WannaCry Ransomware Analysis

Author: Ahmed Mostafa Elmeniawy

Contents

Executive Summary 3

General Information 3

Technical Analysis 4

Static Analysis……...4

File Analysis……..4

Code Analysis……...5

Encryption……7

Dynamic Analysis……..7

Behavioral Analysis 9

Dependencies 10

Indicators of Compromise (IoCs) 10

Executive Summary

In May 2017, the WannaCry ransomware crypto worm caused a global cyberattack, targeting computers running the Microsoft Windows operating system. The ransomware encrypted data on infected systems and demanded ransom payments in Bitcoin to decrypt the files1. WannaCry exploited a vulnerability known as **EternalBlue**, which was developed by the United States National Security Agency (NSA) and later leaked by a group called The Shadow Brokers.

General Information

- **MD5 Hash:** “db349b97c37d22f5ea1d1841e3c89eb4”.
- **SHA256 Hash:** “24d004a104d4d54034dbcffc2a4b19a11f39008a575aa614ea04703480b1022c”.
- **SHA1 Hash:** “e889544aff85ffaf8b0d0da705105dee7c97fe26”.
- **File Type:** Executable File.
- **Category:** Crypto Ransomware.

Technical Analysis

Static Analysis

File Analysis

Name	Raw Addr.	Raw size	Virtual Addr.	Virtual Size	Characteristics	Ptr to Reloc.	Num. of Reloc.	Num. of Linenum.
.text	1000	9000	1000	8BCA	60000020	0	0	0
.rdata	A000	1000	A000	998	40000040	0	0	0
.data	B000	27000	B000	30489C	C0000040	0	0	0
.rsrc	32000	35B000	310000	35A454	40000040	0	0	0

After opening the malware with PE-Bear tool, we compared the raw size with virtual size of different sections in the malware file and we found that the malware is not packed.

Offset	Name	Func. Count	Bound?	OriginalFirstThunk	TimeDateStamp	Forwarder	NameRVA	FirstThunk
A1E0	KERNEL32.dll	32	FALSE	A2B0	0	0	A62A	A030
A1F4	ADVAPI32.dll	11	FALSE	A280	0	0	A724	A000
A208	WS2_32.dll	13	FALSE	A3C4	0	0	A732	A144
A21C	MSVCP60.dll	2	FALSE	A334	0	0	A772	A0B4
A230	iphlpapi.dll	2	FALSE	A3FC	0	0	A7A4	A17C
A244	WININET.dll	3	FALSE	A3B4	0	0	A7EC	A134
A258	MSVCRT.dll	28	FALSE	A340	0	0	A8A0	A0C0

- These are the imported libraries and with a simple search we found that **WS2_32.dll**, **iphlpapi.dll** and **WININET.dll** libraries deal with networks.
- The libraries **KERNEL32.dll** and **ADVAPI32.dll** modify the system.

indicator (48)	detail	level
file > embedded	signature: executable, location: .data, offset: 0x0000B020, size: 5263716	1
file > embedded	signature: executable, location: .data, offset: 0x0000F080, size: 5297524	1
resource > size > suspicious	R,1831, 3514368 bytes	1
file > embedded	signature: executable, location: .rsrc, offset: 0x000320A4, size: 3514368	1
strings > URL	http://www.iugerfsodp9ifjaposdfjhgosurijfaewrwergwea.com	1
file > extensions (Ransomware | Wiper)	159	1
libraries > flag	Windows Socket Library	1
libraries > flag	IP Helper API	1
libraries > flag	Internet Extensions for Win32 Library	1
imports > flag	28	1
strings > size > suspicious	2039 bytes	2
strings > size > suspicious	1403 bytes	2
strings > size > suspicious	2693 bytes	2
strings > size > suspicious	3926 bytes	2
strings > size > suspicious	1554 bytes	2
strings > size > suspicious	1430 bytes	2
strings > size > suspicious	2988 bytes	2
resources > file-ratio	94.41%	2
file > entropy	7.964	3
file > signature	Microsoft Visual C++ v6.0	3
file > sha256sum	24D004A104D4D54034DBCFFC2A4B19A11F39008A575AA614EA04703480...	3
file > size	3723264 bytes	3
rich-header > checksum	0xC33D5D11	3
rich-header > offset	0x00000080	3
rich-header > hash	A1E1E9718FCD02EBA98E1C5D6E3EF166	3
file > tooling	Visual Studio 6.0	3

