WannaCry Malware Profile | Mandiant

By Mandiant

Published: 2017-05-23 · Archived: 2026-04-05 14:17:26 UTC

Written by: Alex Berry, Josh Homan, Randi Eitzman

WannaCry (also known as WCry or WanaCryptor) malware is a self-propagating (worm-like) ransomware that spreads through internal networks and over the public internet by exploiting a vulnerability in Microsoft's Server Message Block (SMB) protocol, MS17-010. The WannaCry malware consists of two distinct components, one that provides ransomware functionality and a component used for propagation, which contains functionality to enable SMB exploitation capabilities.

The malware leverages an exploit, codenamed "EternalBlue", that was released by the Shadow Brokers on April 14, 2017.

The malware appends encrypted data files with the .WCRY extension, drops and executes a decryptor tool, and demands $300 or $600 USD (via Bitcoin) to decrypt the data.

The malware uses encrypted Tor channels for command and control (C2) communications.

File Characteristics

Filename	MD5 Hash	Size (bytes)	Compile Time	Description	Filetype
mssecsvc.exe	db349b97c37d22f5ea1d1841e3c89eb4	3723264	2010-11-20T09:03:08Z	Loader + Worm Component	EXE
tasksche.exe	84c82835a5d21bbcf75a61706d8ab549	3514368	2010-11-20T09:05:05Z	Loader	EXE
Unavailable	f351e1fcca0c4ea05fc44d15a17f8b36	65536	2009-07-14 01:12:55Z	Encryptor	DLL
@WanaDecryptor@.exe	7bf2b57f2a205768755c07f238fb32cc	245760	2009-07-13 23:19:35Z	Decryptor	EXE

Table 1: File characteristics

Persistence Mechanism

The malware creates the following two registry run keys to ensure persistence:

- Key: HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\<*Random*>

Value: <*Full_path*>\tasksche.exe

- Key: HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\<*Random*>

Value: <*Full_path*>\tasksche.exe

The malware creates the following service to ensure persistence of mssecsvc.exe:

- ServiceName: mssecsvc2.0
- DisplayName: Microsoft Security Center (2.0) Service
- BinaryPath: <path to mssecsvc> -m security

The malware creates the following service to ensure persistence of tasksche.exe

- ServiceName: <8-15lower><3number>
- DisplayName: <Same as Service Name>
- BinaryPath <path to tashsche.exe>

Host-Based Signatures

File System Artifacts

Checksum

- Actual: 0x00018AF7
- Header: 0x00000000

Dropped Files

Loader Files

- Name: tasksche.exe
  Path: C:\WINDOWS\
  Path: <*system_drive*>\ProgamData\<*sys_id*>
  Path: <*system_drive*>\Intel\<*sys_id*>
  MD5: 84c82835a5d21bbcf75a61706d8ab549
- Name: qeriuwjhrf
  Path: C:\WINDOWS\
- Name: m_bulgarian.wnry
  Path: %CD%\msg\
  MD5: 95673b0f968c0f55b32204361940d184
- Name: m_chinese (simplified).wnry
  Path: %CD%\msg\
  MD5: 0252d45ca21c8e43c9742285c48e91ad
- Name: m_chinese (traditional).wnry
  Path: %CD%\msg\
  MD5: 2efc3690d67cd073a9406a25005f7cea
- Name: m_croatian.wnry
  Path: %CD%\msg\
  MD5: 17194003fa70ce477326ce2f6deeb270
- Name: m_czech.wnry
  Path: %CD%\msg\
  MD5: 537efeecdfa94cc421e58fd82a58ba9e
- Name: m_danish.wnry
  Path: %CD%\msg\
  MD5: 2c5a3b81d5c4715b7bea01033367fcb5
- Name: m_dutch.wnry
  Path: %CD%\msg\
  MD5: 7a8d499407c6a647c03c4471a67eaad7
- Name: m_english.wnry
  Path: %CD%\msg\
  MD5: fe68c2dc0d2419b38f44d83f2fcf232e
- Name: m_filipino.wnry
  Path: %CD%\msg\
  MD5: 08b9e69b57e4c9b966664f8e1c27ab09
- Name: m_finnish.wnry
  Path: %CD%\msg\
  MD5: 35c2f97eea8819b1caebd23fee732d8f
- Name: m_french.wnry
  Path: %CD%\msg\
  MD5: 4e57113a6bf6b88fdd32782a4a381274
- Name: m_german.wnry
  Path: %CD%\msg\
  MD5: 3d59bbb5553fe03a89f817819540f469
- Name: m_greek.wnry
  Path: %CD%\msg\
  MD5: fb4e8718fea95bb7479727fde80cb424
- Name: m_indonesian.wnry
  Path: %CD%\msg\
  MD5: 3788f91c694dfc48e12417ce93356b0f
- Name: m_italian.wnry
  Path: %CD%\msg\
  MD5: 30a200f78498990095b36f574b6e8690
- Name: m_japanese.wnry
  Path: %CD%\msg\
  MD5: b77e1221f7ecd0b5d696cb66cda1609e
- Name: m_korean.wnry
  Path: %CD%\msg\

MD5: 6735cb43fe44832b061eeb3f5956b099

- Name: m_latvian.wnry
  Path: %CD%\msg\
  MD5: c33afb4ecc04ee1bcc6975bea49abe40
- Name: m_norwegian.wnry
  Path: %CD%\msg\
  MD5: ff70cc7c00951084175d12128ce02399
- Name: m_polish.wnry
  Path: %CD%\msg\
  MD5: e79d7f2833a9c2e2553c7fe04a1b63f4
- Name: m_portuguese.wnry
  Path: %CD%\msg\
  MD5: fa948f7d8dfb21ceddd6794f2d56b44f
- Name: m_romanian.wnry
  Path: %CD%\msg\
  MD5: 313e0ececd24f4fa1504118a11bc7986
- Name: m_russian.wnry
  Path: %CD%\msg\
  MD5: 452615db2336d60af7e2057481e4cab5
- Name: m_slovak.wnry
  Path: %CD%\msg\
  MD5: c911aba4ab1da6c28cf86338ab2ab6cc
- Name: m_spanish.wnry
  Path: %CD%\msg\
  MD5: 8d61648d34cba8ae9d1e2a219019add1
- Name: m_swedish.wnry
  Path: %CD%\msg\
  MD5: c7a19984eb9f37198652eaf2fd1ee25c
- Name: m_turkish.wnry
  Path: %CD%\msg\
  MD5: 531ba6b1a5460fc9446946f91cc8c94b
- Name: m_vietnamese.wnr
  Path: %CD%\msg\
  MD5: 8419be28a0dcec3f55823620922b00fa
- Name: t.wnry
  Path: %CD%
  MD5: 5dcaac857e695a65f5c3ef1441a73a8f
  Description: Encrypted Encryption Tool
- Name: taskdl.exe
  Path: %CD%
  MD5: 4fef5e34143e646dbf9907c4374276f5
  Description: Support tool for removing temporary files
- Name: taskse.exe
  Path: %CD%
  MD5: 8495400f199ac77853c53b5a3f278f3e
  Description: Support tool for launch Decryption Tool
- Name: u.wnry
  Path: %CD%
  MD5: 7bf2b57f2a205768755c07f238fb32cc
  Description: Decryption Tool
- File: b.wnry
  Path: %CD%
  MD5: c17170262312f3be7027bc2ca825bf0c
  Description: Ransom Image (BMP)
- Name: c.wnry
  Path: %CD%
  MD5: ae08f79a0d800b82fcbe1b43cdbdbefc
  Description: Config Data

