 36 38 62 30 31 38 64 34	;75a933b6;b018d4
04BD00F8	37 65 38 35 61 30 33 34 61 35 61 38 63 31 33 66	7e;5a034a5a;c13f
04BD0108	39 30 63 61 38 32 36 39 66 63 37 33 38 37 35 65	90ca;269fc73;75e
04BD0118	31 32 65 39 65 38 32 32 38 32 61 64 30 38 38 34	12e9e;2282ad08;4
04BD0128	61 38 32 65 64 30 61 38 63 36 64 32 63 65 36 37	a82ed0a;c6d2ce67
04BD0138	38 34 37 39 35 36 31 65 38 33 64 38 63 62 61 35	;479561e;3d8cba5
04BD0148	38 63 61 62 33 32 62 65 37 38 39 65 30 37 33 31	;cab32be7;9e0731
04BD0158	39 34 38 63 34 33 38 39 37 65 38 38 31 37 66 37	94;c43897e8;17f7
04BD0168	66 36 65 63 38 63 32 64 34 32 33 62 33 38 62 30	f6ec;c2d423b3;b0

Figure 81

Address	Hex	ASCII
04BD0048	31 61 32 31 32 34 63 30 38 32 30 66 33 61 61 61	1a2124c0;20f3aaa
04BD0058	35 38 34 35 36 62 31 30 39 66 38 38 39 35 61 62	5;456b109f;895ab
04BD0068	64 37 33 38 34 39 64 39 31 36 32 35 38 36 36 32	d73;49d91625;662
04BD0078	62 33 61 36 37 38 38 33 65 64 39 38 61 33 38 32	b3a67;83ed98a3;2
04BD0088	64 33 34 36 31 64 30 38 38 62 66 38 64 61 31 36	d3461d0;8bf8da16
04BD0098	38 65 30 39 33 37 31 34 35 38 32 66 62 61 33 37	;e0937145;2fba37
04BD00A8	30 36 38 34 36 39 33 66 30 31 34 38 32 63 36 35	06;4693f014;2c65
04BD00B8	65 33 63 31 38 33 39 38 31 61 30 64 34 38 34 34	e3c1;3981a0d4;44
04BD00C8	35 37 61 64 31 36 38 66 35 38 64 65 30 36 30 38	57ad16;f58de060;
04BD00D8	63 35 39 39 31 65 39 62 38 62 64 61 66 66 30 63	c5991e9b;bdaff0c
04BD00E8	38 31 64 36 33 37 33 36 62 38 39 35 66 36 63 64	;1d63736b;95f6cd
04BD00F8	37 35 38 62 33 32 37 39 63 34 35 38 38 65 30 61	75;b3279c45;8e0a
04BD0108	36 32 66 31 38 34 61 36 65 34 32 62 36 38 65 31	62f1;4a6e42b6;e1
04BD0118	64 31 32 34 66 61 38 33 33 63 38 30 65 33 38 36	d124fa;33c80e3;6
04BD0128	64 36 64 34 62 61 35 38 35 33 63 63 61 39 65 30	d6d4ba5;53cca9e0
04BD0138	38 33 64 65 34 66 39 34 39 38 65 36 39 37 37 35	;3de4f949;e69775
04BD0148	35 32 38 62 62 62 62 61 35 65 36 38 32 36 39 66	52;bbba5e6;269f
04BD0158	63 37 33 38 31 37 66 37 66 36 65 63 38 61 65 39	c73;17f7f6ec;ae9
04BD0168	30 37 62 30 34 38 32 37 30 34 35 32 66 62 38 65	07b04;270452fb;e

Figure 82

Address	Hex	ASCII
048D0048	4E 46 52 53 00 00 00 00 00 00 00 00 00 00 00 00	NFRS.....

Figure 83

Address	Hex	ASCII
048D0048	6F 75 74 2D 6F 66 66 2D 73 70 61 63 65 00 00 00	out-off-space...

Figure 84

Address	Hex	ASCII
048D0048	30 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	0.....

Figure 85

Address	Hex	ASCII
048D0048	31 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	1.....

Figure 86

Address	Hex	ASCII
048D0048	77 5A 28 39 77 66 48 4C 22 00 00 00 00 00 00 00	wZ(9wFHL".....

Figure 87

Address	Hex	ASCII
032F6A20	2A 47 6C 6F 62 61 6C 5C 00 00 00 00 00 00 00 00	*Global\.....

Figure 88

Address	Hex	ASCII
04E76728	2A 00 2E 00 65 00 78 00 65 00 00 00 00 00 00 00	*...e.x.e.....

Figure 89

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.rdp;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%;%APPDATA%\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\

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[:]//payorgz3j6hs2gj66nk6omfw65atgmqwzxbxnqi3bv2mlwgcirunad[.]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	B1978170	RegOpenKeyExW
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	8A2AACAO	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	D46EE9FF	EqualSid
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
BBDB6B38	GetFileTime	A633633A	CryptStringToBinaryA
2236F20A	GetFileAttributesExW	C380FA58	CryptDecodeObject
4348FE4D	RemovedDirectoryW	2F9F0714	CryptImportPublicKeyInfo
E1369068	DeleteFileW	DD2C7E1F	WTSEnumerateSessionsW
3E4FB2EF	GetHandleInformation	7809AAC1	WTSQueryUserToken
784487EE	QueryFullProcessImageNameW	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		
4D05510D	GetProcAddress		