From PE-Studio tool, the red boxes indicate that there are embedded executable files in different locations and the yellow boxes show a URL and network indicators which may mean that the malware tries to reach this URL:
http://www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com.

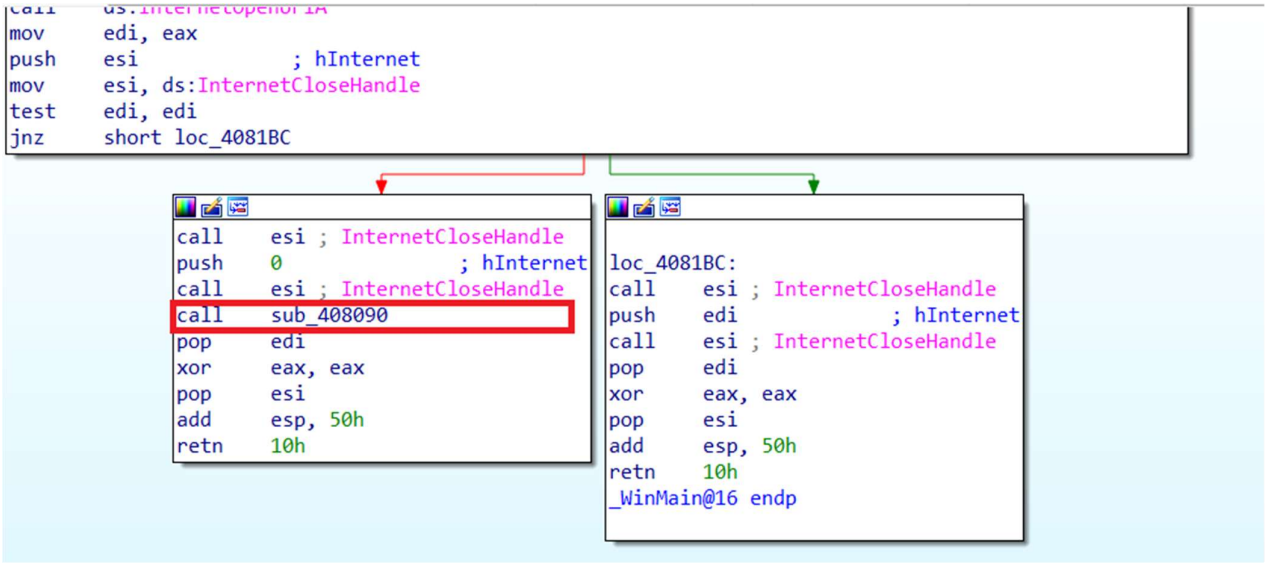
property	value
sha256	24D004A104D4D54034DBCFFC2A4B19A11F39008A575AA614EA04703480B1022C
sha1	E889544AFF85FFAF8B0D0DA705105DEE7C97FE26
md5	DB349B97C37D22F5EA1D1841E3C89EB4
first-bytes-hex	4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00 B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00 00
first-bytes-text	M Z . . . . . @ . . . . .
file-size	3723264 bytes
entropy	7.964
signature	Microsoft Visual C++ v6.0
tooling	Visual Studio 6.0
file-type	executable
cpu	32-bit
subsystem	GUI
file-version	6.1.7601.17514 (win7sp1_rtm.101119-1850)
description	Microsoft® Disk Defragmenter

- The entropy of this malware is high which indicates that the malware uses some kind of encryption and encoding.
- So before we continue our analysis, we need to put in mind these few findings:
  - The malware is not packed.
  - There are codes embedded in different locations from the main code.
  - The malware uses some sort of encryption or encoding.
  - The malware tries to reach a specific URL and uses network APIs.