Encryptor Files

- 00000000.res
- 00000000.pky
- 00000000.eky

- 00000000.dky

Decryptor Files

- c.wnry
- File: taskhsvc.exe
  Path: TaskData\Tor\

The following artifact can be found on remotely exploited systems:

- Name: mssecsvc.exe
  Path: C:\WINDOWS\
  MD5: db349b97c37d22f5ea1d1841e3c89eb4
  Description: Dropper + worm component

Registry Artifacts

- ServiceName: mssecsvc2.0
  DisplayName: Microsoft Security Center (2.0) Service
  BinaryPath: <*GetModuleFileName*> -m security
- HKLM\Software\WanaCrypt0r\wd
- HKCU\Software\WanaCrypt0r\wd

Exports

- 0x00005AE0 TaskStart

Mutex

- MsWinZonesCacheCounterMutexA

Process Arguments

- icacls . /grant Everyone:F /T /C /Q
- attrib +h +s <Drive_Letter>:\$RECYCLE
- taskkill.exe /f /im Microsoft.Exchange.*
- taskkill.exe /f /im MSExchange*
- taskkill.exe /f /im sqlserver.exe
- taskkill.exe /f /im sqlwriter.exe
- taskkill.exe /f /im mysqld.exe
- cmd.exe /c start /b @WanaDecryptor@.exe vs
- cmd.exe /c vssadmin delete shadows /all /quiet & wmic shadowcopy delete & bcdedit /set {default} bootstatuspolicy ignoreallfailures & bcdedit /set {default} recoveryenabled no & wbadmin delete catalog -q
- -m security
- cmd /c <15 digits>.bat
- cscript.exe //nologo <1 character>.vbs

Network-Based Signatures

DNS

- www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com (sinkholed)

Connections

- <*random_ip*>:445
- <*subnet_ip*>:445

WannaCry Analysis

Startup

The malware starts by attempting to connect to the following domain with InternetOpenUrl:

- www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com

NOTE: If this succeeds, the malware immediately exits. For a list of observed killswitch domains, see Appendix A.

If the connection fails, however, the malware checks the number of arguments passed to the program. If zero, the malware continues with installation; otherwise it enters service mode.

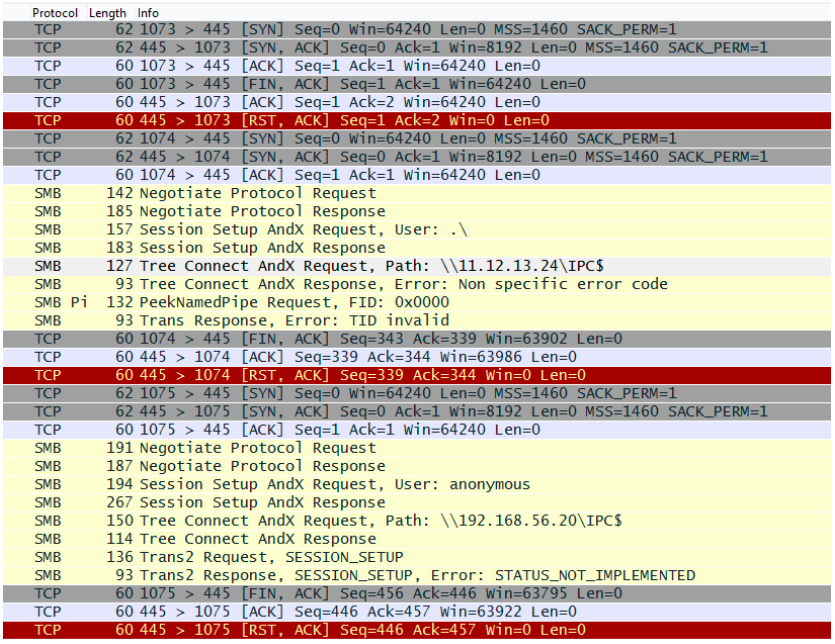
Note: Network proxies and other enterprise network security features may prevent the malware from contacting its killswitch domain and inadvertently trigger encryption. Organizations may wish to adjust their proxy configurations or other network configurations to avoid this problem.

Service Mode

In service mode, the malware first updates the service config so that failure actions occur if the service exits without entering a SERVICE_STOPPED state. The malware then executes the service function, which registers the service handlers and attempts exploitation of MS17-010 against identified SMB services. This allows remote code execution and enables spreading across the network. This execution is performed in a thread, and the service exits after 24 hours regardless of the status of the thread.

The spreader begins by setting up the Windows socket APIs and generating a RSA crypto context. This crypto context is later used to generate random numbers. The malware then builds two DLLs in memory – they are 32 and 64-bit DLLs that have identical functionality. Each one contains a single export named *PlayGame* that loads the W resource, writes it to C:\WINDOWS\mssecsvc.exe, and executes it. The W resource in each case has been populated with a copy of the running binary (MD5: db349b97c37d22f5ea1d1841e3c89eb4).

The malware continues by spawning two threads, the first thread enumerates the network adapters and determines which subnets the system is on. The malware then generates a thread for each IP on the subnet. Each of these threads attempts to connect to the IP on port 445 and, if successful, attempts exploitation of the service via a vulnerability described in MS17-010. An example of an attempt to exploit MS17-010 on a remote system can be seen in Figure 1.



Protocol	Length	Info
TCP	62	1073 > 445 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
TCP	62	445 > 1073 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 SACK_PERM=1
TCP	60	1073 > 445 [ACK] Seq=1 Ack=1 Win=64240 Len=0
TCP	60	1073 > 445 [FIN, ACK] Seq=1 Ack=1 Win=64240 Len=0
TCP	60	445 > 1073 [ACK] Seq=1 Ack=2 Win=64240 Len=0
TCP	60	445 > 1073 [RST, ACK] Seq=1 Ack=2 Win=0 Len=0
TCP	62	1074 > 445 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
TCP	62	445 > 1074 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 SACK_PERM=1
TCP	60	1074 > 445 [ACK] Seq=1 Ack=1 Win=64240 Len=0
SMB	142	Negotiate Protocol Request
SMB	185	Negotiate Protocol Response
SMB	157	Session Setup AndX Request, User: .\
SMB	183	Session Setup AndX Response
SMB	127	Tree Connect AndX Request, Path: \\11.12.13.24\IPC$
SMB	93	Tree Connect AndX Response, Error: Non specific error code
SMB Pi	132	PeekNamedPipe Request, FID: 0x0000
SMB	93	Trans Response, Error: TID invalid
TCP	60	1074 > 445 [FIN, ACK] Seq=343 Ack=339 Win=63902 Len=0
TCP	60	445 > 1074 [ACK] Seq=339 Ack=344 Win=63986 Len=0
TCP	60	445 > 1074 [RST, ACK] Seq=339 Ack=344 Win=0 Len=0
TCP	62	1075 > 445 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
TCP	62	445 > 1075 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 SACK_PERM=1
TCP	60	1075 > 445 [ACK] Seq=1 Ack=1 Win=64240 Len=0
SMB	191	Negotiate Protocol Request
SMB	187	Negotiate Protocol Response
SMB	194	Session Setup AndX Request, User: anonymous
SMB	267	Session Setup AndX Response
SMB	150	Tree Connect AndX Request, Path: \\192.168.56.20\IPC$
SMB	114	Tree Connect AndX Response
SMB	136	Trans2 Request, SESSION_SETUP
SMB	93	Trans2 Response, SESSION_SETUP, Error: STATUS_NOT_IMPLEMENTED
TCP	60	1075 > 445 [FIN, ACK] Seq=456 Ack=446 Win=63795 Len=0
TCP	60	445 > 1075 [ACK] Seq=446 Ack=457 Win=63922 Len=0
TCP	60	445 > 1075 [RST, ACK] Seq=446 Ack=457 Win=0 Len=0

Figure 1: WannaCry network traffic attempting SMB exploit

One of the unique features of this traffic is an SMB Tree Connect AndX Request containing the following UNICODE string:

- \\192.168.56.20\IPC$

This packet is hand-crafted and hard-coded into the malware.

