# Practical Malware Analysis & Triage
# Malware Analysis Report

## WanaCry Ransomware Malware

28FEB23 | 0xNumb3rs | v1.0

# Table of Contents

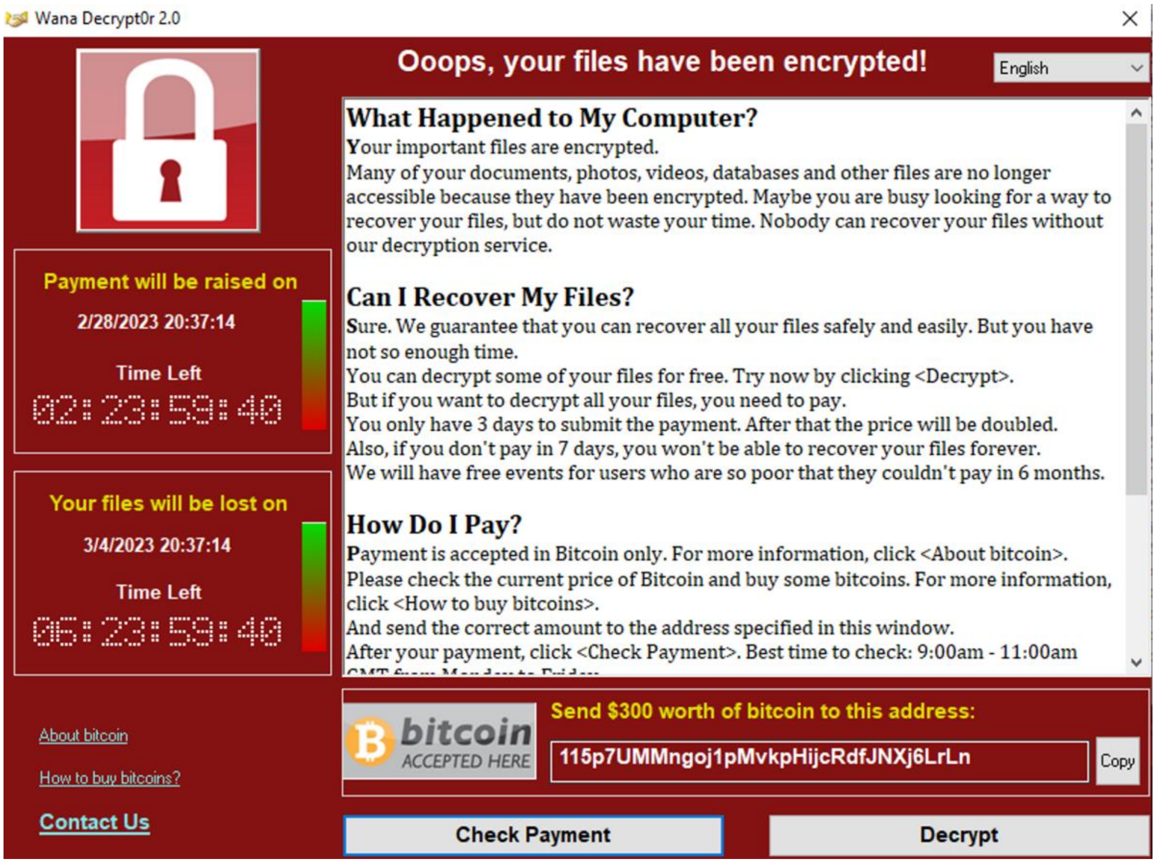
# Executive Summary

| | |
|---|---|
| SHA256 hash | 24d004a104d4d54034dbcffc2a4b19a11f39008a575aa614ea04703480b1022c |

WannaCry, or also known as WannaCrypt was a large-scale ransomware attack that happened in May 2017. It has viciously spread across computer networks in over 150 countries, infecting hundreds and thousands of computers. The attack exploited major vulnerability in Microsoft Windows using the Eternal Blue methodology exploiting the SMBv1 port 445. The Malware was able to encrypt the victims file through various symmetric and asymmetric encryption to encrypt the files on a victim computer.

YARA signature rules are attached in Appendix A. Malware sample and hashes have been submitted to VirusTotal for further examination.

# High-Level Technical Summary



Ooops, your files have been encrypted!

