



Practical Malware Analysis & Triage

Malware Analysis Report

Wannacry Ransomware

August 2023 | KillSwitch | v1.0

Table of Contents

Table of Contents	2
Executive Summary	3
High-Level Technical Summary	4
Malware Composition	5
wannacry.exe	5
tasksche.exe	5
Other Files	6
Basic Static Analysis	14
Analysis using floss	14
Analysis using PEStudio	15
Basic Dynamic Analysis.....	17
1. Initial Detonation (While connected to Inetsim)	17
2. Intial Detonation (Without being conencted to Inetsim/Internet)	18
(i) Initial Analysis	18
(ii) Finding Host Based Indicators using Process Monitor.....	29
(iii) Finding Host Based Inidcators using Process Hacker 2	20
(iv) Finding Network Based Indicators using TCPview	21
(v) Overview of all the processes using Procmon's process tree	22
Analyzing the Decryption Function	25
Advanced Static Analysis	27
Advanced Dynamic Analysis.....	28
Overview of the Malware	30
Rules & Signatures	31
Appendices.....	31
A. Yara Rules.....	32
B. Diassembled Code Snippets.....	33

Executive Summary

SHA256 hash	6C382A1C16DBA41B4FF6F0D728E9E92AEBFE2EE0C7FEB30A0E63B15EAF6C4B44
-------------	--

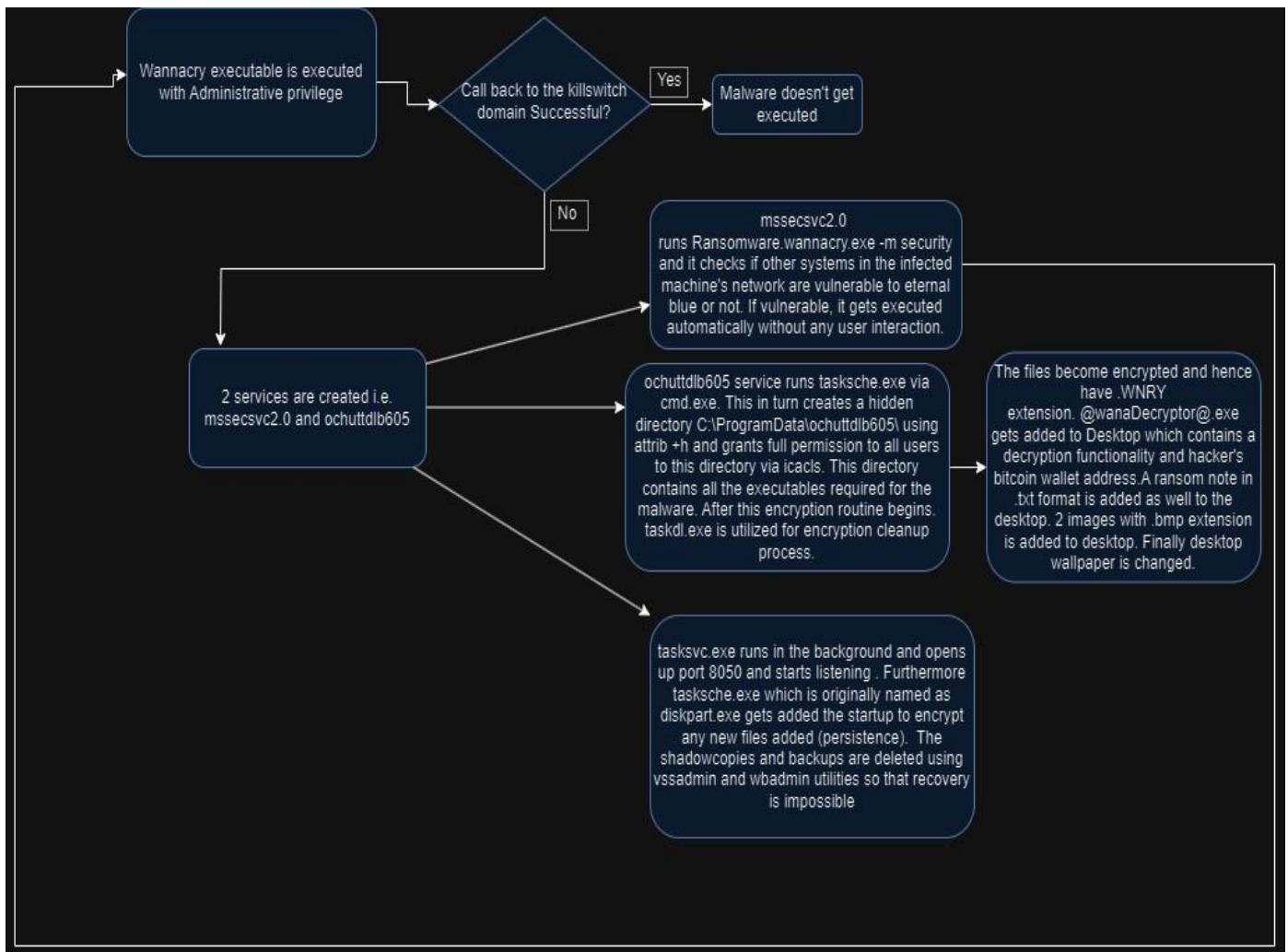
WannaCry, also known as WannaCryptor is a notorious malware that exhibits dual functionality, acting as both ransomware and a worm. It gained worldwide attention in 2017 due to its rapid propagation and impact on organizations and individuals. It is C++ compiled and runs on both x86 and x64 Windows operating systems. Firstly, two services are created i.e., mssecsvc2.0 and ochuttdlb605 (random characters and it varies for everyone). mssecsvc2.0 takes care of the worm activities of the malware. ochuttdlb605 service runs tasksche.exe which in turn creates a hidden directory C:\Program Data\ochuttdlb605 where all the executables required for malware are unpacked and encryption routine is begun and all the files are encrypted. A ransom is asked for decryption. Once infecting the victim, it spreads like a worm to the other computers in the network using eternal blue vulnerability and this spreading requires no user interaction and happens automatically. It affects the other computers in the network only if they are vulnerable to eternal blue i.e., if they use SMBv1 protocol.

Symptoms of infection include:

- The extension of the files change “.WNRY” .
- An executable called @WanaDecryptor@.exe gets added to the desktop.
- Ransom notes and 2 images (with .bmp extension) are also added to the desktop
- Desktop background image is changed indicating that our files have been encrypted.
- Two services called mssecsvc2.0 and ochuttdlb605 are created
- A hidden directory called ochuttdlb605 (random characters an varies) C:\ProgramData\
- tasksche.exe is added to the startup.
- Shadow copies and backups are deleted.
- An executable called taskhsvc.exe listens in port 9050 for remote connections.

High-Level Technical Summary

The URL is a killswitch domain for the malware. On execution, it creates 2 services mssecsvc2.0 and ochuttdlb605. The first one is responsible for worm activities while the second one executes tasksche.exe. tasksche.exe creates a new hidden within directory C:\Programdata called ochuttdlb605 (same name as the service created earlier). All other important executables and encryption keys are unpacked in C:\ProgramData\ ochuttdlb605. The files are encrypted and a ransom in bitcoins worth \$300 has to be made. tasksche.exe is added to startup for persistence, and the newer files added too are encrypted. mssecsvc2.0 checks if other systems in the network of the infected machine are infected automatically without any permission/interaction required if they are vulnerable to eternal blue.



Malware Composition

Wannacry ransomware consists of the following components:

File Name	SHA256 Hash
Ransomware.wannacry.exe	6C382A1C16DBA41B4FF6F0D728E9E92AEBFE2EE0C7FEB30A0E63B15EAF6C4B44
tasksche.exe	ED01EBFBC9EB5BBEA545AF4D01BF5F1071661840480439C6E5BABE8E080E41AA

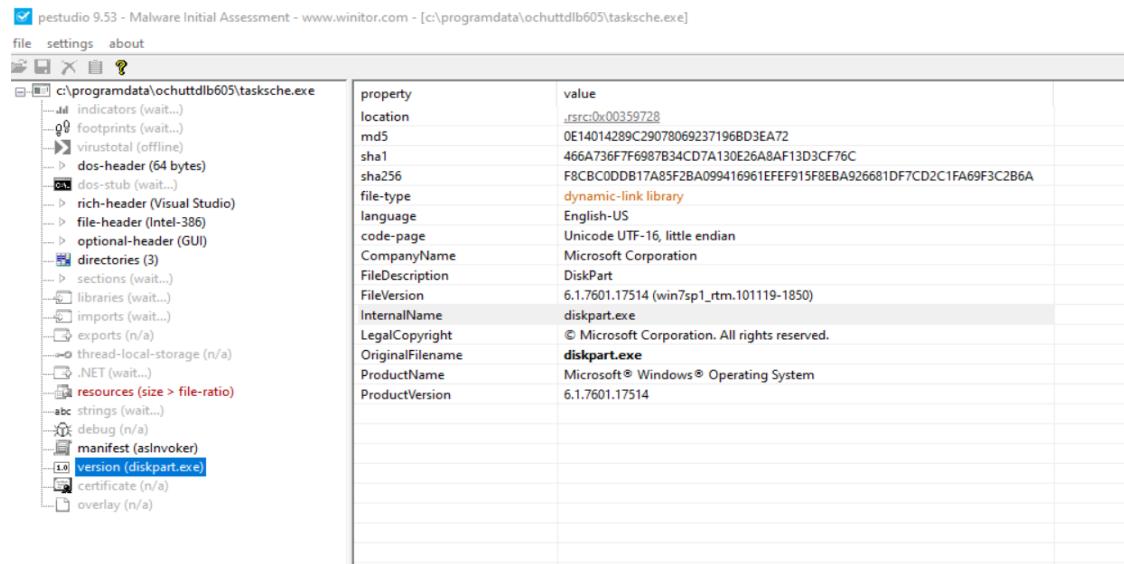
Ransomware.wannacry.exe

It is the initial executable and needs to be run with administrative privilege for it to be executed. In addition to this, the victim system also shouldn't be connected to the internet/inetsim since the killswitch domain it calls back to is registered now. If the callback to the killswitch domain is successful, the malware won't get executed.

tasksche.exe

This executable is executed by ochuttdlb605 via cmd.exe.tasksche.exe creates the hidden directory C:\ProgramData\ochuttdlb605. All executables required for the malware is unpacked here. The encryption routine begins as well.

Its originally called dispart.exe: (diskpart.exe was present in the startup folder when I detonated the malware but after rebooting, I wasn't able to find it in startup)



property	value
location	.rsrc:0x00359728
md5	0E14014289C29078069237196BD3EA72
sha1	466A736F76987B34CD7A130E26A8AF13D3CF76C
sha256	F8CBCODDB17A85F2BA099416961EFEF915F8EBA926681DF7CD2C1FA69F3C2B6A
file-type	dynamic-link library
language	English-US
code-page	Unicode UTF-16, little endian
CompanyName	Microsoft Corporation
FileDescription	DiskPart
FileVersion	6.1.7601.17514 (win7sp1_rtm.101119-1850)
InternalName	diskpart.exe
LegalCopyright	© Microsoft Corporation. All rights reserved.
OriginalFilename	diskpart.exe
ProductName	Microsoft® Windows® Operating System
ProductVersion	6.1.7601.17514

Figure 1: tasksche.exe is originally called diskpart.exe (PEstudio)



Other Files

- The following are the executables present in the C:\ProgramData\ochuttdlb605 folder:

	Name	Date modified	Type	Size
Quick access				
Desktop	msg	25-08-2023 04:49	File folder	
Downloads	TaskData	25-08-2023 04:50	File folder	
Documents	@Please_Read_Me@.txt	25-08-2023 04:49	Text Document	1 KB
Pictures	@WanaDecryptor@.exe	12-05-2017 02:22	Application	240 KB
Music	@WanaDecryptor@.exe	25-08-2023 04:49	Shortcut	1 KB
Videos	00000000.eky	25-08-2023 04:49	EKY File	0 KB
OneDrive	00000000.pky	25-08-2023 04:49	PKY File	1 KB
This PC	00000000.res	25-08-2023 04:58	RES File	1 KB
	b.wnry	11-05-2017 20:13	WNRY File	1,407 KB
	c.wnry	25-08-2023 04:50	WNRY File	1 KB
	f.wnry	25-08-2023 04:49	WNRY File	1 KB
	r.wnry	11-05-2017 15:59	WNRY File	1 KB
	s.wnry	09-05-2017 16:58	WNRY File	2,968 KB
	t.wnry	12-05-2017 02:22	WNRY File	65 KB
	taskdl.exe	12-05-2017 02:22	Application	20 KB
	tasksche.exe	25-08-2023 04:49	Application	3,432 KB
	taskse.exe	12-05-2017 02:22	Application	20 KB
	u.wnry	12-05-2017 02:22	WNRY File	240 KB