The second thread generates random IPs and attempts to connect to them on port 445. If the connection is successful, the malware then attempts to perform the SMB attack on the system. 128 instances of the second thread area created with two seconds separating each thread creations.

Installation

The malware continues by creating a service named *mssecsvc2.0* with a binary path pointing to the running module with the arguments "-m security". Once created, the malware starts the service. The malware then locates its R resource and loads it into memory. The malware then writes the R resource data to the file C:\WINDOWS*tasksche.exe*. The malware executes

C:\WINDOWS\tasksche.exe /i with the CreateProcess API. The malware then attempts to move C:\WINDOWS*tasksche.exe* to C:\WINDOWS*qeriuwjhrf*, replacing the original file if it exists.

The malware begins by generating a unique identifier based on the computer name. The identifier, <*sys_id*>, has the form of 8-15 random lowercase characters followed by 3 numbers. The malware then checks to see if it was passed the /i argument.

Run with /i Command

The /i command copies the running binary to <system_drive>\ProgamData\<*sys_id*>*tasksche.exe* if <*system_drive*>\ProgamData exists, otherwise it will be copied to <*system_drive*>\Intel\<*sys_id*>*tasksche.exe*. <*system_drive*>is the drive letter on which Windows was installed (C:\ for C:\Windows). The malware then updates its current directory to the created directory.

The malware then attempts to open the service named <*sys_id*>. If it does not exist, the malware creates it with a DisplayName of <*sys_id*> and a BinaryPath of cmd /c <*path_to_copied tasksche.exe*>. The malware then starts the service. The malware attempts to open the mutex Global\MsWinZonesCacheCounterMutexA0. If the mutex is not created within 60 seconds, the malware re-launches itself from the new installation directory with no arguments. The malware then waits 60 seconds for the mutex to be created. If the mutex is created in either instance, the initial executable exits. If the mutex fails to be created, the malware continues as if it was run without the /i argument.

Run without /i Command

The malware updates %CD% to the path of the running module and sets HKLM\Software\WanaCrypt0r\wd to %CD%. The malware then loads the XIA resource and decompresses numerous files (see Table 3) to %CD%. The malware then opens %CD%*c.wnry* (the configuration data) and loads it into memory. It expects the file to be of size 0x30C. The malware then chooses randomly between the three strings 13AM4VW2dhxYgXeQepoHkHSQuy6NgaEb94, 12t9YDPgwueZ9NyMgw519p7AA8isjr6SMw, and 115p7UMMngoj1pMvkpHijcRdfJNXj6LrLn; writes it to offset 0xB2 in the configuration file; and writes the updated configuration data back to %CD%*c.wnry*.

The malware then sets the hidden attribute for %CD% by executing the following command with CreateProcess:

- attrib +h

The malware then executes the following command – granting all users permissions to %CD% and all of its subdirectories:

- icacls . /grant Everyone:F /T /C /Q

The malware then imports the hard-coded RSA Private key, shown in Figure 2.

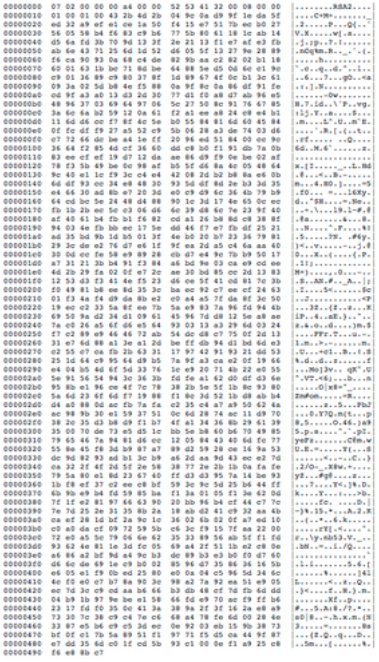


Figure 2: Imported private key

The malware then opens and reads %CD%*t.wnry*. The first 8 bytes of the file are checked to match the magic value WANACRY!. The file has the following structure:

```
struct T_WNRY {
  char magic[8]; // must match WANACRY!
  uint32_t enc_keylen; // needs to be 0x100
  char enc_key[enc_keylen];
  uint32_t unknown;  // was 4
  uint64_t enc_datalen;
  char enc_data[enc_data_len];
}
```

The encrypted key decrypts to the 128-bit AES key BEE19B98D2E5B12211CE211EECB13DE6. This key can then be used to decrypt the enc_data. The decrypted data is saved as a DLL (MD5: f351e1fcca0c4ea05fc44d15a17f8b36). This DLL is then manually loaded into memory and the *TaskStart* export is called. The TaskStart export of the decrypted DLL is the encryption component of the ransomware.

XIA Resource Contents

The files shown in Table 2 are extracted from the XIA resource. They are dropped into the %CD% of the running malware.

Filename	MD5 Hash	Description
r.wnry	3e0020fc529b1c2a061016dd2469ba96	Text ransom note
s.wnry	ad4c9de7c8c40813f200ba1c2fa33083	Zip file containing Tor files
t.wnry	5dcaac857e695a65f5c3ef1441a73a8f	Encrypted encryption tool
taskdl.exe	4fef5e34143e646dbf9907c4374276f5	*.WNCRYT file deletion tool
taskse.exe	8495400f199ac77853c53b5a3f278f3e	Utility used to launch decryption tool
u.wnry	7bf2b57f2a205768755c07f238fb32cc	Decryption tool
b.wnry	c17170262312f3be7027bc2ca825bf0c	Ransom image (BMP)
c.wnry	ae08f79a0d800b82fcbe1b43cdbdbefc	Configuration data

Table 2: XIA extracted resources

Table 3 shows RTF documents containing the ransom note in various languages.

Filename	MD5 Hash
m_bulgarian.wnry	95673b0f968c0f55b32204361940d184
m_chinese (simplified).wnry	0252d45ca21c8e43c9742285c48e91ad
m_chinese (traditional).wnry	2efc3690d67cd073a9406a25005f7cea
m_croatian.wnry	17194003fa70ce477326ce2f6deeb270
m_czech.wnry	537efeecdfa94cc421e58fd82a58ba9e
m_danish.wnry	2c5a3b81d5c4715b7bea01033367fcb5
m_dutch.wnry	7a8d499407c6a647c03c4471a67eaad7
m_english.wnry	fe68c2dc0d2419b38f44d83f2fcf232e
m_filipino.wnry	08b9e69b57e4c9b966664f8e1c27ab09
m_finnish.wnry	35c2f97eea8819b1caebd23fee732d8f
m_french.wnry	4e57113a6bf6b88fdd32782a4a381274

m_german.wnry	3d59bbb5553fe03a89f817819540f469
m_greek.wnry	fb4e8718fea95bb7479727fde80cb424
m_indonesian.wnry	3788f91c694dfc48e12417ce93356b0f
m_italian.wnry	30a200f78498990095b36f574b6e8690
m_japanese.wnry	b77e1221f7ecd0b5d696cb66cda1609e
m_korean.wnry	6735cb43fe44832b061eeb3f5956b099
m_latvian.wnry	c33afb4ecc04ee1bcc6975bea49abe40
m_norwegian.wnry	ff70cc7c00951084175d12128ce02399
m_polish.wnry	e79d7f2833a9c2e2553c7fe04a1b63f4
m_portuguese.wnry	fa948f7d8dfb21ceddd6794f2d56b44f
m_romanian.wnry	313e0ececd24f4fa1504118a11bc7986
m_russian.wnry	452615db2336d60af7e2057481e4cab5
m_slovak.wnry	c911aba4ab1da6c28cf86338ab2ab6cc
m_spanish.wnry	8d61648d34cba8ae9d1e2a219019add1
m_swedish.wnry	c7a19984eb9f37198652eaf2fd1ee25c
m_turkish.wnry	531ba6b1a5460fc9446946f91cc8c94b
m_vietnamese.wnry	8419be28a0dcec3f55823620922b00fa