**What Happened to My Computer?**

Your important files are encrypted.
Many of your documents, photos, videos, databases and other files are no longer accessible because they have been encrypted. Maybe you are busy looking for a way to recover your files, but do not waste your time. Nobody can recover your files without our decryption service.

**Can I Recover My Files?**

Sure. We guarantee that you can recover all your files safely and easily. But you have not so enough time.
You can decrypt some of your files for free. Try now by clicking <Decrypt>.
But if you want to decrypt all your files, you need to pay.
You only have 3 days to submit the payment. After that the price will be doubled.
Also, if you don't pay in 7 days, you won't be able to recover your files forever.
We will have free events for users who are so poor that they couldn't pay in 6 months.

**How Do I Pay?**

Payment is accepted in Bitcoin only. For more information, click <About bitcoin>.
Please check the current price of Bitcoin and buy some bitcoins. For more information, click <How to buy bitcoins>.
And send the correct amount to the address specified in this window.
After your payment, click <Check Payment>. Best time to check: 9:00am - 11:00am GMT from Monday to Friday.

Send $300 worth of bitcoin to this address:

115p7UMMngoj1pMvkpHijcRdfJNXj6LrLn

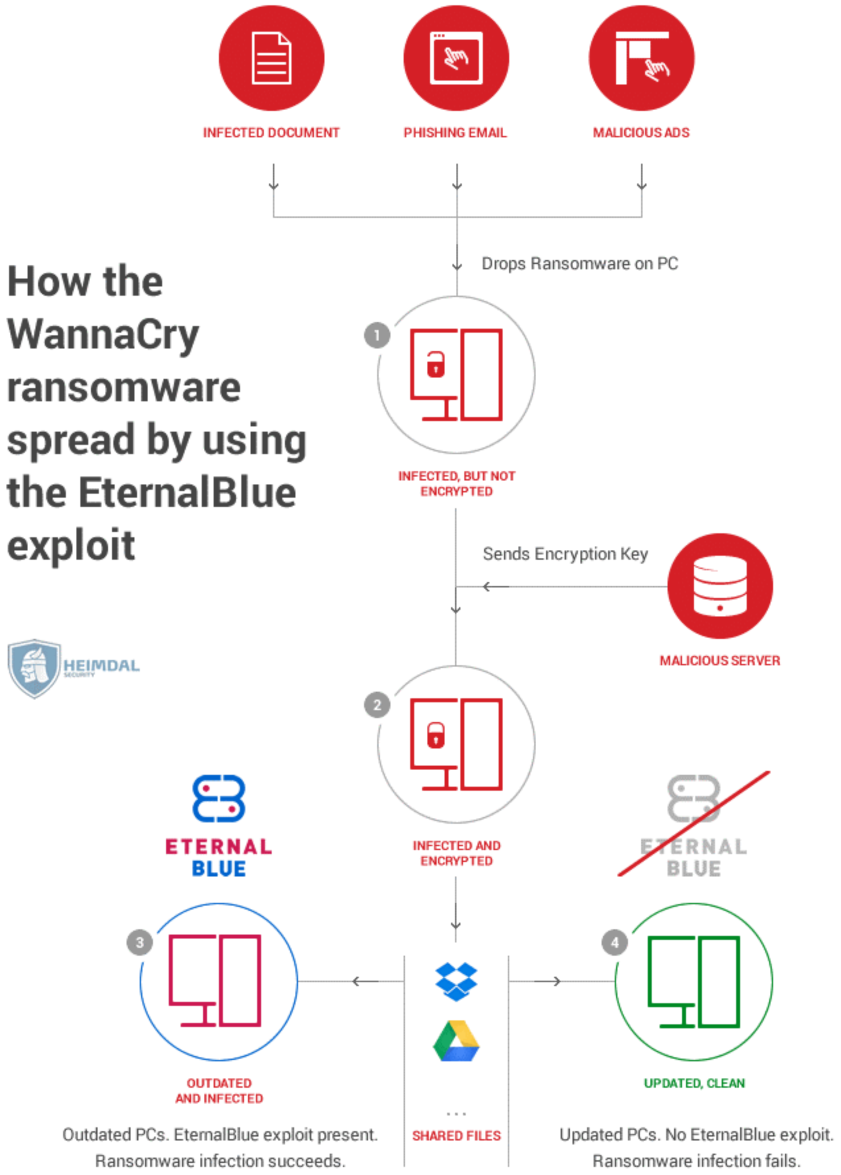
WannaCry is a type of malware known as ransomware that was first detected in May 2017. It spread rapidly across the globe and infected hundreds of thousands of computers in over 150 countries.