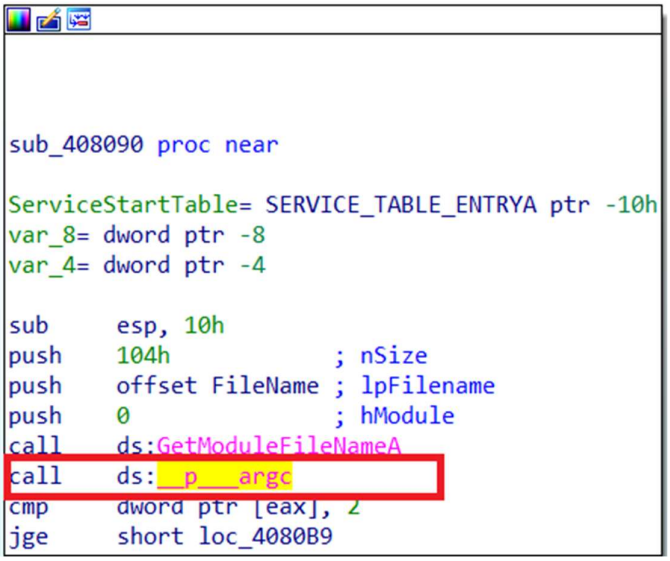
Code Analysis

```
mov     ecx, 0Eh
mov     esi, offset aHttpWwwIuqerfs ; "http://www.iuqerfsodp9ifjaposdfjhgosuri"...
lea     edi, [esp+58h+szUrl]
xor     eax, eax
rep movsd
movsb
mov     [esp+58h+var_17], eax
mov     [esp+58h+var_13], eax
mov     [esp+58h+var_F], eax
mov     [esp+58h+var_B], eax
mov     [esp+58h+var_7], eax
mov     [esp+58h+var_3], ax
push    eax             ; dwFlags
push    eax             ; lpszProxyBypass
push    eax             ; lpszProxy
push    1               ; dwAccessType
push    eax             ; lpszAgent
mov     [esp+6Ch+var_1], al
call    ds:InternetOpenA
push    0               ; dwContext
push    84000000h       ; dwFlags
push    0               ; dwHeadersLength
lea     ecx, [esp+64h+szUrl]
mov     esi, eax
push    0               ; lpszHeaders
push    ecx             ; lpszUrl
push    esi             ; hInternet
call    ds:InternetOpenUrlA
mov     edi, eax
push    esi             ; hInternet
mov     esi, ds:InternetCloseHandle
test    edi, edi
jnz     short loc_4081BC
```

From main function, the malware tries to open the URL:
http://www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com
using **"InternetOpenUrlA"** function, stores the return value from **"eax"** register into **"edi"** register and uses the **"test"** instruction to compare if the result is zero or not.