Table 3: Ransom notes in various languages

Encryption Component

The *TaskStart* export takes two arguments; the handle to the module and an integer that must be zero. *TaskStart* first creates a mutex named "MsWinZonesCacheCounterMutexA" and reads the contents of *c.wnry* from the current directory. If the mutex exists or *c.wnry* is not present, the malware exits. The malware creates another mutex named "Global\MsWinZonesCacheCounterMutexA0".

The malware then loads and verifies a key from the file *00000000.dky*. The malware then attempts to load a key *00000000.pky*. If the key does not exist, the malware imports a public RSA key (seen in Figure 3), generates a new 2048-bit RSA key and saves the public key to *00000000.pky*. The malware then saves the generated private key to *00000000.eky*, encrypted with the embedded public key.

```
00000000  06 02 00 00 00 a4 00 00  52 53 41 31 00 08 00 00 ........RSA1....
00000010  01 00 01 00 75 97 4c 3b  84 46 de 2c 2a f4 95 a8 ....u.L;.F.,*...
00000020  5d c0 cd 6d da d7 d4 92  1e 13 82 34 6a 70 8d 8f ]..m.......4jp..
00000030  7c f7 04 92 55 7f f1 a2  27 b2 9e 41 ac 90 80 91 |...U...'..A....
00000040  18 93 c2 b1 7b ad 2b f3  ff af db 2b 51 be 1d a3 ....{.+....+Q...
00000050  27 e3 a7 57 08 5a be c1  1d f6 04 f8 1c be 5b b1 '..W.Z........[.
00000060  67 fb e4 c8 da 75 00 70  b1 17 70 24 6c 09 63 74 g....u.p..p$l.ct
00000070  ac 4b 0a 1d 71 ae 7f ae  65 b8 c5 86 79 c5 7e 9f .K..q...e...y.~.
00000080  98 60 4c 52 b9 29 62 cb  23 29 ed 31 91 74 7b 7b .`LR.)b.#).1.t{{
00000090  0b 26 1b f2 7d 67 bf da  7a 40 da f2 61 4d 94 a5 .&..}g..z@..aM..
000000A0  7d ad 59 6b ad 9e a3 3a  39 c6 5b 6e 9f d2 bb 36 }.Yk...:9.[n...6
000000B0  b5 f5 d2 65 f5 2c 30 d8  c1 17 bd af 28 00 96 20 ...e.,0.....(..
000000C0  46 a7 2d 62 03 0c d7 d0  75 a0 0b 07 ea d4 1f ca F.-b....u.......
000000D0  e8 d9 4e db 38 f2 26 75  cb 12 a6 88 70 9b e1 ea ..N.8.&u....p...
000000E0  32 dc f8 71 72 50 41 e6  17 81 68 27 42 8e df e5 2..qrPA...h'B...
000000F0  de a1 72 d9 3b fb e5 9d  30 11 69 92 cd 60 2b e2 ..r.;...0.i..`+.
00000100  d5 46 3c 28 cf 9d 30 4a  f7 ad b9 fb 0f 91 fe 2e .F<(..0J........
00000110  be 18 f1 ce                                      ....
```

The *00000000.eky* starts with the number of bytes in little endian (0x500) followed by the encrypted key.

The malware launches a thread that writes 136 bytes to *00000000.res* every 25 seconds. The buffer written includes the current time of the system. If the file *00000000.res* does not exist while the malware is initializing, it creates the file. The initial contents begins with eight randomly generated bytes followed by 128 zero bytes.

The malware launches another thread that verifies it can encrypt and decrypt using the keys contained in *00000000.dky* and *00000000.pky* every 25 seconds. If the decryption is successful, the malware sets a global flag that stops the encryption process.

The malware launches another thread that scans for new drives attached to the system every three seconds. If a new drive is attached to the system and is not identified as a type CDROM drive, the malware begins the encryption process on the new drive. On new drives attached to the system, the malware may create the directory <*Drive_letter*>:\$RECYCLE and execute the following command:

- attrib +h +s <*Drive_Letter*>:\$RECYCLE

The malware creates a thread that executes the process *taskdl.exe* every 30 seconds.

and creates another thread that executes either of the following two binaries (depending on administrator permissions and if the malware is running at system level):

- @WanaDecryptor@.exe
- taskse.exe <*Full_Path*>\@WanaDecryptor@.exe

A registry key name starting with 8 to 15 characters between 'a' and 'z' followed by three random values between '0' and '9' is then generated by the malware. It may then create the following registry paths with the generated key name:

- HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\<Key>
- HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\<Key>

To create the registry key, the malware executes the following command:

- cmd.exe /c reg add <*Registry_Ru_Path*> /v "<*Random*>" /t REG_SZ /d "\"<*Full_Path*>\tasksche.exe\"" /f

User File Encryption

The malware loads another embedded RSA public key shown in Figure 4.

```
00000000  06 02 00 00 00 a4 00 00  52 53 41 31 00 08 00 00 ........RSA1....
00000010  01 00 01 00 43 2b 4d 2b  04 9c 0a d9 9f 1e da 5f ....C+M+.......
00000020  ed 32 a9 ef e1 ce 1a 50  f4 15 e7 51 7b ec b0 27 .2.....P...Q{..'
00000030  56 05 58 b4 f6 83 c9 b6  77 5b 80 61 18 1c ab 14 V.X.....w[.a....
00000040  d5 6a fd 3b 70 9d 13 3f  2e 21 13 f1 e7 af e3 fb .j.;p..?.!......
00000050  ab 6e 43 71 25 6d 1d 52  d6 05 5f 13 27 9e 28 89 .nCq%m.R.._.'.(.
00000060  f6 ca 90 93 0a 68 c4 de  82 9b aa c2 82 02 b1 18 .....h..........
00000070  60 01 63 1b bc 71 8d be  64 88 5e d5 0d 6c c1 9c `.c..q..d.^..l..
00000080  c9 01 36 89 c9 80 37 8f  1d 89 67 4f 0c b1 3c 61 ..6...7...gO..<a
00000090  09 3a 02 5d b8 4e f5 88  0a 9f 8c 0a 86 df 91 fe .:.].N..........
000000A0  cd 9f a3 a0 13 d3 2d 30  77 d1 f0 a8 d7 ab 96 e5 ......-0w.......
000000B0  48 96 37 03 69 64 97 06  5c 27 50 8c 91 76 67 85 H.7.id..\'P..vg.
000000C0  3a 6c 6a b2 59 12 0a 61  f2 a1 ee a8 24 c8 e4 b1 :lj.Y..a....$...
000000D0  11 6d d6 cc f7 8f 4c 5e  b0 55 84 81 6d 60 45 84 .m....L^.U..m`E.
000000E0  0f fc df f9 27 a5 52 c9  5b 06 28 a3 de 74 03 d6 ....'.R.[.(..t..
000000F0  c7 72 66 dc be a4 1e ff  20 96 ed 51 84 00 cc 9c .rf..... ..Q....
00000100  36 64 f2 85 4d cf 36 60  dd c8 b0 f1 91 db 7a 0b 6d..M.6`......z.
00000110  83 ee cf ef                                      ....
```

Figure 4: Additional embedded RSA public key

The malware executes the file *@WanaDecryptor@.exe* with the argument "fi". This appears to be an initial check-in with the server and the response may contain an updated bitcoin address. The malware updates *c.wnry* with the current time at offset 0x60.