The malware was able to exploit a vulnerability in Microsoft Windows called Eternal Blue, which was developed by the United States National Security Agency (NSA). Eternal Blue exploited a flaw in the Windows Server Message Block (SMB) protocol, allowing the malware to spread from one computer to another on the same network.

Once a computer was infected with WannaCry, the malware would encrypt the victim's files using both symmetric and asymmetric encryption, effectively rendering them inaccessible. The attackers would then demand a ransom in exchange for a decryption key to unlock the files.

The WannaCry attack was notable not only for its widespread impact but also for the fact that it targeted critical infrastructure, such as healthcare systems and transportation networks. It served as a wake-up call for the importance of cybersecurity and the need for companies and organizations to take proactive measures to protect their systems and data.

This document covers various techniques for analyzing WanaCry, including Basic Static Analysis, Basic Dynamic Analysis, Advanced Static Analysis, and Advanced Dynamic Analysis. These techniques involve examining the software's behavior, code, network activity, and other characteristics to gain insight into its purpose, and functionality. By combining these different types of analysis, researchers can develop a more comprehensive understanding of the software and better protect against potential threats.


INFECTED DOCUMENT
PHISHING EMAIL
MALICIOUS ADS
Drops Ransomware on PC
How the WannaCry ransomware spread by using the EternalBlue exploit
HEIMDAL SECURITY
1
INFECTED, BUT NOT ENCRYPTED
Sends Encryption Key
MALICIOUS SERVER
2
INFECTED AND ENCRYPTED
ETERNAL BLUE
ETERNAL BLUE
3
OUTDATED AND INFECTED
Outdated PCs. EternalBlue exploit present. Ransomware infection succeeds.
SHARED FILES
4
UPDATED, CLEAN
Updated PCs. No EternalBlue exploit. Ransomware infection fails.

(Reference: B. Soare 2020 WannaCry Ransomware Explained (heimdalsecurity.com))

# BASIC ANALYSIS

## Basic Static Analysis

The basic static analysis involved the use of FLOSS, & PEstudiou

### Strings

Some of the most interesting strings that were identified during the initial static analysis are as follows:

- C:\%s\qeriuwjhrf
- C:\%s\%s WINDOWS
- tasksche.exe
- CloseHandle
- WriteFile
- CreateFileA
- CreateProcessA

During analysis, some interesting strings were discovered which suggest that the malware may have created a new directory. A notable finding from the investigation was a set of strings that hint at the possibility of the malware having created a new directory. The examination revealed a set of strings that imply the malware could have established a new directory.

### API

- GetProcessWindowStation
- GetUserObjectInformationW
- GetLastActivePopup
- GetActiveWindow

During the initial scanning of FLOSS and PEview, its identified that these notable API were imported. These API's signifies:

- GetProcessWindowStation: This API function retrieves a handle to the current window station for the calling process. A window station is a secure object that contains a clipboard, a set of desktop objects, and one or more window stations.
- GetUserObjectInformationW: This API function retrieves information about a window station or desktop object associated with the calling thread's process. This function

can be used to retrieve a variety of information about the object, such as its name, type, and security descriptor.

- GetLastActivePopup: This API function retrieves a handle to the most recent active popup window owned by the specified window. A popup window is a window that is displayed in response to a user action, such as clicking a button.
- GetActiveWindow: This API function retrieves a handle to the active window (the foreground window) on the desktop. The active window is the window that the user is currently interacting with.

- CryptGenKey
- CryptDecrypt
- CryptEncrypt
- CryptDestroyKey
- CryptImportKey
- CryptAcquireContextA

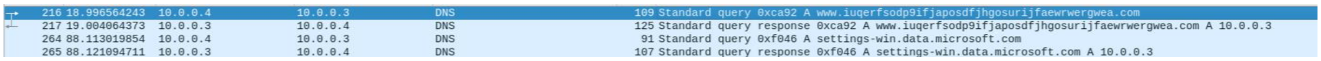
These are functions from the Microsoft Windows Cryptography API, which are used for cryptographic operations such as key generation, encryption, and decryption.
If these functions are found in a malware, it indicates that the malware is using cryptography to hide its activities and communication with its command-and-control server. The malware may be using these functions to encrypt its communications or to encrypt files on the infected system, making it difficult for security researchers to analyze the malware and for victims to recover their files. The use of these functions can also indicate that the malware authors have some knowledge of cryptography and are using it to make the malware more sophisticated and effective.