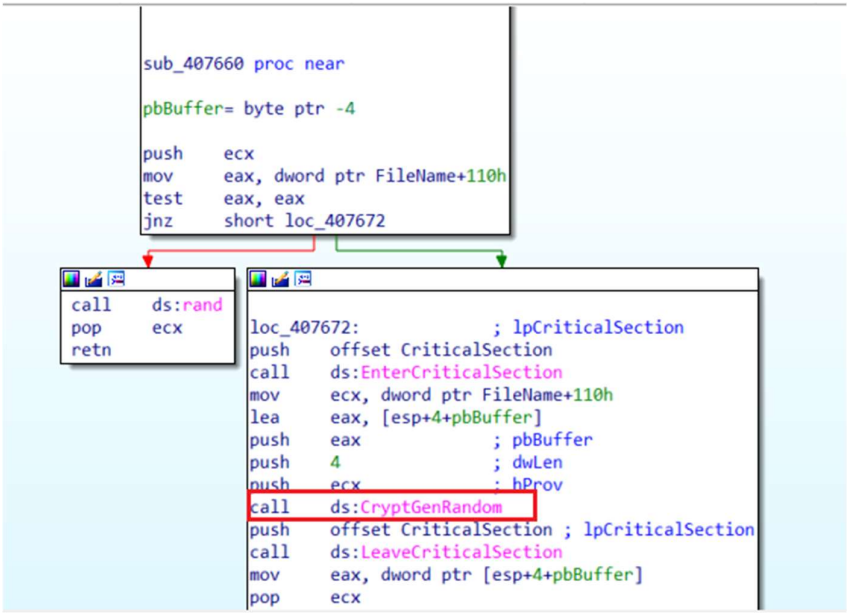


- If the return value is true (the URL opens successfully) , the malware will execute the true branch and terminates the program otherwise, the malware will execute the false branch and call the function **"sub_408090"** which we will analyze later in this report.
- The URL is random and doesn't exist which means that the malware will be executed every time the URL is unreachable.



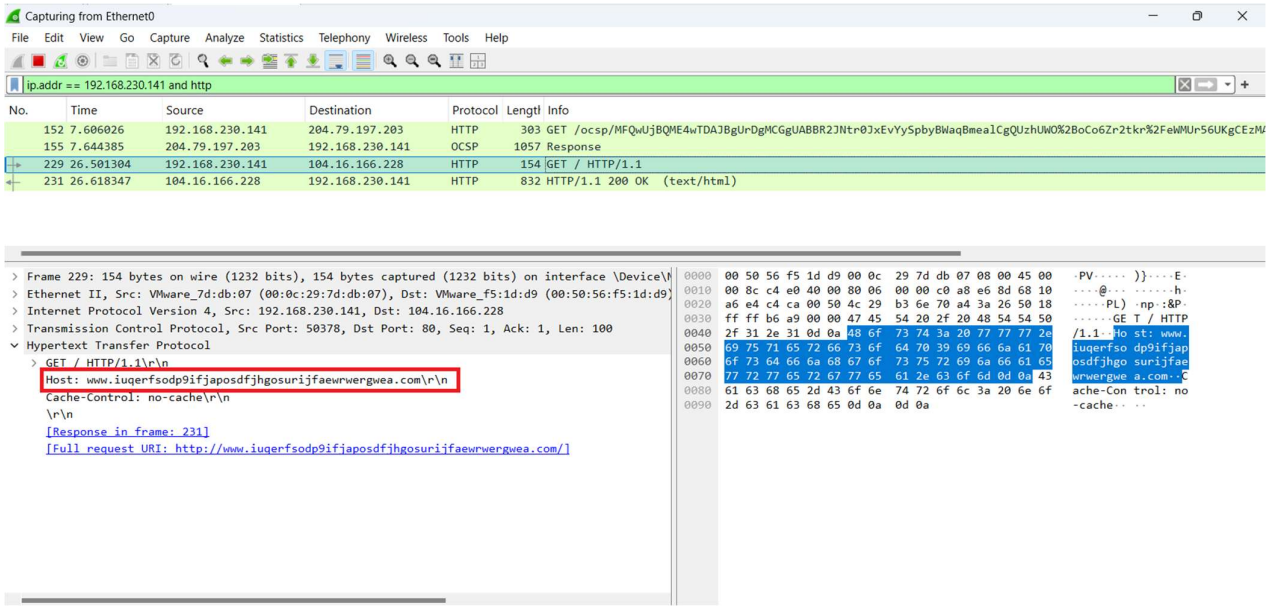
After going into the **"sub_408090"** function, I didn't understand what it is doing but it didn't go to the malicious code. So I will modify the flags in order to go to the malicious code.