The malware then copies *u.wrny* to *@WanaDecryptor@.exe* and executes the script shown in Figure 5 to create *@WanaDecryptor@.exe.lnk*. The script is saved to a randomly generated filename based on the current time and a random value using characters from '0' to '9'. Example filename: "188391494652743.bat".

```
@echo off
echo SET ow = WScript.CreateObject("WScript.Shell")> m.vbs
echo SET om = ow.CreateShortcut("[Full Path]\@WanaDecryptor@.exe.lnk")>> m.vbs
echo om.TargetPath = "[Full Path]\@WanaDecryptor@.exe">> m.vbs
echo om.Save>> m.vbs
cscript.exe //nologo m.vbs
del m.vbs

del /a %0
```

Figure 5: WannaCry internal script for moving and deleting files

The malware then writes either "$<*Value*>worth of bitcoin" or "%.<*Value*> BTC" depending on the configuration – followed by the contents of the file *r.wnry* to *@Please_Read_Me@.txt*, which reads as follows:

Q: What's wrong with my files?

A: Ooops, your important files are encrypted. It means you will not be able to access them anymore until they are decrypted.

If you follow our instructions, we guarantee that you can decrypt all your files quickly and safely!

Let's start decrypting!

Q: What do I do?

A: First, you need to pay service fees for the decryption.

Please send <*Ransom Amount*> to this bitcoin address: <*Bitcoin_address*>

Next, please find an application file named "@WanaDecryptor@.exe". It is the decrypt software.

Run and follow the instructions! (You may need to disable your antivirus for a while.)

Q: How can I trust?

A: Don't worry about decryption.

We will decrypt your files surely because nobody will trust us if we cheat users.

* If you need our assistance, send a message by clicking <Contact Us> on the decryptor window.

Figure 6: Encryption warning displayed to user

The malware then targets files on the user's desktop and documents folders. When the malware starts scanning a directory it creates a temporary file with the prefix "~SD", and deletes it if successful.

When selecting which files to encrypt, the malware skips over files with .exe, .dll, and .wncry extensions. The files with the extensions shown in Figure 7 are selected for encryption. Files larger than 209,715,200 bytes may also be encrypted.

```
.der, .pfx, .key, .crt, .csr, .p12, .pem, .odt, .ott, .sxw, .stw, .uot, .3ds, .max, .3dm,
.ods, .ots, .sxc, .stc, .dif, .slk, .wb2, .odp, .otp, .sxd, .std, .uop, .odg, .otg, .sxm,
.mml, .lay, .lay6, .asc, .sqlite3, .sqlitedb, .sql, .accdb, .mdb, .db, .dbf, .odb, .frm, .myd,
.myi, .ibd, .mdf, .ldf, .sln, .suo, .cs, .c, .cpp, .pas, .h, .asm, .js, .cmd, .bat, .ps1,
.vbs, .vb, .pl, .dip, .dch, .sch, .brd, .jsp, .php, .asp, .rb, .java, .jar, .class, .sh, .mp3,
.wav, .swf, .fla, .wmv, .mpg, .vob, .mpeg, .asf, .avi, .mov, .mp4, .3gp, .mkv, .3g2, .flv,
.wma, .mid, .m3u, .m4u, .djvu, .svg, .ai, .psd, .nef, .tiff, .tif, .cgm, .raw, .gif, .png,
.bmp, .vcd, .iso, .backup, .zip, .rar, .7z, .gz, .tgz, .tar, .bak, .tbk, .bz2, .PAQ, .ARC,
.aes, .gpg, .vmx, .vmdk, .vdi, .sldm, .sldx, .sti, .sxi, .602, .hwp, .edb, .potm, .potx,
.ppam, .ppsx, .ppsm, .pps, .pot, .pptm, .xltm, .xltx, .xlc, .xlm, .xlt, .xlw, .xlsb, .xlsm,
.dotx, .dotm, .dot, .docm, .docb, .jpg, .jpeg, .snt, .onetoc2, .dwg, .pdf, .wk1, .wks, .123,
.rtf, .csv, .txt, .vsdx, .vsd, .eml, .msg, .ost, .pst, .pptx, .ppt, .xlsx, .xls, .docx, .doc
```

Figure 7: Files targeted for encryption

The malware may ignore folders with the following names:

- \\
- $\
- Intel
- ProgramData
- WINDOWS
- Program Files
- Program Files (x86)
- AppData\Local\Temp
- Local Settings\Temp
- Temporary Internet Files
- Content.IE5

The malware will also compare folder names with the following string, and avoid encryption if identified:

- " This folder protects against ransomware. Modifying it will reduce protection"

Note: The string contains a leading whitespace. This particular check is likely included for testing/development purposes.

When a directory contains a file that will be encrypted, the malware copies *@Please_Read_Me@.txt* and *@WanaDecryptor@.exe* to the directory. It verifies that the first eight bytes do not contain the string *WANACRY!* and performs additional checks on the header to verify the file is not already encrypted.

The files are encrypted with a randomly generated 128-bit AES key in CBC mode with a NULL initialization vector. The key is generated per file, is encrypted with the generated RSA public key, and included in the encrypted file header. Each file encrypted by the malware starts with the string *WANACRY!* and has the *WNCRY* extension. Depending on the file properties, the malware may also stage files in a *WNCRYT* extension.

Table 4 shows the file format of encrypted files.

Offset	**Value**
0x0000	WANACRY!
0x0008	Length of RSA encrypted data
0x000C	RSA encrypted AES file encryption key
0x010C	File type internal to WannaCry
0x0110	Original file size
0x0118	Encrypted file contents (AES-128 CBC)

Table 4: Encrypted file format

When encrypting the AES key with RSA, the malware may use the embedded RSA key or a key randomly generated. If the file f.wnry does not exist during initialization, the malware generates a random number if the file size is less than 209,715,200 bytes. If the number is a multiple of 100, the malware uses the embedded RSA key to encrypt the AES key. A maximum of ten files can be encrypted with this key. When an AES key is encrypted with this RSA key, the malware writes the file path to the file f.wnry. If the random number is not a multiple of 100 or the file f.wnry already exists on the system, the malware will encrypt the AES key with the randomly generated RSA key.

Once the malware completes encrypting the desktop and documents folders, it executes the following commands:

- taskkill.exe /f /im Microsoft.Exchange.*
- taskkill.exe /f /im MSExchange*
- taskkill.exe /f /im sqlserver.exe
- taskkill.exe /f /im sqlwriter.exe
- taskkill.exe /f /im mysqld.exe

The malware then encrypts files found on logical drives attached to the system that are not type DRIVE_CDROM.