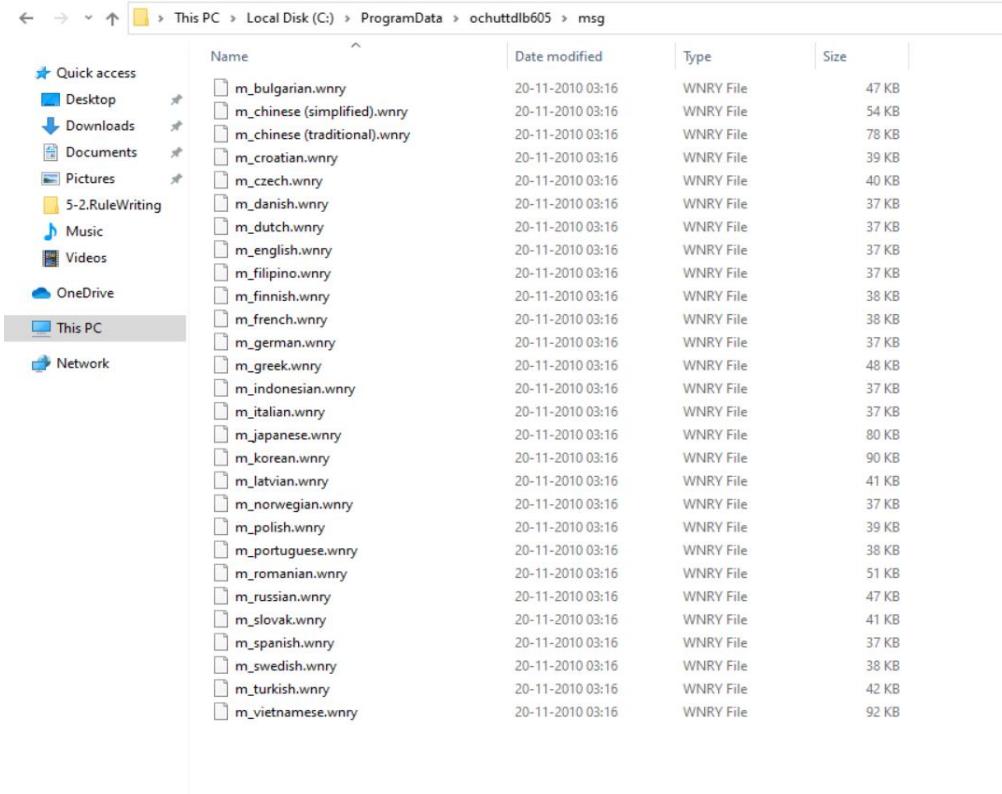
Figure 2: C:\ProgramData\ochuttdlb605

(i) Keys and Other files:

- 00000000.pky, 00000000.eky → Both are encrypted RSA keys (I know this from PEStudio results which show RSA1 as the first byte)
- 00000000.res → Not sure what it does
- t.wnry and s.wnry → Not sure what they do

(ii) msg folder:

- msg folder contains ransom message in various languages:



Name	Date modified	Type	Size
m_bulgarian.wnry	20-11-2010 03:16	WNRY File	47 KB
m_chinese (simplified).wnry	20-11-2010 03:16	WNRY File	54 KB
m_chinese (traditional).wnry	20-11-2010 03:16	WNRY File	78 KB
m_croatian.wnry	20-11-2010 03:16	WNRY File	39 KB
m_czech.wnry	20-11-2010 03:16	WNRY File	40 KB
m_danish.wnry	20-11-2010 03:16	WNRY File	37 KB
m_dutch.wnry	20-11-2010 03:16	WNRY File	37 KB
m_english.wnry	20-11-2010 03:16	WNRY File	37 KB
m_filipino.wnry	20-11-2010 03:16	WNRY File	37 KB
m_finnish.wnry	20-11-2010 03:16	WNRY File	38 KB
m_french.wnry	20-11-2010 03:16	WNRY File	38 KB
m_german.wnry	20-11-2010 03:16	WNRY File	37 KB
m_greek.wnry	20-11-2010 03:16	WNRY File	48 KB
m_indonesian.wnry	20-11-2010 03:16	WNRY File	37 KB
m_italian.wnry	20-11-2010 03:16	WNRY File	37 KB
m_japanese.wnry	20-11-2010 03:16	WNRY File	80 KB
m_korean.wnry	20-11-2010 03:16	WNRY File	90 KB
m_latvian.wnry	20-11-2010 03:16	WNRY File	41 KB
m_norwegian.wnry	20-11-2010 03:16	WNRY File	37 KB
m_polish.wnry	20-11-2010 03:16	WNRY File	39 KB
m_portuguese.wnry	20-11-2010 03:16	WNRY File	38 KB
m_romanian.wnry	20-11-2010 03:16	WNRY File	51 KB
m_russian.wnry	20-11-2010 03:16	WNRY File	47 KB
m_slovak.wnry	20-11-2010 03:16	WNRY File	41 KB
m_spanish.wnry	20-11-2010 03:16	WNRY File	37 KB
m_swedish.wnry	20-11-2010 03:16	WNRY File	38 KB
m_turkish.wnry	20-11-2010 03:16	WNRY File	42 KB
m_vietnamese.wnry	20-11-2010 03:16	WNRY File	92 KB

Figure 3: msg folder



```

m_english.wnry - Notepad
File Edit Format View Help

\par \hich\af31502\dbch\af53\loch\f31502 You can decrypt some of your files for free. Try now by clicking <Decrypt>.
\par \hich\af31502\dbch\af53\loch\f31502 But if you want to decrypt all your files, you need to pay.
\par \hich\af31502\dbch\af53\loch\f31502 You only have 3 days to submit the payment. After that the \hich\af31502\dbch\af53\loch\f31502 price will be doubl
\par \hich\af31502\dbch\af53\loch\f31502 Also, if you don't pay in 7 days, you won't be able to recover your files forever.
\par \hich\af31502\dbch\af53\loch\f31502 We will have free events for users who are so poor that they couldn't pay in 6 months.
\par
\par }{\rtlch\fcs1 \af1\afs22 \ltrch\fcs0 \b\fs28\loch\af31502\hich\af31502\dbch\af53\insrsid14313477\charrsid5733561 \hich\af31502\dbch\af53\loch\f31502 H
\par }{\rtlch\fcs1 \af1\afs22 \ltrch\fcs0 \b\fs24\loch\af31502\hich\af31502\dbch\af53\insrsid14313477\charrsid5733561 \hich\af31502\dbch\af53\loch\f31502 P
\fs22\loch\af31502\hich\af31502\dbch\af53\insrsid14313477\charrsid5733561 \hich\af31502\dbch\af53\loch\f31502 ayment is accepted in Bitcoin only. For more
\par \hich\af31502\dbch\af53\loch\f31502 Please check the current price of Bitcoin and buy some bitcoins. For more information, click <How to buy bitcoins>
\par \hich\af31502\dbch\af53\loch\f31502 And send the correct amount to the address specified in this window.
\par \hich\af31502\dbch\af53\loch\f31502 After your payment, click <Check Pay>\hich\af31502\dbch\af53\loch\f31502 ment>. Best time to check: 9:00am - 11:00a
\par \hich\af31502\dbch\af53\loch\f31502 Once the payment is checked, you can start decrypting your files immediately.
\par
\par }{\rtlch\fcs1 \af1\afs22 \ltrch\fcs0 \b\fs28\loch\af31502\hich\af31502\dbch\af53\insrsid14313477\charrsid5733561 \hich\af31502\dbch\af53\loch\f31502 C
\par }{\rtlch\fcs1 \af1\afs22 \ltrch\fcs0 \fs22\loch\af31502\hich\af31502\dbch\af53\insrsid14313477\charrsid5733561 \hich\af31502\dbch\af53\loch\f31502 If
\par
\par }{\rtlch\fcs1 \af1\afs22 \ltrch\fcs0 \fs22\cf6\loch\af31502\hich\af31502\dbch\af53\insrsid14313477\charrsid5733561 \hich\af31502\dbch\af53\loch\f31502 not remove this software, and disable your anti-virus for a while, until you pay and the payment gets processed. If your anti-virus gets updated and remov
\rtlch\fcs1 \af1\afs22 \ltrch\fcs0 \fs22\cf6\loch\af31502\hich\af31502\dbch\af53\insrsid5268979\charrsid5733561

```

Figure 4: I confirmed that its ransom note message by reading the one in English

(iii) taskdl.exe:

- taskdl.exe is used for cleanup process.



- API like DeleteFileW is used for the cleanup process:



Figure 5: Analysis of taskdl.exe using PEStudio

(iv) taskse.exe:

- It's not clear on what taskse.exe does.

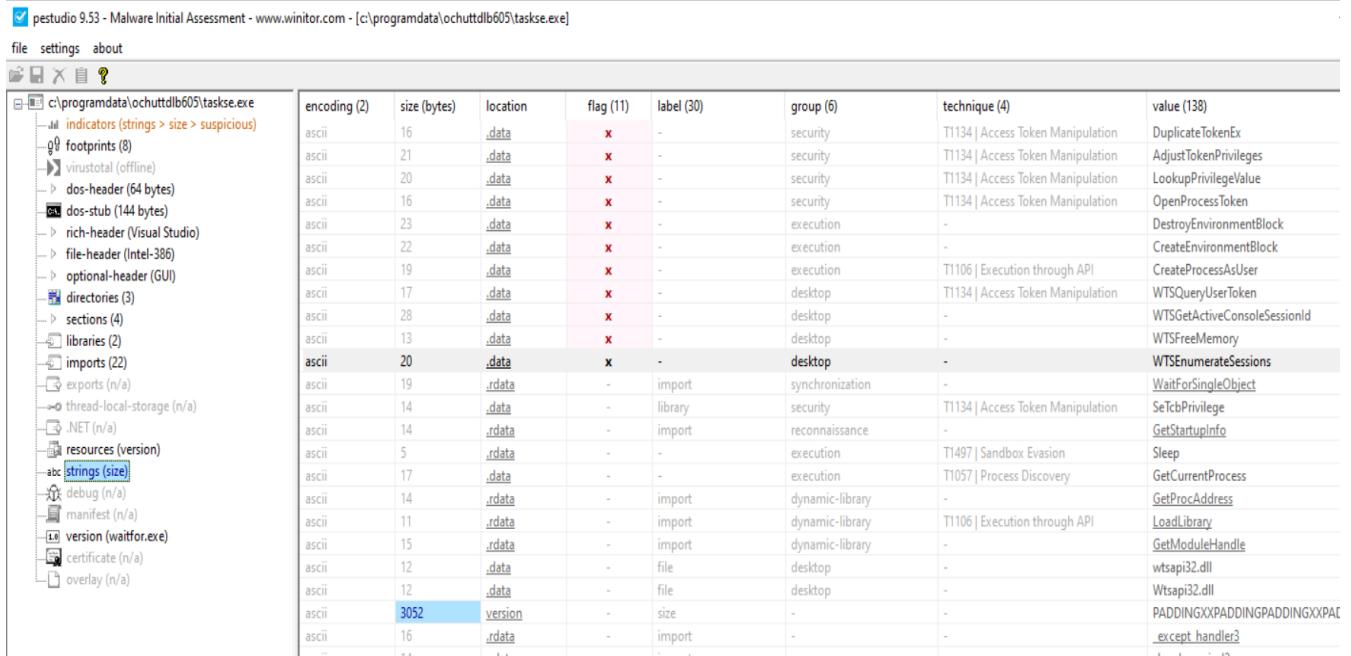


Figure 6: Analysis of taskse.exe using PEStudio

(v) c.wnry:

- c.wnry is not an executable but contains some crucial strings (C2 communication server URL, tor browser download link and the bictoin wallet address.)

pestudio 9.53 - Malware Initial Assessment - www.winitor.com - [c:\programdata\ochuttdlb605\c.wnry]