Encryption

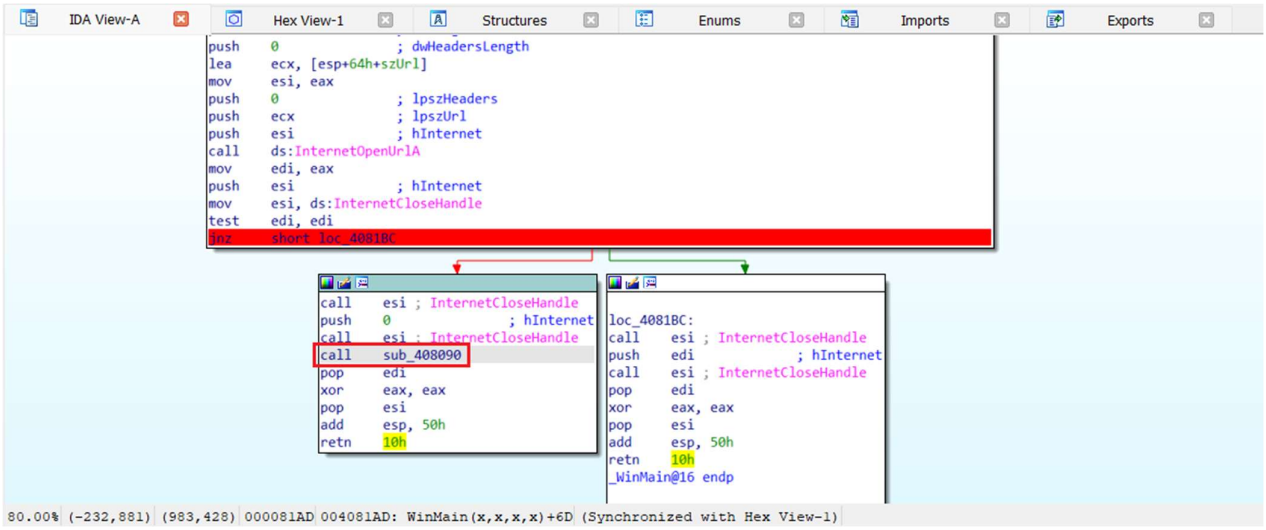


This part of the malware may be the code responsible for the encryption part as it takes a handle of the real encryption code.

Dynamic Analysis

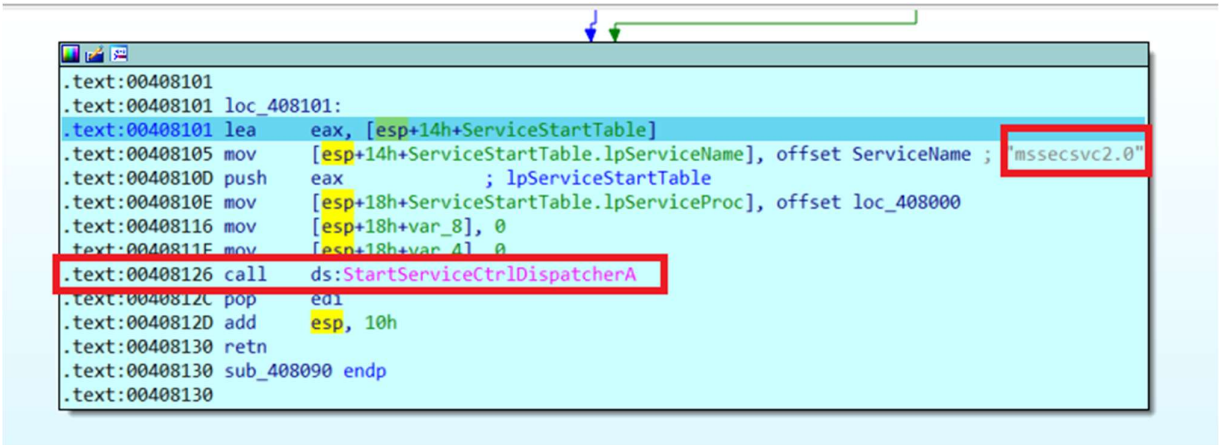


- In the above image, the packets captured after running the malware and we found that the malware tried to connect to the following URL: **www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com**.
- The malware uses this URL as a condition to check it will execute the ransomware code or not.