The malware may execute the command:

- @WanaDecryptor@.exe co

The malware executes the command:

- cmd.exe /c start /b @WanaDecryptor@.exe vs

The malware will copy *b.wnry* to *@WanaDecryptor@.bmp* and place it in each user's desktop folder, as well as a copy of *@WanaDecryptor@.exe*.

Decryptor Component

The malware communicates with an Onion server using a Tor server running on local host TCP port 9050. The malware registers the system with the Onion server, transferring encryption keys and deleting volume shadows. Once the ransom is paid, the malware obtains the decrypted RSA private key from the Onion server and decrypts ransomed files.

It first attempts to read the contents of the registry path HKLM\Software\WanaCrypt0r\wd. If this fails, the malware attempts to read the contents from a similar registry path within the HKCU registry hive. If one of the registry paths exists, the malware sets the current directory to value read from the registry.

The malware attempts to open *c.wnry* from the current directory and read 780 bytes if it exists. If the file does not exist, the file is created with the contents shown in Figure 8.

```
00000000  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
[Null bytes omitted]
00000060  00 00 00 00 00 00 00 00 00 00 00 00 42 03 14 59  ............B..Y
00000070  00 00 00 00 00 00 00 00 00 00 96 43 00 00 00 00  ..........–C....
[Null bytes omitted]
000000B0  00 00 31 33 41 4D 34 56 57 32 64 68 78 59 67 58  ..13AM4VW2dhxYgX
000000C0  65 51 65 70 6F 48 6B 48 53 51 75 79 36 4E 67 61  eQepoHkHSQuy6Nga
000000D0  45 62 39 34 00 00 00 00 00 00 00 00 00 00 00 00  Eb94............
[Null bytes omitted]
00000300  00 00 00 00 00 00 00 00 00 00 00 00              ............
```

Figure 8: Contents of c.wnry

The value at offset 0x6c (0x59140342) in *c.wnry* is the timestamp the file was created. The remaining values are hardcoded within the binary.

Accepted Commands

The decryptor component accepts the command line arguments shown in Table 5.

Argument	Description
fi	Connects to an Onion server sending details from the system including the host name, user name and eight bytes from 00000000.res. The response may include a Bitcoin address that is updated in *c.wnry*.
co	Appears to be an initial check-in with the ransom server without displaying the ransom interface.
vs	Deletes volume shadow copies using the vssadmin utility.

Table 5: Accepted commands

fi Argument

The malware reads 136 bytes from the file "00000000.res" in the current path. If the file does not exist the malware exits. The malware reads two URLs from *c.wnry* at offsets 0x242 and 0x1DE.

The first URL at offset 0x1DE in *c.wnry* is:

- *https://dist.torproject.org/torbrowser/6.5.1/tor-win32-0.2.9.10.zip*

The alternate URL at offset 0x242 is not configured.

The malware then binds a TCP socket to the localhost (127.0.0.1) and connects to port 9050 on the localhost.

The malware then checks if the path "TaskData\Tor*taskhsvc.exe*" exists. If the file does not exist it is extracted from the archive *s.wnry*. If *s.wnry* does not exist, the malware downloads the first URL in the configuration – and if this fails it attempts the second.

When downloading from a URL, the downloaded file is first saved to a filename generated with GetTempFileNameA with a "t" prefix within the TaskData folder. The downloaded file is a Zip archive that is extracted to the "TaskData" folder.

Once extracted, the malware copies "TaskData\Tor*tor.exe*" to "TaskData\Tor*taskhsvc.exe*" and executes it.

The malware parses the string obtained at offset 0xE4 in the configuration file *c.wnry* for Onion servers to connect to. The Onion servers listed in the configuration file are as follows:

- gx7ekbenv2riucmf.onion
- 57g7spgrzlojinas.onion
- xxlvbrloxvriy2c5.onion
- 76jdd2ir2embyv47.onion
- cwwnhwhlz52maqm7.onion

The malware sends the first eight bytes of the file *00000000.res*, the host name, user name and the string "+++" to the Onion server. The command and control protocol appears to be custom and XOR encoded with a randomly generated buffer.

The response from the server is added to *c.wnry* if the string is 30 to 50 characters in length. The following is an example message sent to the server:

- *<8 bytes from 00000000.res><Host name>*\x00*<Unknown Byte><User name>*\x00+++

co Argument

This argument the malware scans for file names in the format *<8_Uppercase_Hex>*.res. The file the malware is likely looking for is *00000000.res* that is created by the encryption DLL. The malware then generates a C2 message containing four values (Table 6) obtained from the ".res" file in the following format:

- --- *<Time0> <Time1> <Unknown_int0> <Unknown_long> <Index>*

Note: In the aforementioned example, the values are separated with a TAB character.

Value	**Description**
---	Hard-coded string likely intended to identify the command
Time0	Time value obtained from offset 0x60
Time1	Time value obtained from offset 0x78
Unknown int0	Integer obtained from offset 0x7C
Unknown long	64-bit Integer obtained from offset 0x80
Index	Count of the current file when scanning for files in the format *<8_Uppercase_Hex>*.res

Table 6: C2 message values

Figure 9 shows an example of a message.

```
00000000  aa aa bb bb 12 34 56 78  57 37 58 36 34 5f 41 4e .....4VxW7X64_AN
00000010  41 4c 59 53 49 53 00 0b  52 45 00 2d 2d 2d 09 32 ALYSIS..RE.---.2
00000020  30 31 37 2d 30 35 2d 31  33 20 30 35 3a 31 35 3a 017-05-13 05:15:
00000030  35 35 09 32 30 30 36 2d  30 34 2d 30 35 20 30 31 55.2006-04-05 01
00000040  3a 34 39 3a 30 35 09 31  32 34 09 31 32 38 09 31 :49:05.124.128.1
00000050  00                                              .
```

Figure 9: Sample C2 message

After sending the message, the malware exits.

vs Argument

The malware sleeps for 10 seconds and then executes the following command using CreateProcess or RunAs (depending on group membership):

- cmd.exe /c vssadmin delete shadows /all /quiet & wmic shadowcopy delete & bcdedit /set {default} bootstatuspolicy ignoreallfailures & bcdedit /set {default} recoveryenabled no & wbadmin delete catalog -q

No Argument

The malware copies *b.wnry* from the current directory to the desktop with the filename *@WanaDecryptor@.bmp*. The desktop wallpaper is then set to the path of the bitmap and the dialog shown in Figure 6 is then displayed.

When the user clicks on the "Contact us" link, the malware sends the message to the Onion server using the following format:

- *<8 bytes from 00000000.res><Host name>*\x00*<Unknown Byte><User name>*\x00***<Tab><Message contents>*

Depending on the response from the server, the malware may display a message box with one of the following values:

1. Your message has been sent successfully!
2. Failed to send your message!
   Please make sure that your computer is connected to the Internet and
   your Internet Service Provider (ISP) does not block connections to the TOR Network!
3. You are sending too many mails! Please try again <Integer value> minutes later.

When the user clicks on "Check Payment". The malware first check if the file *00000000.dky* is present on the system. If the file is present, it attempts to verify the key by encrypting a file with the key obtained from *00000000.pky* and decrypting it with the key obtained from *00000000.dky*.

If the file is not present, the malware sends the contents of *00000000.eky* to the Onion server. The response from the server is saved to *00000000.dky*. If the key cannot be validated, the malware displays a message box with the contents:

You did not pay or we did not confirmed your payment!
Pay now if you didn't and check again after 2 hours.
Best time to check: 9:00am - 11:00am GMT from Monday to Friday.