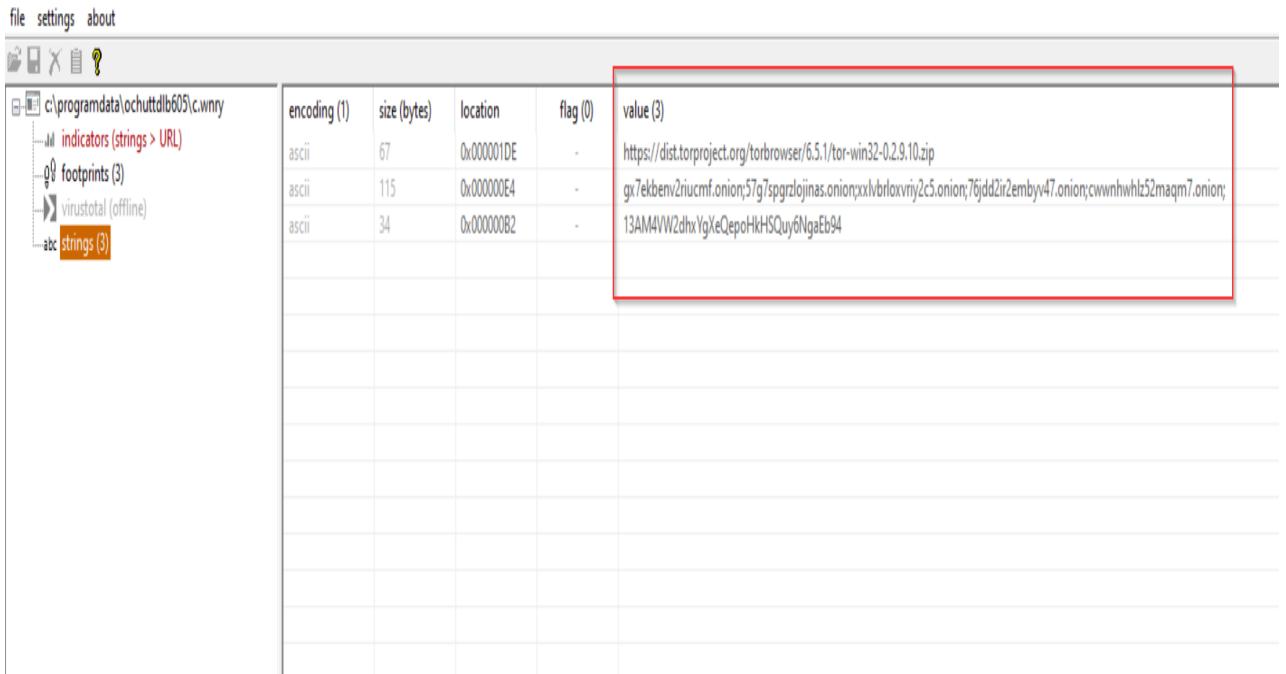


Figure 7: Analysis of c.wnry using PEStudio

Tor browser download link:

- hxxps [://]dist[.]torproject[.]org/torbrowser/6[.]5[.]1/tor-win32-0[.]2[.]9[.]10[.]zip is the URL for downloading the tor browser (it doesn't work now)

List of C2 server sites:

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

Bitcoin wallet address:

- 13AM4VW2dhxYgXeQepoHkHSQuy6NgaEb94 is the Bitcoin wallet address



(vi) taskhsvc.exe /tor.exe:

- We can view both taskhsvc.exe and tor.exe in same location i.e.,
C:\ProgramData\ochuttdlb605\TaskData\Tor: (actually tor.exe is copied into taskhsvc.exe)

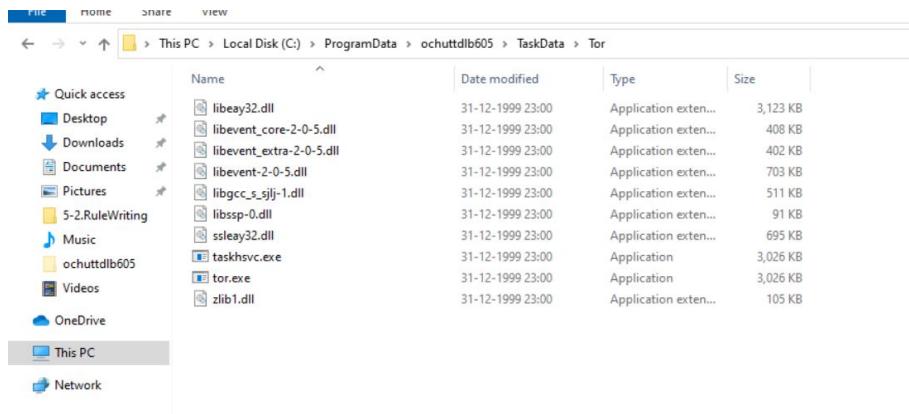


Figure 8: Tor folder

- The main takeaway for us here is that taskhsvc.exe is actually tor.exe which is renamed and this opens up the port 9050 in the machine locally (127.0.0.1) and starts listening and might connect to one of the C2 servers (onion sites) present in c.wncry using the tor browser which is installed from hxxps[://]dist[.]torproject[.]org/torbrowser/6[.]5[.]1/tor-win32-0[.]2[.]9[.]10[.]zip

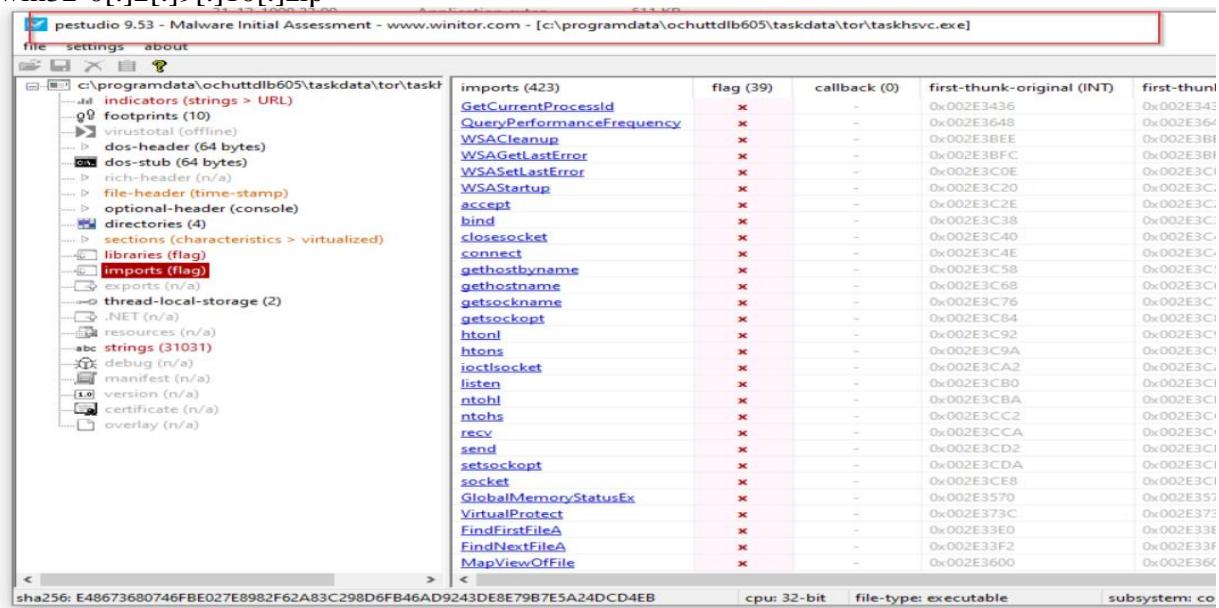


Figure 9: Both taskhsvc.exe and tor.exe have same API's and file hash confirming our theory that both are same files (PEStudio)



(vii) b.wnry:

- The first byte BM6 indicates that it's an image (with .bmp extension):

property		value
sha256		D5EDE8694DC0548D8E6B87C83D50F4AB85C1DEBADB106D6A6A794C3E746F4FA
sha1		F19ECEDA82973239A1FD5C5826BC7691E5DCB4FB
md5		C17170262312F3BE7027BC2CA825BF0C
first-bytes-hex		42 4D 36 F9 15 00 00 00 00 36 00 00 28 00 00 20 03 00 00 58 02 00 00 01 00 18 00 00 00 00
first-bytes-text		BM6.....6.....X.....
file-size		144004 bytes
entropy		0.336

Figure 10: b.wnry file signature (PEstudio)

- There is bmp image in Desktop and b.wnry is @WanaDecryptor@.bmp:

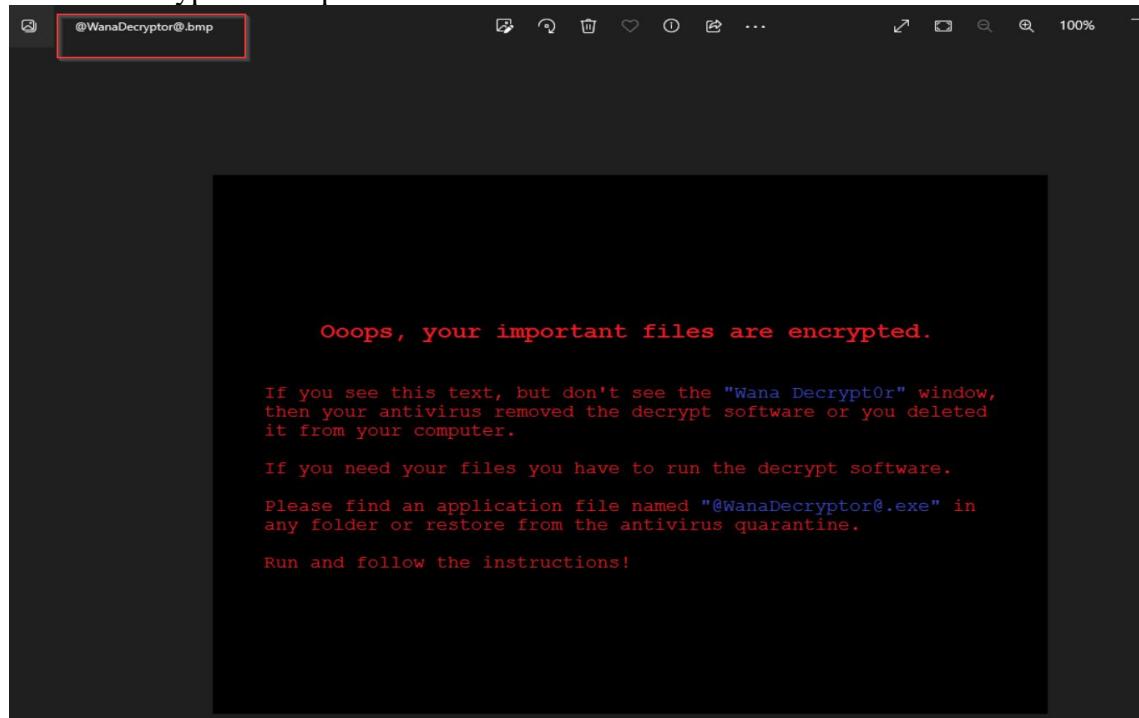


Figure 11: b.wnry is @WanaDecryptor@.bmp

(viii) r.wnry:

- r.wnry contains the ransom note:

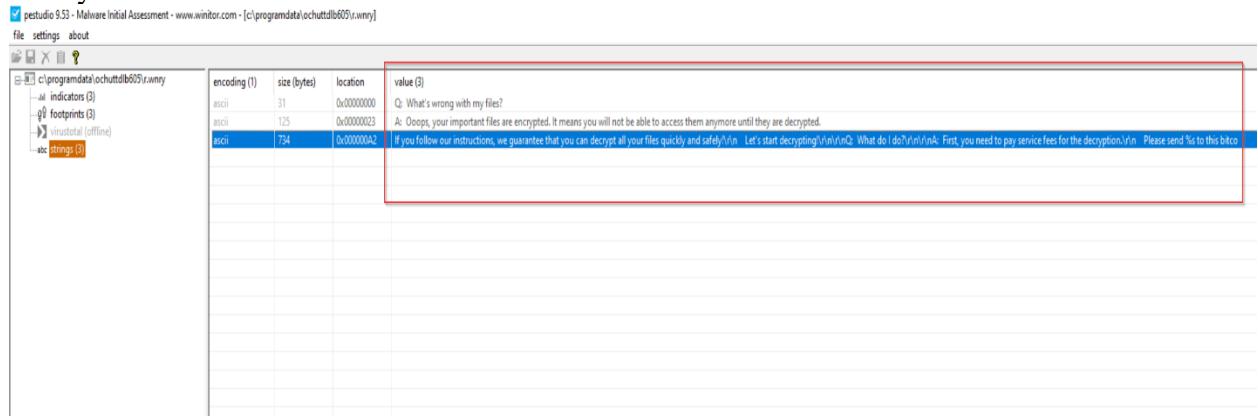


Figure 12: r.wnry analyzed using PEStudio

- The ransom note we find on the Desktop is actually r.wnry: (@Please_Read_Me@.txt)

```

@Please_Read_Me@.txt - Notepad
File Edit Format View Help
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 $300 worth of bitcoin to this bitcoin address: 13AM4VW2dPxYgXeQepoHkHSQuy6NgaEb94

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 13: r.wnry is @Please_Read_Me@.txt (ransom note in desktop)

(ix) u.wnry/@wanaDecryptor@.exe:

- u.wnry is an executable and it is probably @wanaDecryptor@.exe as both have the same strings and API's:
- We could confirm our theory both are same since both @wanaDecryptor@.exe and u.wnry have the same original name i.e. LODCTR.EXE:

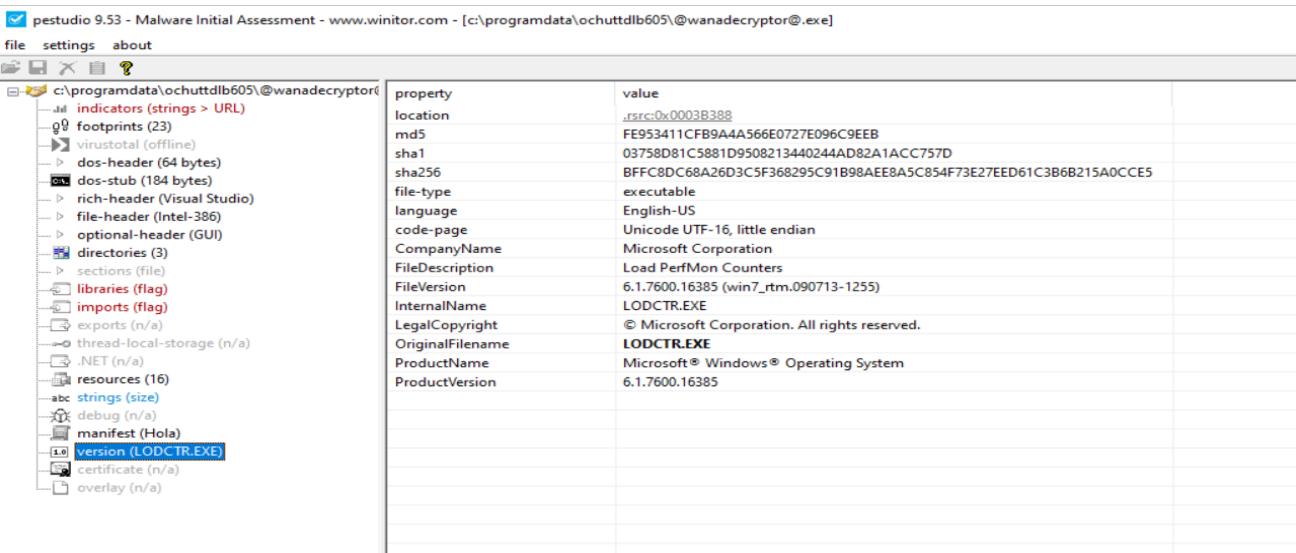


Figure 14: u.wnry is @wanaDecryptor@.exe (PEstudio)



Basic Static Analysis

1. Analysis using floss

- Important strings highlighted:

```
TREEPATH_REPLACE_
\\%s\IPC$ Microsoft Base Cryptographic Provider v1.0
%d.%d.%d.%d
msscsvc2.0 Microsoft Security Center (2.0) Service
% - security
[ : \ % s \ q e r i u w j h n f
[ : \ % s \ %
WINDOWS tasksche.exe
CloseHandle
WriteFile
CreateFileA
CreateProcessA
http://www.iuqerfsodo9ifiaoposdfjhgosuriifaewrwergwea.com
!This program cannot be run in DOS mode.
^.rdata
MWNWNPj
V, YYG;~
tIht Ht
k|_ ^][Y
=1&LZ661A??~
f" "D~**T
V22dN::t
o%%Jr.. \$

U|x8+^_
Zm#om
2/0-_X8w.+
|~}{.15
Microsoft Enhanced RSA and AES Cryptographic Provider
CryptGenKey
CryptDecrypt
CryptEncrypt
CryptDestroyKey
CryptImportKey
CryptAcquireContextA
cmd.exe /c "%s"
115p7UMMngoj1pMvkpHijcRdfJNXj6LrLn
12t9YDPgwueZ9NyMgw519p7AA8isjr6SMw
13AM4VW2dhwYgXeQeoHkHSQuy6NgaEb94
Global\MsWinZonesCacheCounterMutexA
tasksche.exe
TaskStart
t.wnry
icacls . /grant Everyone:F /T /C /O
attrib +h .
WNCry@2017
GetNativeSystemInfo

interesting file paths
we could see a SMB share path being referenced here
Appears to be intentionally misspelling TaskSched.exe
Could be a C2 server domain
Interesting cryptographic API'S
executes some file
interesting executable
Full control is given to everyone for the current directory
current directory is also hidden
```

Figure 15: floss output



```
U|x8+^_
Zm#om
2/0-_.X8w.+
|~j%.15
Microsoft Enhanced RSA and AES Cryptographic Provider
CryptGenKey
CryptDecrypt
CryptEncrypt
CryptDestroyKey
CryptImportKey
CryptAcquireContextA
cmd.exe /c "%s"
115p7UMMngoj1pMvkpHijcRdfJNjxJ6LrLn
12t9YDPgwueZ9NyMgw519p7AA8isjr6SMw
13AM4VM2dhwYgXeQepoHkhHSQuy6NgaEb94
Global\MsWinZonesCacheCounterMutexA
tasksche.exe
TaskStart
t.wnry
icacls . /grant Everyone:F /T /C /Q
attrib +h .
Wincry@2017
GetNativeSystemInfo

Annotations:


- interesting cryptographic API'S
- executes some file
- interesting executable
- Full control is given to everyone for the current directory
- current directory is also hidden

```

Figure 16: floss output

2. Analysis using PEStudio

- Wannacry is an executable with 32-bit architecture and is written in C++.

Figure 17: PE studio output



- 1 => These service APIs indicate that some service might be created for persistence
- 2 => These APIs indicates some internet related activities
- 3 => These are cryptography related APIs

The screenshot shows the PE studio interface with the file 'ransomware.wannacry.exe.malz' open. The left sidebar displays various analysis sections like indicators, footprints, and resources. The main window is a table of imports:

Imports (91)	flag (28)	callback (0)	first-thunk-original (INT)	first-thunk (IAT)	hint	group (10)
StartServiceCtrlDispatcherA	x	-	0x0000A6F6	0x0000A6F6	586 (0x024A)	services
ChangeServiceConfig2A	x	-	0x0000A6C0	0x0000A6C0	52 (0x0034)	services
CreateServiceA	x	-	0x0000A688	0x0000A688	100 (0x0064)	services
QueryPerformanceFrequency	x	-	0x0000A43A	0x0000A43A	676 (0x02A4)	reconnaissance
3 (closesocket)	x	-	0x80000003	0x80000003	0 (0x0000)	network
16 (recv)	x	-	0x80000010	0x80000010	0 (0x0000)	network
19 (send)	x	-	0x80000013	0x80000013	0 (0x0000)	network
8 (htonl)	x	-	0x80000008	0x80000008	0 (0x0000)	network
14 (ntohl)	x	-	0x8000000E	0x8000000E	0 (0x0000)	network
115 (WSAStartup)	x	-	0x80000073	0x80000073	0 (0x0000)	network
12 (inet_ntoa)	x	-	0x8000000C	0x8000000C	0 (0x0000)	network
10 (ioctlsocket)	x	-	0x8000000A	0x8000000A	0 (0x0000)	network
18 (select)	x	-	0x80000012	0x80000012	0 (0x0000)	network
9 (htons)	x	-	0x80000009	0x80000009	0 (0x0000)	network
23 (socket)	x	-	0x80000017	0x80000017	0 (0x0000)	network
4 (connect)	x	-	0x80000004	0x80000004	0 (0x0000)	network
11 (inet_addr)	x	-	0x8000000B	0x8000000B	0 (0x0000)	network
GetAdaptersInfo	x	-	0x0000A792	0x0000A792	28 (0x001C)	network
InternetOpenA	x	-	0x0000A7DC	0x0000A7DC	146 (0x0092)	network
InternetOpenUrlA	x	-	0x0000A7C8	0x0000A7C8	147 (0x0093)	network
InternetCloseHandle	x	-	0x0000A7B2	0x0000A7B2	105 (0x0069)	network
MoveFileExA	x	-	0x0000A576	0x0000A576	623 (0x026F)	file
GetCurrentThreadId	x	-	0x0000A524	0x0000A524	326 (0x0146)	execution
GetCurrentThread	x	-	0x0000A53A	0x0000A53A	325 (0x0145)	execution
CryptGenRandom	x	-	0x0000A650	0x0000A650	150 (0x0096)	cryptography
CryptAcquireContextA	x	-	0x0000A638	0x0000A638	133 (0x0085)	cryptography
rand	x	-	0x0000A824	0x0000A824	678 (0x02A6)	cryptography
strand	x	-	0x0000A852	0x0000A852	692 (0x02B4)	cryptography

Figure 18: Analyzing APIs used by malware using PE studio

Basic Dynamic Analysis

1. Initial Detonation (while connected to Inetsim)

The malware didn't execute although I ran it with administrative privilege. I ran inetsim in Remnux VM which is present in the same network as the victim Windows machine I used for testing while detonating the malware. I utilized wireshark to capture the packets sent by the malware detonating machine (Windows) to Remnux VM. A HTTP GET request is sent to the hxxp[://]www[.]iuquerfsodp9ifjaposdfjhgosurijfaewrwerwgwea[.]com which is killswitch domain and a response is generated by inetsim for this. So , if the callback the killswitch domain is successful, the malware won't get executed.

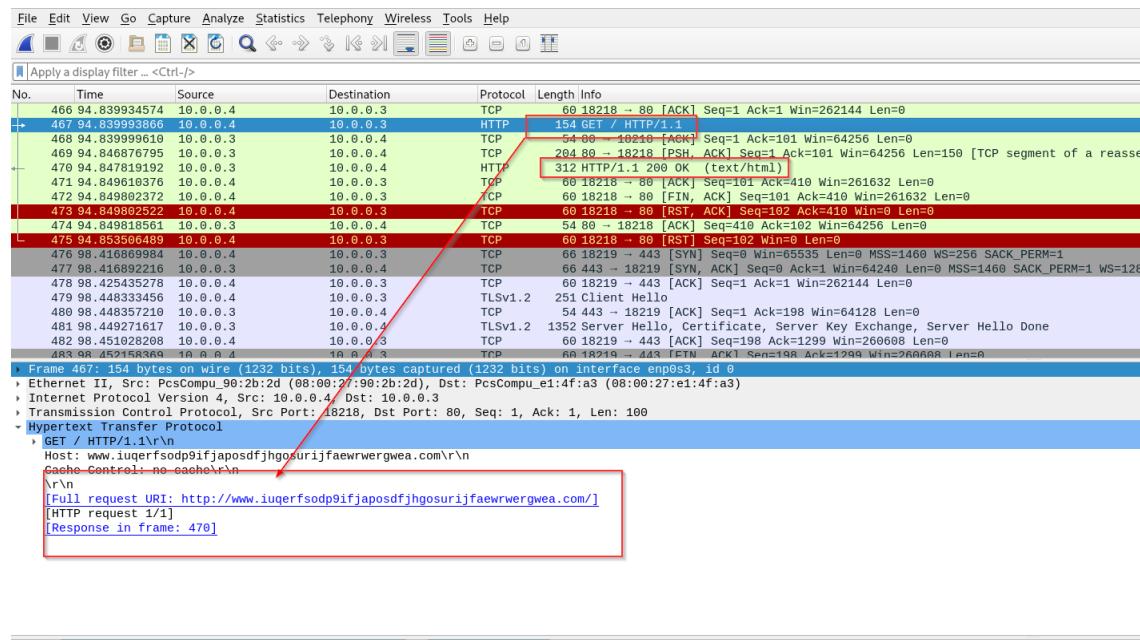


Figure 19: Wireshark output (from Remnux VM)

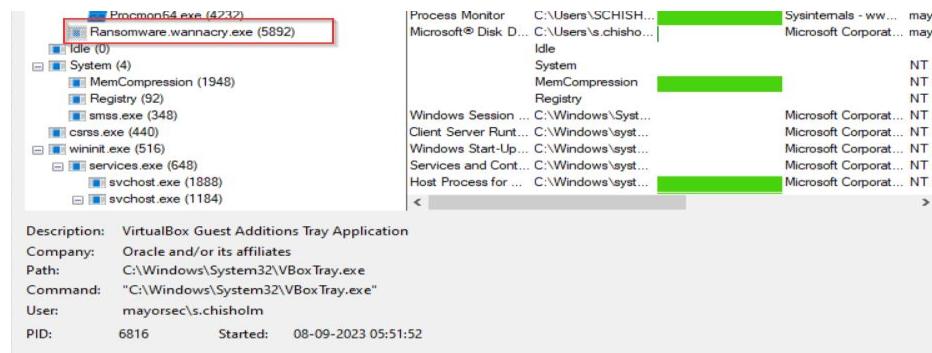


Figure 20: Procmon's process tree (shows that the malware didn't execute further)

2. Initial Detonation (without being connected to Inetsim/Internet)

(i) Intital Analysis

- The malware is run with administrative privileges.
- After detonating, firstly the files get encrypted and their extension gets changed to WNCRY. After that, an executable called @WanaDecryptor@.exe gets added to the desktop. Furthermore, a ransom note and also 2 images are added to the desktop. Finally, the desktop background image has been changed to the image that was previously added to the desktop, indicating that our files have been encrypted.
- Only certain types of files are affected by this encryption.

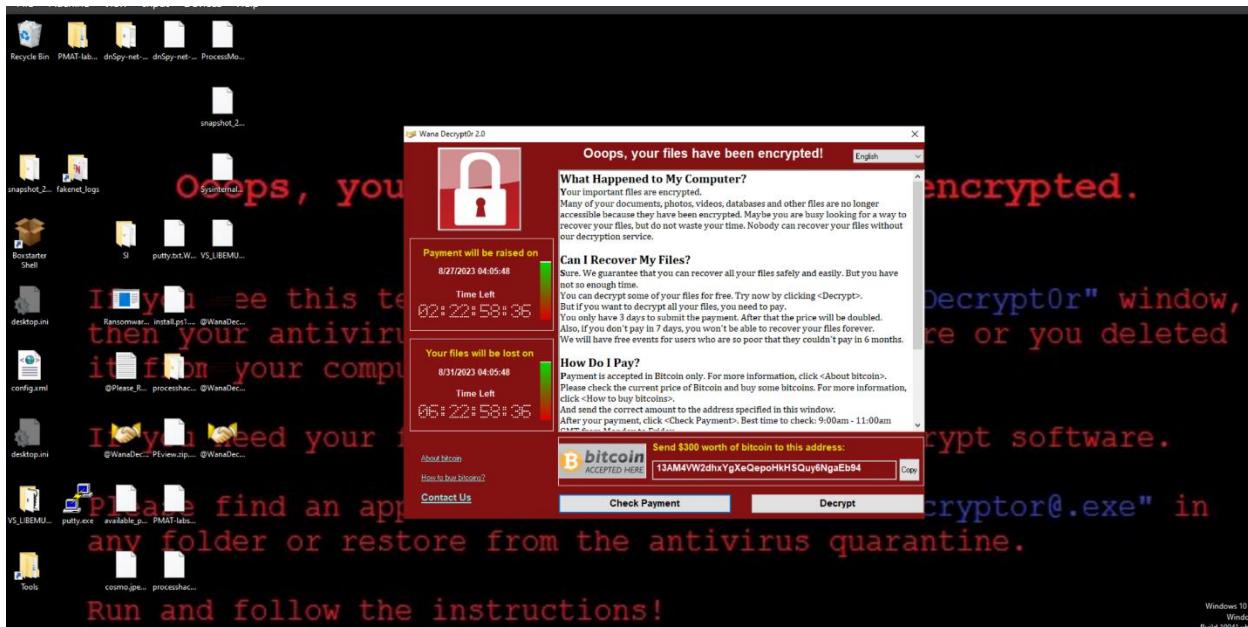


Figure 21: screenshot of the Desktop post detonation



(ii) Finding Host Based Indicators using Process Monitor

- Using process monitor we found that tasksche.exe executable was created and we confirmed it by navigation to C:\Windows

Figure 22: Creation of a file(tasksche.exe) (Procmon output)

- Result of filtering processes with parent PID 4460 (tasksche.exe's PID):

Time ...	Process Name	PID	Operation	Path	Result
04:49...	task sche.exe	1628	Create File	C:\Windows\SysWOW64\KernelBase.dll	SUCCESS
04:49...	task sche.exe	1628	Create File	C:\Windows\apppatch\aymain.sdb	SUCCESS
04:49...	task sche.exe	1628	Create File Mapp.	C:\Windows\apppatch\aymain.sdb	FILE LOCKED WITH ONLY READERS
04:49...	task sche.exe	1628	Create File Mapp.	C:\Windows\apppatch\aymain.sdb	SUCCESS
04:49...	task sche.exe	1628	Create File	C:\Windows\tasksche.exe	SUCCESS
04:49...	task sche.exe	1628	Create File	C:\Windows\TaskscheMain.sdb	SUCCESS
04:49...	task sche.exe	1628	Create File	C:\Windows\tasksche.exe	SUCCESS
04:49...	task sche.exe	1628	Create File	C:\Windows\SysWOW64\win32u.dll	SUCCESS
04:49...	task sche.exe	1628	Create File	C:\Windows\SysWOW64\userbase.dll	SUCCESS
04:49...	task sche.exe	1628	Create File	C:\Windows\SysWOW64\usercpq.win.dll	SUCCESS
04:49...	task sche.exe	1628	Create File	C:\Windows\SysWOW64\gd32full.dll	SUCCESS
04:49...	task sche.exe	1628	Create File	C:\Windows\SysWOW64\gd32.dll	SUCCESS
04:49...	task sche.exe	1628	Create File	C:\Windows\SysWOW64\user32.dll	SUCCESS
04:49...	task sche.exe	1628	Create File	C:\Windows\SysWOW64\msvcrt.dll	SUCCESS
04:49...	task sche.exe	1628	Create File	C:\Windows\SysWOW64\vcprt4.dll	SUCCESS
04:49...	task sche.exe	1628	Create File	C:\Windows\SysWOW64\sechost.dll	SUCCESS
04:49...	task sche.exe	1628	Create File	C:\Windows\SysWOW64\adnsapi32.dll	SUCCESS
04:49...	task sche.exe	1628	Create File	C:\Windows\SysWOW64\imm32.dll	SUCCESS
04:49...	task sche.exe	1628	Create File	C:\Windows\SysWOW64\imm32.dll	SUCCESS
04:49...	task sche.exe	1628	Create File Mapp.	C:\Windows\SysWOW64\imm32.dll	FILE LOCKED WITH ONLY READERS
04:49...	task sche.exe	1628	Create File	C:\Windows\SysWOW64\imm32.dll	SUCCESS
04:49...	task sche.exe	1628	Create File	C:\ProgramData	SUCCESS
04:49...	task sche.exe	1628	Create File	C:\ProgramData	NAME COLLISION
04:49...	task sche.exe	1628	Create File	C:\ProgramData	SUCCESS
04:49...	task sche.exe	1628	Create File	C:\ProgramData\ochuttdll605	SUCCESS
04:49...	task sche.exe	1628	Create File	C:\ProgramData\ochuttdll605	SUCCESS
04:49...	task sche.exe	1628	Create File	C:\ProgramData\ochuttdll605\ochuttdll605	NAMED NOT FOUND
04:49...	task sche.exe	1628	Create File	C:\ProgramData\ochuttdll605\ochuttdll605	NAME NOT FOUND
04:49...	task sche.exe	1628	Create File	C:\Windows\tasksche.exe	SUCCESS
04:49...	task sche.exe	1628	Create File	C:\ProgramData\ochuttdll605\tasksche.exe	SUCCESS
04:49...	task sche.exe	1628	Create File	C:\Windows\SysWOW64\vtmarta.dll	SUCCESS
04:49...	task sche.exe	1628	Create File	C:\Windows\SysWOW64\vtmarta.dll	SUCCESS
04:49...	task sche.exe	1628	Create File Mapp.	C:\Windows\SysWOW64\vtmarta.dll	FILE LOCKED WITH ONLY READERS
04:49...	task sche.exe	1628	Create File Mapp.	C:\Windows\SysWOW64\vtmarta.dll	SUCCESS

Figure 23: tasksche.exe creates C:\ProgramData\ochuttdlb605 (hidden directory) (Procmon output)



(iii) Finding Host Based Indicators using Process Hacker 2

- Two services called mssecsvc2.0 and ochuttdlb605 were created:

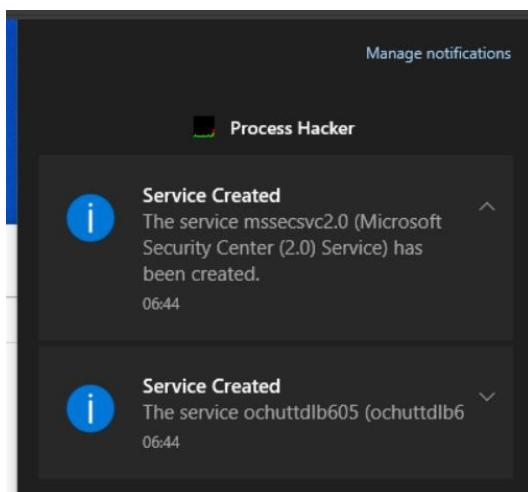


Figure 24: Process Hacker 2 output

- Two services called mssecsvc2.0 service implements Ransomware.wannacry.exe -m security (This service mainly checks if other systems in the network of the infected machine are vulnerable to eternal blue or not and infects them if they are vulnerable)

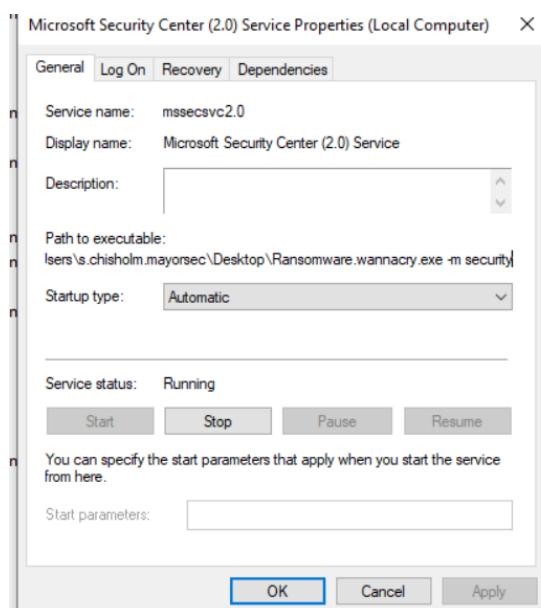


Figure 25: Screenshot of mssecsvc2.0 service (Windows Service)



(iv) Finding Network Based Indicators using TCPview

- ochuttdlb605 service runs tasksche.exe via cmd.exe which in turn creates the hidden directory C:\ProgramData\ochuttdlb605 where all the executables required by the malware is unpacked.

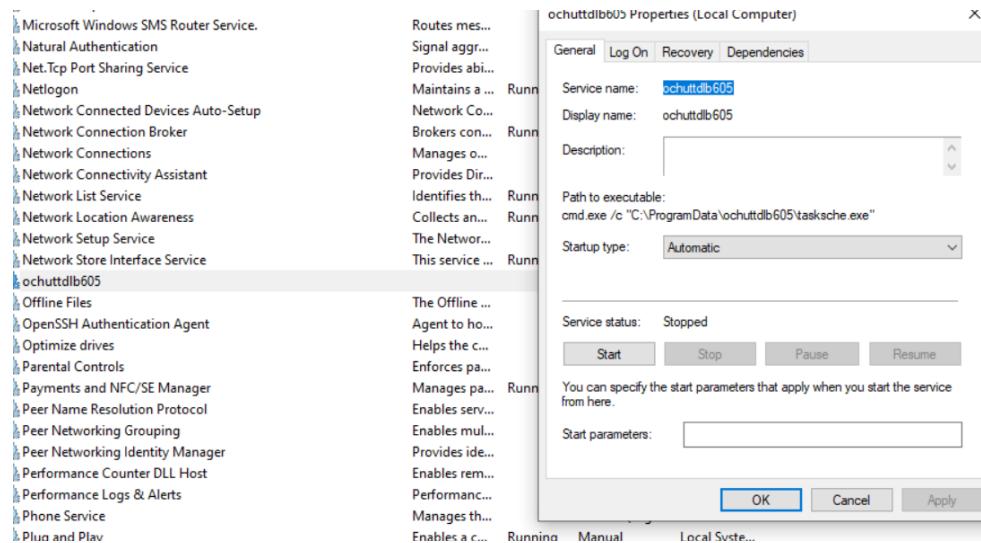


Figure 26: Screenshot of ochuttdlb605 service (Windows Service)

- C:\ProgramData\ochuttdlb605 directory containing all executables required by the malware.

Name	Date modified	Type	Size
msg	25-08-2023 04:49	File folder	
TaskData	25-08-2023 04:50	File folder	
@Please_Read_Me@.txt	25-08-2023 04:49	Text Document	1 KB
@WanaDecryptor@.exe	12-05-2017 02:22	Application	240 KB
00000000.eky	25-08-2023 04:49	Shortcut	1 KB
00000000.pky	25-08-2023 04:49	PKY File	0 KB
00000000.res	25-08-2023 04:58	RES File	1 KB
b.wnry	11-05-2017 20:13	WNRY File	1,407 KB
c.wnry	25-08-2023 04:50	WNRY File	1 KB
f.wnry	25-08-2023 04:49	WNRY File	1 KB
r.wnry	11-05-2017 15:59	WNRY File	1 KB
s.wnry	09-05-2017 16:58	WNRY File	2,968 KB
t.wnry	12-05-2017 02:22	WNRY File	65 KB
taskd.exe	12-05-2017 02:22	Application	20 KB
tasksche.exe	25-08-2023 04:49	Application	3,432 KB
tasks.e	12-05-2017 02:22	Application	20 KB
u.wnry	12-05-2017 02:22	WNRY File	240 KB

Figure 27: Screenshot of C:\ProgramData\ochuttdlb605 location

- msseccsvc2.0 service is checking if any of the other system in the network of the infected machine is vulnerable to eternal blue (eternal blue exploits SMBv1 protocol which runs in port 445 and hence we can see massive number of packets sent to port 445 to various addresses in the local network)

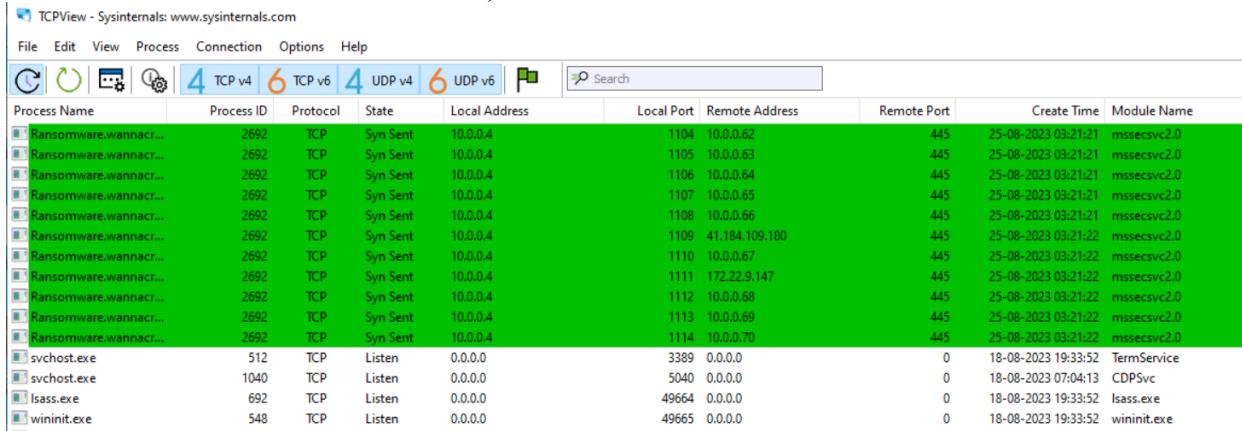


Figure 28: TCPView output

- taskhsvc.exe opens up local port 9050 and starts listening in it for remote connections.

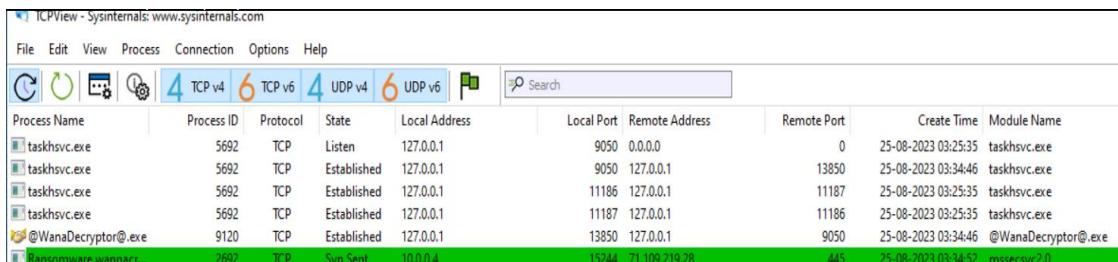


Figure 29: TCPView output (taskhsvc.exe)

(v) Overview of all the Processes using Procmon's processstree (Refer Figure 31 and 30)

1. Firstly two services are created i.e., msseccsvc2.0 and ochuttdlb605. Then both the services are run.
2. tasksche.exe runs (due to the ochuttdlb605 service) and it creates the hidden directory ochuttdlb605 in C:\ProgramData where all the executables necessary for the malware is unpacked. (The encryption routine begins and files begin to get encrypted)
3. attrib +h makes the current directory i.e., C:\ProgramData\ochuttdlb605 hidden

4. icacls . /grant Everyone:F /T /C /Q => Givens Full permission/control to everyone for the current directory i.e. C:\ProgramData\ochuttdlb605
5. After the files are continuosly AES encrypted they are stored in the original file location with WNCRY extension and original file name. taskdl.exe is used delete the original encrypted file created once it's copied to original file location. (It is periodically run several times)
6. 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 => This command utilizes vssadmin (Volume Shadow Copies-vss) to delete shadow copies of all the files and folders. It also uses wbadmin (Windows Backup-wb) and deletes all the backup catlogs which contain the information of backups in Windows backup.
7. cmd.exe /c reg add HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run /v "ochuttdlb605" /t REG_SZ /d "\"C:\ProgramData\ochuttdlb605\tasksche.exe\"" /f => Creates a service called ochuttdlb605 which runs tasksche.exe everytime a user log in (Persistence).
7. reg add HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run /v "ochuttdlb605" /t REG_SZ /d "\"C:\ProgramData\ochuttdlb605\tasksche.exe\"" /f => This command adds a registry entry named "ochuttdlb605" to the Windows Registry's "Run" key, specifying a program located at "C:\ProgramData\ochuttdlb605\tasksche.exe" to run at startup. The Run key makes the program run every time the user logs on (Persistence).
8. @WanaDecryptor@.exe launches taskhsvc.exe (a.k.a. tor.exe) which starts to listen in local port 9050 for remote connections.