I set a breakpoint at the **"jnz"** instruction in order to control the execution flow of this ransomware and I changed the zero flag from 0 to 1 to go to the left block of code and get into the **"sub_408090**" function.

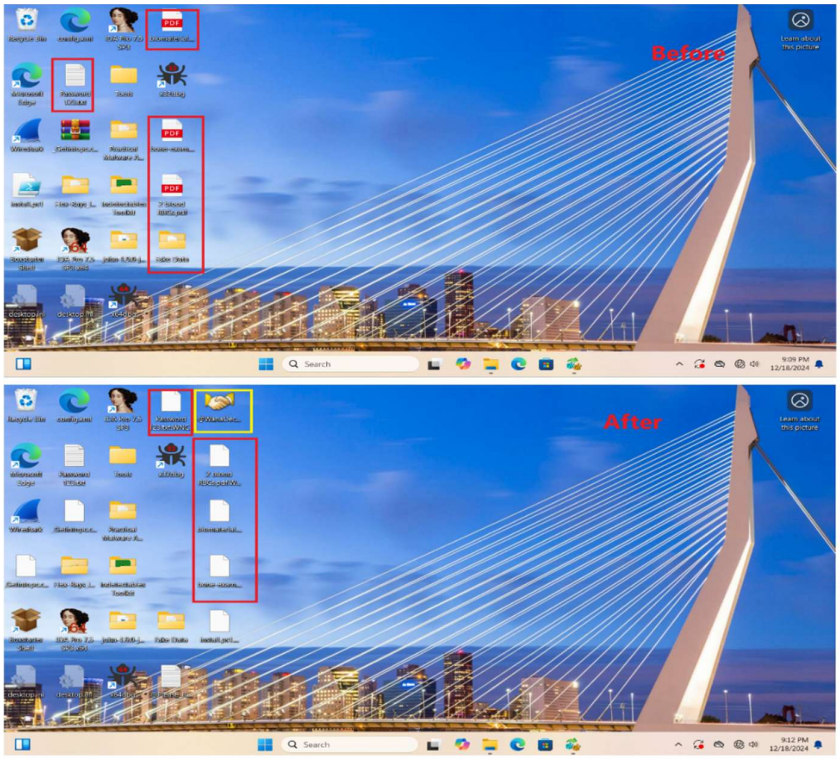
After getting into **"sub_408090"** function, I changed the sign flag from 1 to 0 in order to go to the right block of code.



The ransomware here begins to execute its code in a new service with a name **“mssecsvc2.0”** and encrypts all the data on the machine.

Behavioral Analysis

- After running the malware in our controlled environment with administrator privileges, I found a few files are created in desktop and this image is from **“@WanaDecryptor@.exe”** file.
- This malware has encrypted all of my files in this machine.



- In the above image, we observe the difference before and after running the malware. The existing files are encrypted by the ransomware and the yellow box is the **“@WanaDecryptor@.exe”** file that we discussed before and original files are deleted from the machine.
- Each encrypted file has a file extension **“.WNCRY”** and it is added after the original file name like **“FILE_NAME.pdf.WNCRY”**.

Dependencies

- **DLLs Imported into the Ransomware:**
    - KERNEL32.dll.
    - ADVAPI32.dll.
    - WS2_32.dll.
    - MSVCP60.dll.
    - iphlpapi.dll.
    - WININET.dll.
    - MSVCRT.dll.

Indicators of Compromise (IoCs)

- **URL:** “http://www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com”.
- **Extension File:** “.WNCRY”.
- **Executable File:** “@WanaDecryptor@.exe”.
- **Executable File:** “mssecsvc.exe”.
- **File:** “@Please_Read_Me@.txt”.
- **Executable File:** “tasksche.exe”.