When the decrypt button is clicked without the ransom being paid, the malware decrypts the files listed in *f.wnry*. The files listed in *f.wnry* are those randomly selected to be encrypted with the embedded public key. This process is covered in the Encryption component section above.

Unique Strings

mssecsvc.exe

(MD5: db349b97c37d22f5ea1d1841e3c89eb4)

- SMBr
- PC NETWORK PROGRAM 1.0
- LANMAN1.0
- Windows for Workgroups 3.1a
- LM1.2X002
- LANMAN2.1
- NT LM 0.12
- SMBs
- Windows 2000 2195
- Windows 2000 5.0
- SMBu
- __USERID__PLACEHOLDER__@
- \\172.16.99.5\IPC$
- __TREEID__PLACEHOLDER__
- __USERID__PLACEHOLDER__@
- SMB3
- __TREEID__PLACEHOLDER__
- __USERID__PLACEHOLDER__@
- \t
- h6agLCqPqVyXi2VSQ8O6Yb9ijBX54jY6KM+sz33NmS6TK8XlOk920s0E0aajOV++wrR92ds1FOLBO+evLPj4sIvAjLvaLdgk8+BlNZs8PMa9b
- h5DH0RqsyNfEbXNTxRzla1zNfWz0bB4fqzrdNNfNXvtTv9FWqyXCEHLhOz9p7JXzJBBUd0OR9rg8DFXIyNXMHCfeX5v/YjDkYmaBrFWuC
- SMB3
- __TREEID__PLACEHOLDER__
- __USERID__PLACEHOLDER__@
- userid
- treeid
- __TREEPATH_REPLACE__
- \\%s\IPC$
- Microsoft Base Cryptographic Provider v1.0
- %d.%d.%d.%d
- mssecsvc2.0
- Microsoft Security Center (2.0) Service
- %s -m security
- C:\%s\qeriuwjhrf
- C:\%s\%s
- WINDOWS
- tasksche.exe
- CloseHandle
- WriteFile
- CreateFileA
- CreateProcessA
- 32.dll
- http://www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com

tasksche.exe

(MD5: 84c82835a5d21bbcf75a61706d8ab549)

- .der .pfx .key .crt .csr .p12 .pem .odt .ott .sxw .stw .uot .3ds .max .3dm .ods .ots .sxc .stc .dif .slk .wb2 .odp .otp .sxd .std .uop .odg .otg .sxm .mml .lay .lay6 .asc .sqlite3 .sqlitedb .sql .accdb .mdb .db .dbf .odb .frm .myd .myi .ibd .mdf .ldf .sln .suo .cs .c .cpp .pas .h .asm .js .cmd .bat .ps1 .vbs .vb .pl .dip .dch .sch .brd .jsp .php .asp .rb .java .jar .class .sh .mp3 .wav .swf .fla .wmv .mpg .vob .mpeg .asf .avi .mov .mp4 .3gp .mkv .3g2 .flv .wma .mid .m3u .m4u .djvu .svg .ai .psd .nef .tiff .tif .cgm .raw .gif .png .bmp .jpg .jpeg .vcd .iso .backup .zip .rar .7z .gz .tgz .tar .bak .tbk .bz2 .PAQ .ARC .aes .gpg .vmx .vmdk .vdi .sldm .sldx .sti .sxi .602 .hwp .snt .onetoc2 .dwg .pdf .wk1 .wks .123 .rtf .csv .txt .vsdx .vsd .edb .eml .msg .ost .pst .potm .potx .ppam .ppsx .ppsm .pps .pot .pptm .pptx .ppt .xltm .xltx .xlc .xlm .xlt .xlw .xlsb .xlsm .xlsx .xls .dotx .dotm .dot .docm .docb .docx .doc
- WANACRY!
- %s\\%s
- %s\\Intel
- %s\\ProgramData
- cmd.exe /c \"%s\"
- XIA

- 115p7UMMngoj1pMvkpHijcRdfJNXj6LrLn
- 12t9YDPgwueZ9NyMgw519p7AA8isjr6SMw
- 13AM4VW2dhxYgXeQepoHkHSQuy6NgaEb94
- %s%d
- Global\\MsWinZonesCacheCounterMutexA
- tasksche.exe
- TaskStart
- t.wnry
- icacls . /grant Everyone:F /T /C /Q
- attrib +h .
- WNcry@2ol7

Encryptor

(MD5: f351e1fcca0c4ea05fc44d15a17f8b36)

- kgptbeilcq
- TaskStart
- c.wnry
- %s
- del /a %%0
- %d%d.bat
- ConvertSidToStringSidW
- advapi32.dll
- SYSTEM
- S-1-5-18
- EVERYONE
- %s\%d%s
- .WNCRYT
- WANACRY!
- .WNCRY
- .WNCYR
- \\
- @WanaDecryptor@.bmp
- @WanaDecryptor@.exe.lnk
- @Please_Read_Me@.txt
- %s\%s
- ..
- %s*
- .dll
- .exe
- ~SD
- @WanaDecryptor@.exe
- Content.IE5
- Temporary Internet Files
- This folder protects against ransomware. Modifying it will reduce protection
- \Local Settings\Temp
- \AppData\Local\Temp
- \Program Files (x86)
- \Program Files
- \WINDOWS
- \ProgramData
- \Intel
- $\
- TESTDATA
- %08X.dky
- Global\MsWinZonesCacheCounterMutexA
- Global\MsWinZonesCacheCounterMutexW
- cmd.exe /c reg add %s /v "%s" /t REG_SZ /d "\"%s\"" /f
- HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
- %s %s
- taskse.exe
- @WanaDecryptor@.exe
- tasksche.exe

- %s\%s\%s
- %s*.*
- @WanaDecryptor@.exe.lnk
- @echo off
- echo SET ow = WScript.CreateObject("WScript.Shell")> m.vbs
- echo SET om = ow.CreateShortcut("%s%s")>> m.vbs
- echo om.TargetPath = "%s%s">> m.vbs
- echo om.Save>> m.vbs
- cscript.exe //nologo m.vbs
- del m.vbs
- u.wnry
- %.1f BTC
- $%d worth of bitcoin
- wb
- r.wnry
- b.wnry
- attrib +h +s %C:\%s
- $RECYCLE
- %C:\%s
- $RECYCLE
- %s\hibsys%s
- taskdl.exe
- f.wnry
- cmd.exe /c start /b %s vs
- %s co
- taskkill.exe /f /im mysqld.exe
- taskkill.exe /f /im sqlwriter.exe
- taskkill.exe /f /im sqlserver.exe
- taskkill.exe /f /im MSExchange*
- taskkill.exe /f /im Microsoft.Exchange.*
- %s fi
- %08X.eky
- %08X.pky
- %08X.res

Decryptor

(MD5: 7bf2b57f2a205768755c07f238fb32cc)

- Connecting to server...
- s.wnry
- %08X.eky
- %08X.res
- 00000000.res
- %08X.dky
- %08X.pky
- Connected
- Sent request
- Succeed
- Received response
- Congratulations! Your payment has been checked!
- Start decrypting now!
- Failed to check your payment!
- Please make sure that your computer is connected to the Internet and
- your Internet Service Provider (ISP) does not block connections to the TOR Network!
- You did not pay or we did not confirmed your payment!
- Pay now if you didn't and check again after 2 hours.
- Best time to check: 9:00am - 11:00am GMT from Monday to Friday.
- You have a new message:
- c.wnry
- runas
- WanaCrypt0r
- Software\
- %04d-%02d-%02d %02d:%02d:%02d