(Microsoft et al, Reference link https://learn.microsoft.com/en-us/windows/win32/desktop-programming)

## DNS

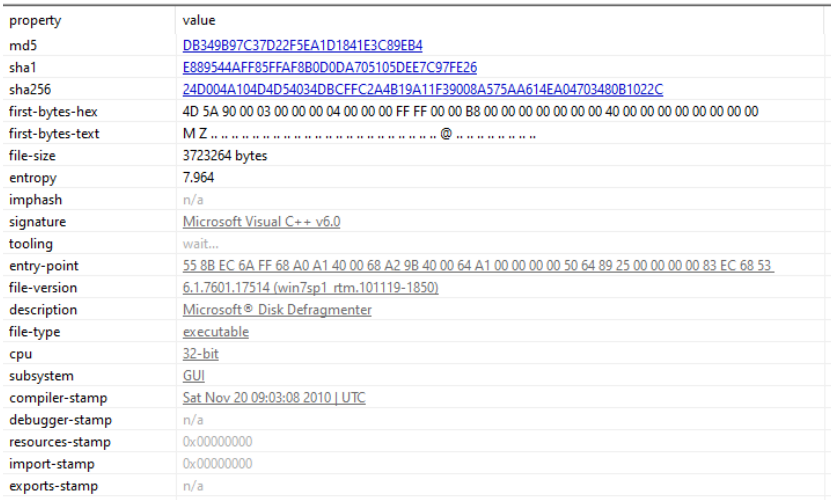
http://www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com

This link was identified during the scan, this link was used as part of the kill switch of the WanaCry program. This detail is to be explained further in the Advanced Analysis section.



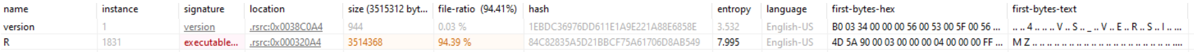
```
216 18.996564243  10.0.0.4    10.0.0.3    DNS    109 Standard query 0xca92 A www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com
217 19.004064373  10.0.0.3    10.0.0.4    DNS    125 Standard query response 0xca92 A www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com A 10.0.0.3
264 88.113019854  10.0.0.4    10.0.0.3    DNS     91 Standard query 0xf046 A settings-win.data.microsoft.com
265 88.121094711  10.0.0.3    10.0.0.4    DNS    107 Standard query response 0xf046 A settings-win.data.microsoft.com A 10.0.0.3
```