```

Command
.... TaskData\Tor\taskhsvc.exe
.... \?C:\Windows\system32\conhost.exe 0xffffffff -ForceV1
.... cmd.exe /c start /b @WanaDecryptor@.exe vs
.... \?C:\Windows\system32\conhost.exe 0xffffffff -ForceV1
.... @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 -quiet
.... \?C:\Windows\system32\conhost.exe 0xffffffff -ForceV1
.... wmic shadowcopy delete
.... winlogon.exe
.... "fontdrvhost.exe"
.... "dwm.exe"
.... taskse.exe C:\ProgramData\ochuttdlb605\@WanaDecryptor@.exe
.... "C:\ProgramData\ochuttdlb605\@WanaDecryptor@.exe"
.... cmd.exe /c reg add HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run /v "ochuttdlb605" /t REG_SZ /d "\"C:\ProgramData\ochuttdlb605\tasksche.exe\"" /f
.... \?C:\Windows\system32\conhost.exe 0xffffffff -ForceV1
.... reg add HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run /v "ochuttdlb605" /t REG_SZ /d "\"C:\ProgramData\ochuttdlb605\tasksche.exe\"" /f
.... taskdl.exe
.... C:\Windows\system32\vssvc.exe
.... C:\Windows\system32\lsass.exe
.... "fontdrvhost.exe"
.... %SystemRoot%\system32\csrss.exe ObjectDirectory=\\Windows SharedSection=1024,20480,768 Windows=On SubSystemType=Windows ServerDll=basesrv,1 ServerDll=winsrv:UserServerDlInitialization,3 Server...
.... C:\Windows\Explorer.EXE
.... "C:\Windows\System32\SecurityHealthSystray.exe"
.... "C:\Windows\System32\VBoxTrav.exe"

```

Figure 30: Process tree view (procmon) during and post detonation