- WANACRY!
- .org
- .WNCYR
- .WNCRY
- @WanaDecryptor@.bmp
- @WanaDecryptor@.exe.lnk
- @Please_Read_Me@.txt
- %s\%s
- ..
- %s*
- Content.IE5
- Temporary Internet Files
- This folder protects against ransomware. Modifying it will reduce protection
- \Local Settings\Temp
- ppData\Local\Temp
- \Program Files (x86)
- \Program Files
- \WINDOWS
- \ProgramData
- \Intel
- Please select a host to decrypt.
- All your files have been decrypted!
- Pay now, if you want to decrypt ALL your files!
- f.wnry
- My Computer
- *.res
- open
- mailto:
- Wana Decrypt0r 2.0
- %s %s
- cmd.exe
- /c vssadmin delete shadows /all /quiet & wmic shadowcopy delete & bcdedit /set {default} bootstatuspolicy ignoreallfailures & bcdedit /set {default} recoveryenabled no & wbadmin delete catalog -quiet
- 13AM4VW2dhxYgXeQepoHkHSQuy6NgaEb94
- English
- m_%s.wnry
- msg\
- <https://
- <http://
- %d/%d/%d %02d:%02d:%02d
- 00;00;00;00
- http://www.btcfrog.com/qr/bitcoinPNG.php?address=%s
- mailto:%s
- https://www.google.com/search?q=how+to+buy+bitcoin
- https://en.wikipedia.org/wiki/Bitcoin
- Send %.1f BTC to this address:
- %.1f BTC
- Send $%d worth of bitcoin to this address:
- %02d;%02d;%02d;%02d
- b.wnry
- --- %s %s %d %I64d %d
- Failed to send your message!
- Please make sure that your computer is connected to the Internet and
- your Internet Service Provider (ISP) does not block connections to the TOR Network!
- Your message has been sent successfully!
- You are sending too many mails! Please try again %d minutes later.
- Too short message!
- %d%%
- %s\%s
- tor.exe
- %s\%s\%s
- TaskData
- taskhsvc.exe
- 127.0.0.1

Appendix A

Observed Killswitch Domains

The following table contains observed killswitch domains and their associated sample hash.

Domain	Associated Sample MD5 Hash
iuqssfsodp9ifjaposdfjhgosurijfaewrwergwea.com	c2559b51cfd37bdbd5fdb978061c6c16
ayylmaotjhsstasdfasdfasdfasdfasdfasdf.com (This domain matches the format of WannaCry-associated domains, but has not yet been clearly linked to a specific sample. Organizations wish to maintain awareness of this domain in the event that it is associated with WannaCry activity.)	a44964a7be94072cdfe085bc43e7dc95
ifferfsodp9ifjaposdfjhgosurijfaewrwergwea.com	80ce983d22c6213f35867053bec1c293
iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com	db349b97c37d22f5ea1d1841e3c89eb4
iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.test	96dff36b5275c67e35097d77a120d0d4

Appendix B

Yara Rules

FireEye has developed the following Yara rules for WannaCry detection:

```
 rule FE_RANSOMWARE_WANNACRY {
            meta:version=".4"
            filetype="PE"
            author="Ian.Ahl@fireeye.com @TekDefense"
            date="2017-05-12"
            description="Generic detection for most WannaCry variants"
strings:
            // Bitcoin URLs
            $bcURL1 = "http://www.btcfrog.com/qr/bitcoinPNG.php?address=%" ascii wide nocase
            $bcURL2 = "https://www.google.com/search?q=how+to+buy+bitcoin" ascii wide nocase

            // Ransom Message
            $msg1 = "Congratulations! Succeed to check your payment!" ascii wide
            $msg2 = "Start decrypting now!" ascii wide
            $msg3 = "All your files have been decrypted!" ascii wide
            $msg4 = "Pay now, if you want to decrypt ALL your files!" ascii wide
            $msg5 = "Send $%d worth of bitcoin to this address:" ascii wide
            $msg6 = "Ooops, your files have been encrypted!" ascii wide

            // WANNA Strings
            $wanna1 = "Wanna Decryptor 1.0" ascii wide
            $wanna2 = "Wana Decrypt0r" ascii wide
            $wanna3 = "Wana Decryptor" ascii wide
            $wanna4 = "WANNACRY" ascii wide nocase
            $wanna5 = "WanaCrypt0r" ascii wide nocase
            $wanna6 = "WANACRY!" ascii wide
            $wanna7 = "WNcry@2ol7" ascii wide
            $wanna8 = "wcry@123"
            $wanna9 = "wcry@2016"

            // File references
            $fileA1 = "!WannaCryptor!.bmp" ascii wide
            $fileA2 = "!WannaDecryptor!.exe.lnk" ascii wide
            $fileA3 = "!Please Read Me!.txt" ascii wide
```

```
            $fileB1 = "@WanaDecryptor@.bmp" ascii wide
            $fileB2 = "@WanaDecryptor@.exe.lnk" ascii wide
            $fileB3 = "@Please_Read_Me@.txt" ascii wide

            // CMDS
            $cmd1 = "cmd.exe /c start /b vssadmin.exe Delete Shadows /All /Quiet" ascii wide nocase
            $cmd2 = "wmic shadowcopy delete" ascii wide
            $cmd3 = "bcdedit /set {default} bootstatuspolicy ignoreallfailures" ascii wide
            $cmd4 = "bcdedit /set {default} recoveryenabled no" ascii wide
            $cmd5 = "wbadmin delete catalog -quiet" ascii wide
            $cmd6 = "icacls . /grant Everyone:F /T /C /Q" ascii wide

            // MISC
            $misc1 = "StartTask" wide ascii
            $misc2 = "b.wry" wide ascii
            $misc3 = "c.wry" wide ascii
            $misc4 = "m.wry" wide ascii
            $misc5 = "inflate 1.1.3 Copyright 1995-1998 Mark Adler" wide ascii
            $misc6 = "?AVtype_info@@" wide ascii

condition:
            (
                        (
                                    (uint16(0) == 0x5A4D)
                        )
                        and
                        (
                                    all of ($fileA*)
                                    or
                                    all of ($fileB*)
                                    or
                                    (4 of ($msg*) and 2 of ($bcURL*))
                                    or
                                    2 of ($wanna*)
                                    or
                                    (2 of ($msg*) and 1 of ($cmd*))
                                    or
                                    4 of ($cmd*)
                                    or
                                    (1 of ($wanna*) and 1 of ($cmd*))
                                    or
                                    (1 of ($wanna*) and 3 of ($misc*))
                        )
            )
}
```

```
 rule FE_RANSOMWARE_WANNACRY_EB {
           meta:version=".1"
           filetype="PE"
           author="Ian.Ahl@fireeye.com @TekDefense"
           date="2017-05-12"
           description="Focusing on the WannaCry variants with worm capabilities"
strings:

            // EB related strings in WANNACRY
            $eb1 = "__USERID__PLACEHOLDER__@" ascii wide
            $eb2 = "__TREEID__PLACEHOLDER__" ascii wide
            $eb3 = "LANMAN1.0" ascii wide
            $eb4 = "LANMAN2.1" ascii wide
            $eb5 = "\\PIPE\\" ascii wide
            $eb6 = "\\\\%s\\IPC$" ascii wide
            $eb7 = "__TREEPATH_REPLACE__" ascii wide
            $eb8 = "/K__USERID__PLACEHOLDER__" ascii wide

condition:
            (
                        (
                                    (uint16(0) == 0x5A4D)
                        )
                        and
```

```
                        (
                                all of ($eb*)
                        )
            )
}
```

Posted in

- Threat Intelligence