```
Command
.. C:\Users\s.chisholm.mayorsec\Desktop\Ransomware.wannacry.exe -m security
.. cmd.exe /c "C:\ProgramData\ochuttdlb605\tasksche.exe"
.. C:\ProgramData\ochuttdlb605\tasksche.exe
.. attrib +h .
.. \??\C:\Windows\system32\conhost.exe 0xffffffff -ForceV1
.. icacls . /grant Everyone:F /T /C /Q
.. \??\C:\Windows\system32\conhost.exe 0xffffffff -ForceV1
.. taskkl.exe
.. C:\Windows\system32\cmd.exe /c 89381693379808.bat
.. \??\C:\Windows\system32\conhost.exe 0xffffffff -ForceV1
.. cscript.exe //nologo m.vbs
.. taskkl.exe
.. taskkl.exe
.. taskkl.exe
.. @WanaDecryptor@.exe co
.. TaskData\Tor\taskhsvc.exe
.. \??\C:\Windows\system32\conhost.exe 0xffffffff -ForceV1
.. cmd.exe /c start /b @WanaDecryptor@.exe vs
.. \??\C:\Windows\system32\conhost.exe 0xffffffff -ForceV1
.. @WanaDecryptor@.exe vs
.. cmd.exe /c vssadmin delete shadows /all /quiet & wmic shadowcopy delete & bcdedit /set {default} bootstatuspolicy ignorefailures & bcdedit /set {default} recoveryenabled no & wbadmin delete catalog -quiet
.. \??\C:\Windows\system32\conhost.exe 0xffffffff -ForceV1
.. wmic shadowcopy delete
.. winlogon.exe
.. "fontdrv\host.exe"
.. "dwm.exe"
.. tasks.exe C:\ProgramData\ochuttdlb605\@WanaDecryptor@.exe
.. "C:\ProgramData\ochuttdlb605\@WanaDecryptor@.exe"
.. cmd.exe /c reg add HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run /v "ochuttdlb605" /t REG_SZ /d "\"C:\ProgramData\ochuttdlb605\tasksche.exe\""" /
.. \??\C:\Windows\system32\conhost.exe 0xffffffff -ForceV1
.. reg add HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run /v "ochuttdlb605" /t REG_SZ /d "\"C:\ProgramData\ochuttdlb605\tasksche.exe\""" /
.. taskkl.exe
.. C:\Windows\system32\vssvc.exe
.. C:\Windows\system32\lsass.exe
.. "fontdrv\host.exe"
.. "%SystemRoot%\System32\csrss.exe ObjectDirectory=Windows SharedSection=1024,20480,768 Windows=On SubSystemType=Windows ServerDll=basesrv,1 ServerDll=winsrv:UserServerDllInitialization,3 Server
.. C:\Windows\Explorer.EXE
.. "C:\Windows\System32\SecurityHealthSystray.exe"
```

Figure 31: Process tree view (procmon) during and post detonation

Analyzing the decryption function

- @WanaDecryptor@.exe present in the desktop is the decryption function and it launches automatically after the malware executes.
- When we click decrypt, the executable decrypts random files as sample for us:

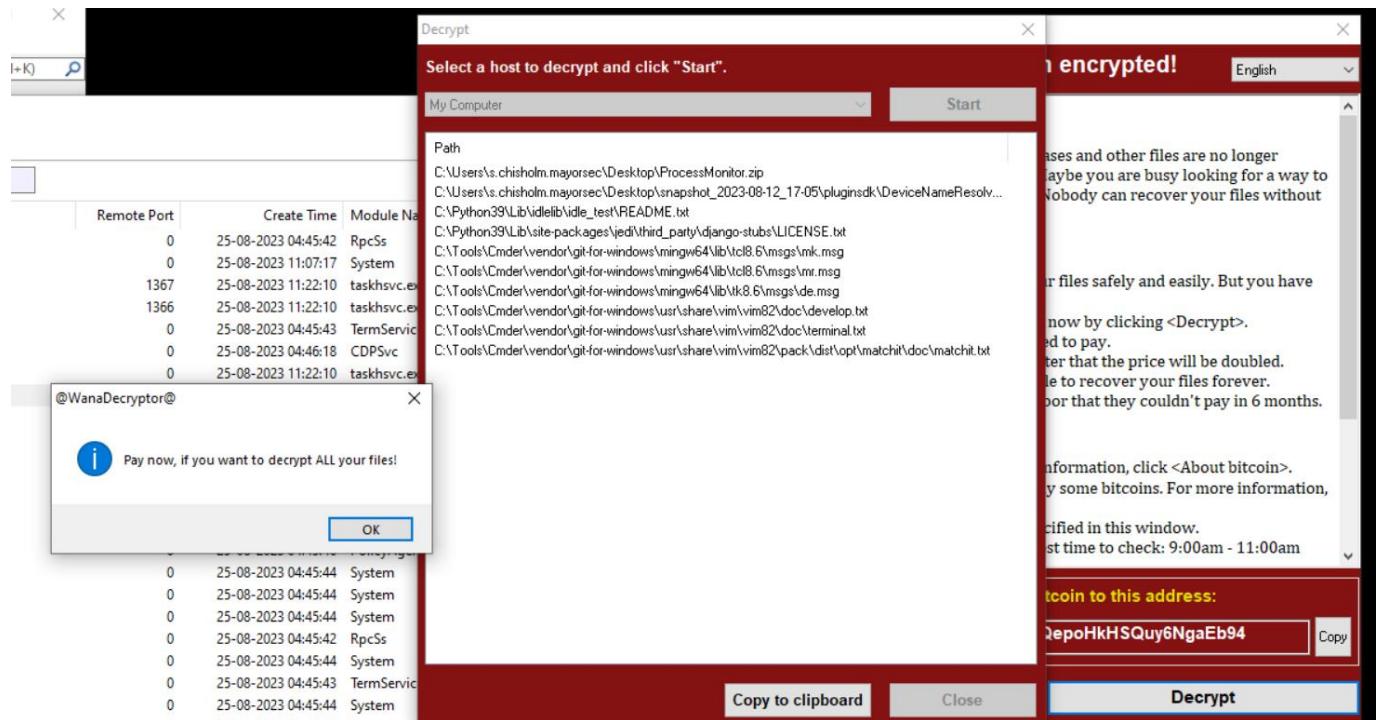


Figure 32: @WanaDecryptor@.exe decrypting certain number of random files as sample

©Please_Read_Me@.txt - Notepad

File Edit Format View Help

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 \$300 worth of bitcoin to this bitcoin address: 13AM4VW2dhxYgXeQepoHkHSQuy6NgaEb94

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 33: ransom note in the desktop containing the instructions for decryption and ransom payment



- Running @WanaDecryptor@.exe automatically launches taskhsvc.exe (a.k.a. tor.exe) which starts listening in port 9050 for remote connection from C2 server (tor browser is installed in the host for this purpose)

ONEDrive.exe	5049	1.164 MB	mayorsec\chisholm	MICROSOFT OFFICE			
msedge.exe	5440	42.73 MB	mayorsec\chisholm	Microsoft Edge			
msedge.exe	5548	2.09 MB	mayorsec\chisholm	Microsoft Edge			
msedge.exe	5844	10.99 MB	mayorsec\chisholm	Microsoft Edge			
msedge.exe	5872	8.55 MB	mayorsec\chisholm	Microsoft Edge			
msedge.exe	5896	6.22 MB	mayorsec\chisholm	Microsoft Edge			
msedge.exe	660	18.27 MB	mayorsec\chisholm	Microsoft Edge			
msedge.exe	6708	12.55 MB	mayorsec\chisholm	Microsoft Edge			
msedge.exe	6264	21.26 MB	mayorsec\chisholm	Microsoft Edge			
MicrosoftEdgeUpdate.exe	6468	1.91 MB		Microsoft Edge Update			
taskhsche.exe	528	41.14	72.23 kB/s	19.07 MB	DiskPart		
@WanaDecryptor@.exe	5052			1.86 MB	Load PerfMon Counters		
taskhsvc.exe	4856	0.02	176 B/s	7 MB			
conhost.exe	3812			6.13 MB	Console Window Host		
@WanaDecryptor@.exe	5480	0.38		2.38 MB	mayorsec\chisholm	Load PerfMon Counters	

Figure 34: @WanaDecryptor@.exe launching taskhsvc.exe (captured from processs tree of procmon)

- TCPview snapshot also shows how @WanaDecryptor@.exe launches taskhsvc.exe which starts listening in port 9050:

TCPView - Sysinternals: www.sysinternals.com								
File	Edit	View	Process	Connection	Options	Help	4 TCP v4	6 TCP v6
							4 TCP v4	6 TCP v6
Process Name	Process ID	Protocol	State	Local Address	Local Port	Remote Address	Remote Port	Create Time
taskhsvc.exe	5692	TCP	Listen	127.0.0.1	9050	0.0.0.0	0	25-08-2023 03:25:35
taskhsvc.exe	5692	TCP	Established	127.0.0.1	9050	127.0.0.1	13850	25-08-2023 03:34:46
taskhsvc.exe	5692	TCP	Established	127.0.0.1	11186	127.0.0.1	11187	25-08-2023 03:25:35
taskhsvc.exe	5692	TCP	Established	127.0.0.1	11187	127.0.0.1	11186	25-08-2023 03:25:35
@WanaDecryptor@.exe	9120	TCP	Established	127.0.0.1	13850	127.0.0.1	9050	25-08-2023 03:34:46
Ransomware.wanacryptor...	2693	TCP	Syn Sent	10.0.0.4	15344	71.109.219.28	445	25-08-2023 03:34:53

Figure 35: TCPview Output

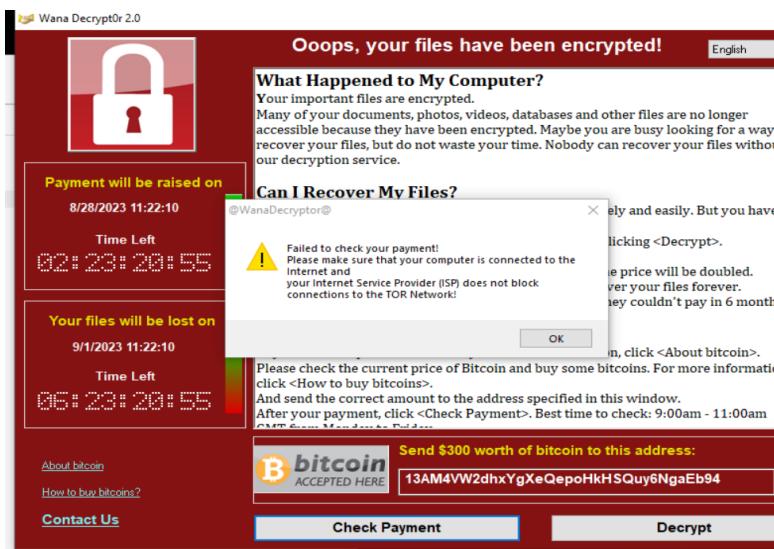


Figure 36: clicking check payment in @WanaDecryptor@.exe (we aren't connected to the internet and hence showed that message)



Advanced Static Analysis

- I utilized cutter's graphical view for advanced static analysis
- hxxp[://] www[.]jiuquerfsodp9ifjaposdfjhgosurijfaewrwerwgwea[.]com/ → This URL is stored in ESI register which is the killswitch domain
- Using InternetOpenUrlA API a http/https request is sent to the above url which is stored in the ESI register.
- Now the result of InternetOpenUrlA is stored in eax register and the value in EAX register is stored in EDI register (the result of InternetOpenUrlA is either 1 or 0, 1 if it can reach the URL and 0 if it can't)
- Now if edi is 1 → test edi edi will return a value 1 and hence ZE=0 (URL is reachable)
- if edi is 0 → test edi edi will return a value 0 and hence ZE=1 (URL is unreachable)
- jne 0x4081bc → jumps if ZE=0 ie if the zero flag is not set (URL is reachable)

```
0x004081a3 mov    esi, str.http://www.jiuquerfsodp9ifjaposdfjhgosurijfaewrwerwgwea.com ; 0x4313d0
0x004081a4 lea    edi, [var_50h]
0x004081a5 xor    eax, eax
0x004081a6 rep    movsd dword es:[edi], dword ptr [esi]
0x004081a7 movsb byte es:[edi], byte ptr [esi]
0x004081a8 mov    dword [var_17h], eax
0x004081a9 mov    dword [var_13h], eax
0x004081a0 mov    dword [var_fh], eax
0x004081a1 mov    dword [var_bh], eax
0x004081a2 mov    dword [var_7h], eax
0x004081a3 mov    dword [var_3h], eax
0x004081a4 push   eax
0x004081a5 push   eax
0x004081a6 push   eax
0x004081a7 push   1           ; 1
0x004081a8 push   eax
0x004081a9 mov    byte [var_1h], al
0x004081a9 call   dword [InternetOpenA] ; 0x40a134
0x004081a9 push   0
0x004081a9 push   0x84000000
0x004081a9 push   0
0x004081a9 lea    ecx, [var_64h]
0x004081a9 mov    esi, eax
0x004081a9 push   0
0x004081a9 push   ecx
0x004081a9 push   esi
0x004081a9 call   dword [InternetOpenUrlA] ; 0x40a138
0x004081a9 mov    edi, eax
0x004081a9 push   esi
0x004081a9 mov    esi, dword [InternetCloseHandle] ; 0x40a13c
0x004081a9 test   edi, edi
0x004081a9 jne   0x4081bc
0x004081a9 call   esi
0x004081a9 push   0
0x004081a9 call   esi
0x004081a9 call   fcn.00408090 ; fcn.00408090
0x004081a9 pop    edi
0x004081a9 xor    eax, eax
0x004081a9 pop    esi
0x004081a9 add    esp, 0x50
0x004081a9 ret    0x10
0x004081a9 call   esi
0x004081a9 ret
```

Figure 37: Cutter's graphical view

```
0x004081a3 test   edi, edi
0x004081a5 jne   0x004081bc
[0x004081a7]
0x004081a7 call   esi
0x004081a9 push   0
0x004081a9 call   esi
0x004081a9 call   fcn.00408090 ; fcn.00408090
0x004081a9 pop    edi
0x004081a9 xor    eax, eax
0x004081a9 pop    esi
0x004081a9 add    esp, 0x50
0x004081a9 ret    0x10
[0x004081bc]
0x004081bc call   esi
0x004081be push   edi
0x004081bf call   esi
0x004081c1 pop    edi
0x004081c2 xor    eax, eax
0x004081c4 pop    esi
0x004081c5 add    esp, 0x50
0x004081c8 ret    0x10
```

Figure 38: Cutter's graphical vi

Advanced Dynamic Analysis

- We are now clear that the malware doesn't get executed if it can call back to its killswitch domain (`hxxp[://] www[.]iuquerfsodp9ifjaposdfjhgosurijfaewrwegwea[.]com`)
- So if we are either connected to internet/inetsim , the malware won't get executed.
- For the malware to get executed even if it calls to its killswitch domain, we need execute it within a debugger.
- In the debugger just before the jump statement, we set the zero flag (`ZF=1`) so that the jump is not taken: (`ZF=0` previously since we are connected internet/inetsim and hence callback is made to the URL and this results in output 1 when test edi, edi instruction is implemented which doesn't set the zero flag)

Screenshot of the x32Dbg debugger showing assembly code for Ransomware.wannacry.exe. The assembly code is as follows:

```

Ransomware.wannacry.exe - PID: 4776 - Module: ransomware.wannacry.exe - Thread: Main Thread 7164 - x32dbg
File View Debug Tracing Plugins Favourites Options Help Aug 12 2023 (TitanEngine)
CPU Log Notes Breakpoints Memory Map Call Stack SEH Script Symbols Source References Threads Handles Trace
0040814A BE D0134300 mov esi,ransomware.wannacry_4313D0
0040814F 8D7C24 08 lea edi,dword ptr ss:[esp+8]
00408153 33C0 xor eax,eax
00408155 F7AS rep movsd
00408157 A1 mov byte ptr ss:[edi],al
00408158 894424 41 mov dword ptr ss:[esp+41],eax
0040815C 894424 45 mov dword ptr ss:[esp+45],eax
00408160 894424 49 mov dword ptr ss:[esp+49],eax
00408164 894424 4D mov dword ptr ss:[esp+4D],eax
00408168 894424 51 mov dword ptr ss:[esp+51],eax
0040816C 66:894424 55 mov word ptr ss:[esp+55],ax
00408171 50 push eax
00408172 50 push eax
00408173 50 push eax
00408174 6A 01 push 1
00408176 50 push eax
00408177 884424 6B mov byte ptr ss:[esp+6B],al
0040817B FF15 34A14000 call dword ptr ds:[<InternetopenA>]
00408181 6A 00 push 0
00408183 68 00000084 push 84000000
00408188 6A 00 push 0
0040818A 8D4C24 14 lea ecx,dword ptr ss:[esp+14]
0040818E 8BF0 mov edi,esi
00408190 6A 00 push 0
00408192 91 push ecx
00408193 56 push esi
00408194 FF15 38A14000 call dword ptr ds:[<InternetopenUrlA>]
0040819A 8BF8 mov edi,eax
0040819C 56 push esi
0040819D 8B35 3CA14000 mov esi,dword ptr ds:[<InternetCloseHandle>]
004081A3 85FF test edi,edi
004081A5 75 15 jne ransomware.wannacry_4081BC
004081A6 FFD6 call esi
004081A9 6A 00 push 0
004081AB FFD6 call esi
004081AD E8 DEFEFFFF call ransomware.wannacry_408090
004081B2 5F pop edi
004081B3 33C0 xor eax,eax
004081B5 SE pop esi
004081B6 83C4 50 add esp,50
004081B9 C2 1000 ret 10
004081BC FFD6 call esi
004081BE 57 push edi
004081C0 FFD6 call esi
004081C1 89C8 non opf

```

The assembly code shows various memory operations, function calls, and register manipulations. The instruction at address `004081A5` is a `jne` (jump if not equal) instruction that跳转到 `ransomware.wannacry_4081BC`. This instruction is highlighted in yellow, indicating it is the target of the analysis.

Figure 39: Setting the zero flag in debugger (x32Dbg.exe so jump won't be taken)

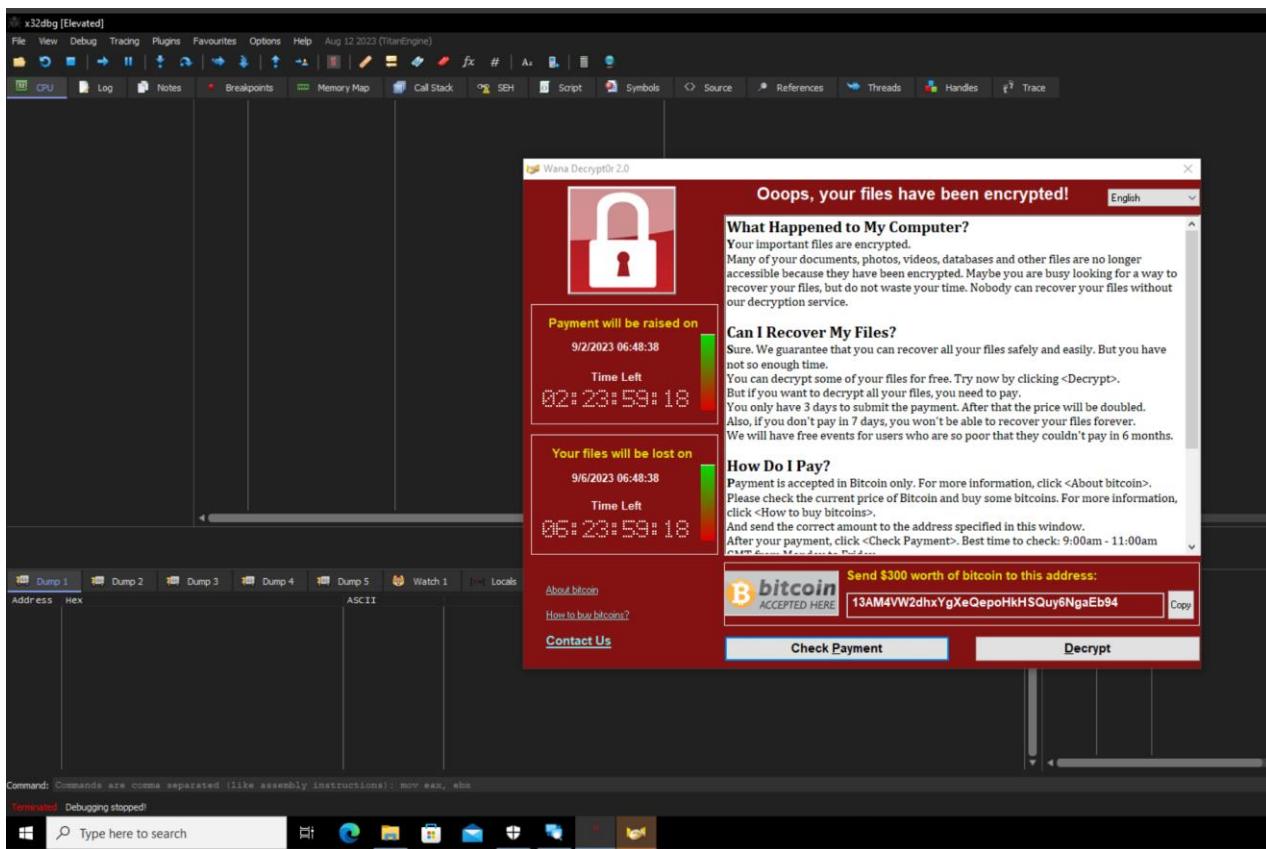


Figure 40: Malware getting executed after setting zero flag to 1 while connected to inetsim

Overview of the Malware:

- The malware executable needs to run with administrative privilege for it to execute. In addition to this connection to internet or intesim shouldn't be there so that callback to the killswitch domain (hxxp [://] www[.]iuquerfsodp9ifjaposdfjhgosurijfaewrwegwea[.]com) is not made.
- 2 services are created first i.e., mssecsvc2.0 and ochuttdlb605.
- mssecsvc2.0 service checks if other systems in the network of the infected machine are vulnerable to eternal blue or not. It does some scanning in port 445 for this.
- ochuttdlb605 service launches tasksche.exe via cmd.exe. tasksche.exe in turns creates a hidden directory C:\ProgramData\ochuttdlb605. All executables required by the malware is added to this location. The encryption routine begins.
- taskhsvc.exe opens up port 9050 and starts listening in it.
 - The shadow copies and file backups are deleted using vssadmin and wbadmin.
 - The files get encrypted and get appended with the extension “.WNRY”.@WanaDecryptor@.exe gets added to the desktop. Alongside, taskdl.exe takes care of the cleaning process. 2 iamges with .bmp extension get added to desktop and the desktop wallpaper is changed. A ransom note is added to the desktop as well.
 - Both the ransom notes and @WanaDecryptor@.exe contain the ransom amount and bitcoin wallet address to which the ransom should be paid.
 - When we click decrypt in @WanaDecryptor@.exe, it launches taskhsvc.exe and starts listening for in port 9050 and decrypts random files as a sample.
 - We also confirmed that tor.exe is copied to taskhsvc.exe and this service listens in port 9050 for C2 server (onion site) connection.

Rules & Signatures

- Firstly, we need include the file signature which in our case is MZ (for executables)
- The killswitch domain URL is the most important indicator that the sample we have is wannacry (2017 original version).
- We also saw strings like tasksche.exe (one of the most important executables), `\\\%s\IPC$` (IPC share being accessed) and wnr when we analyzed the strings present in the binary using floss. These strings too indicate that the executable we have is wannacry ransomware.

Appendices

A. Yara Rules

```
B.    rule Wannacry_Ransomware {
C.        meta:
D.            last_updated = "2023-09-09"
E.            author = "killswitch"
F.            description = "A sample Yara rule for Wannacry
Ransomware"
G.            Hash =
H.                "6C382A1C16DBA41B4FF6F0D728E9E92AEBFE2EE0C7FEB30A0E63B15EAF6C4B44"
I.            strings:
J.                $PE_magic_byte = "MZ"
K.                $string1 =
L.                    "http://www.iuqerfsodp9ifjaposdfjhgosurijfaewrwegwea.com"
M.                    $string2="tasksche.exe"
N.                    $string3="wnry"
O.                    $string4 = "\\\\$%s\\IPC$"
P.
Q.            condition:
R.                $PE_magic_byte at 0 and $string1 and $string2 and
$string3 and $string4
S.            }
```



B. Disassembled Code Snippet

```
[0x00408140]
int main (int argc, char **argv, char **envp);
; var int32_t var_64h @ stack - 0x64
; var int32_t var_50h @ stack - 0x50
; var int32_t var_17h @ stack - 0x17
; var int32_t var_13h @ stack - 0x13
; var int32_t var_fh @ stack - 0xf
; var int32_t var_bh @ stack - 0xb
; var int32_t var_7h @ stack - 0x7
; var int32_t var_3h @ stack - 0x3
; var int32_t var_1h @ stack - 0x1
0x00408140    sub    esp, 0x50
0x00408143    push   esi
0x00408144    push   edi
0x00408145    mov    ecx, 0xe ; 14
0x0040814a    mov    esi, str.http://www.iuqerfsodp9ifjaposdfjhgosurijfaewrwegwea.com ; 0x4313d0
0x0040814f    lea    edi, [var_50h]
0x00408153    xor    eax, eax
0x00408155    rep    movsd dword es:[edi], dword ptr [esi]
0x00408157    movsb  byte es:[edi], byte ptr [esi]
0x00408158    mov    dword [var_17h], eax
0x0040815c    mov    dword [var_13h], eax
0x00408160    mov    dword [var_fh], eax
0x00408164    mov    dword [var_bh], eax
0x00408168    mov    dword [var_7h], eax
0x0040816c    mov    word [var_3h], ax
0x00408171    push   eax
0x00408172    push   eax
0x00408173    push   eax
0x00408174    push   1 ; 1
0x00408176    push   eax
0x00408177    mov    byte [var_1h], al
0x0040817b    call   dword [InternetOpenA] ; 0x40a134
0x00408181    push   0
0x00408183    push   0x84000000
0x00408188    push   0
0x0040818a    lea    ecx, [var_64h]
0x0040818e    mov    esi, eax
0x00408190    push   0
0x00408192    push   ecx
0x00408193    push   esi
0x00408194    call   dword [InternetOpenUrlA] ; 0x40a138
0x0040819a    mov    edi, eax
0x0040819c    push   esi
0x0040819d    mov    esi, dword [InternetCloseHandle] ; 0x40a13c
0x004081a3    test   edi, edi
0x004081a5    jne    0x4081bc

[0x004081a7]
0x004081a7    call   esi
0x004081a9    push   0
0x004081ab    call   esi
0x004081ad    call   fcn.00408090 ; fcn.00408090
0x004081b2    pop    edi
0x004081b3    xor    eax, eax
0x004081b5    pop    esi
0x004081b6    add    esp, 0x50
0x004081b9    ret    0x10

[0x004081bc]
0x004081bc    call   esi
0x004081be    push   edi
0x004081bf    call   esi
0x004081c1    pop    edi
0x004081c2    xor    eax, eax
0x004081c4    pop    esi
0x004081c5    add    esp, 0x50
0x004081c8    ret    0x10
```