## PEview



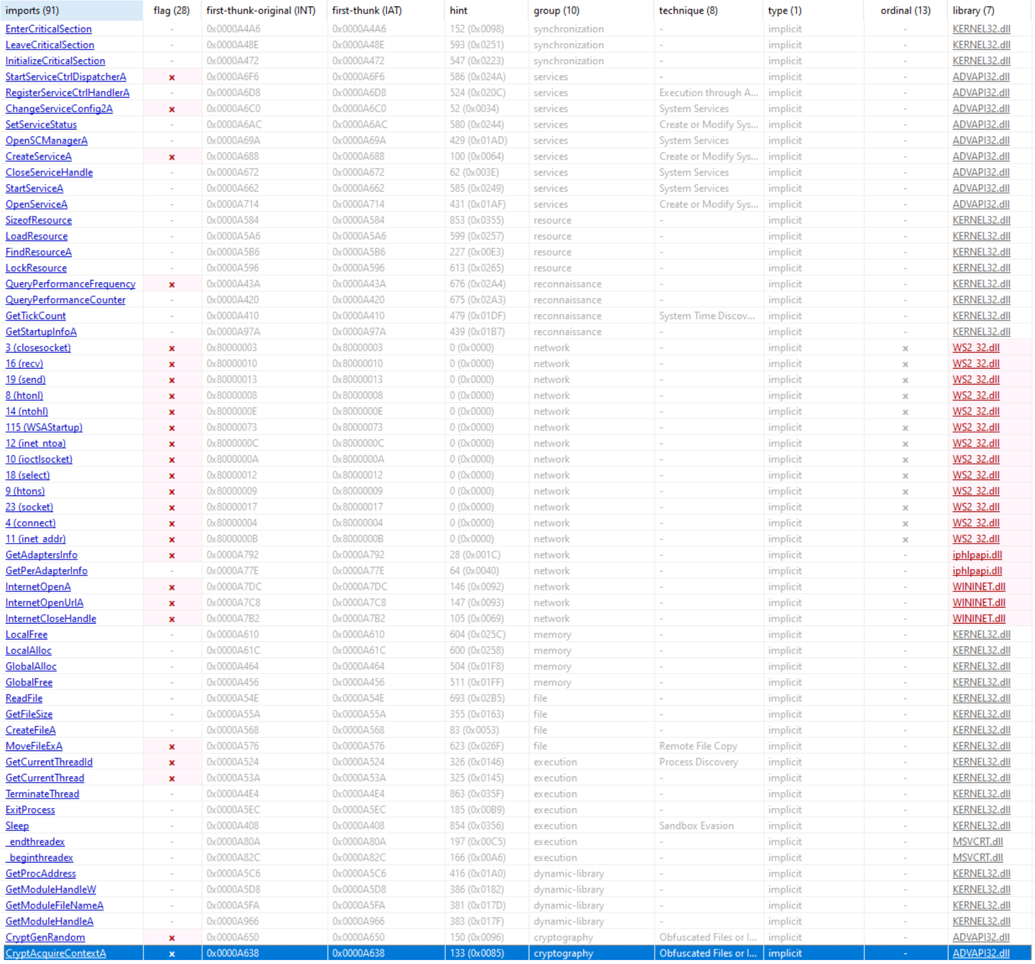
| property | value |
|---|---|
| md5 | DB349B97C37D22F5EA1D1841E3C89EB4 |
| sha1 | E889544AFF85FFAF8B0D0DA705105DEE7C97FE26 |
| sha256 | 24D004A104D4D54034DBCFFC2A4B19A11F39008A575AA614EA04703480B1022C |
| first-bytes-hex | 4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00 B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00 |
| first-bytes-text | M Z .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. @ .. .. .. .. .. .. .. .. .. .. .. |
| file-size | 3723264 bytes |
| entropy | 7.964 |
| imphash | n/a |
| signature | Microsoft Visual C++ v6.0 |
| tooling | wait... |
| entry-point | 55 8B EC 6A FF 68 A0 A1 40 00 68 A2 9B 40 00 64 A1 00 00 00 00 50 64 89 25 00 00 00 00 83 EC 68 53 |
| file-version | 6.1.7601.17514 (win7sp1_rtm.101119-1850) |
| description | Microsoft® Disk Defragmenter |
| file-type | executable |
| cpu | 32-bit |
| subsystem | GUI |
| compiler-stamp | Sat Nov 20 09:03:08 2010 \| UTC |
| debugger-stamp | n/a |
| resources-stamp | 0x00000000 |
| import-stamp | 0x00000000 |
| exports-stamp | n/a |

The information provided by PEStudio (as seen in the image) played a key role in the creation of the YARA rule. These details helped to support the identification of WanaCry.



| name | instance | signature | location | size (3515312 byt... | file-ratio (94.41%) | hash | entropy | language | first-bytes-hex | first-bytes-text |
|---|---|---|---|---|---|---|---|---|---|---|
| version | 1 | version | .rsrc:0x0038C0A4 | 944 | 0.03 % | 1EBDC36976DD611E1A9E221A88E6858E | 3.532 | English-US | B0 03 34 00 00 00 56 00 53 00 5F 00 56 ... | .. .. 4 .. .. .. V .. S .. _ .. V .. E .. R .. S .. I .. ... |
| R | 1831 | executable... | .rsrc:0x000320A4 | 3514368 | 94.39 % | 84C82835A5D21BBCF75A61706D8AB549 | 7.995 | English-US | 4D 5A 90 00 03 00 00 00 04 00 00 00 FF ... | M Z .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .... |

The executable hex code was used to conduct VirusTotal searches in order to identify other related signatures of the malware. This same hex code can also be incorporated into a YARA rule set to flag any identified PE files that contain similar hex code as the WanaCry Ransomware.



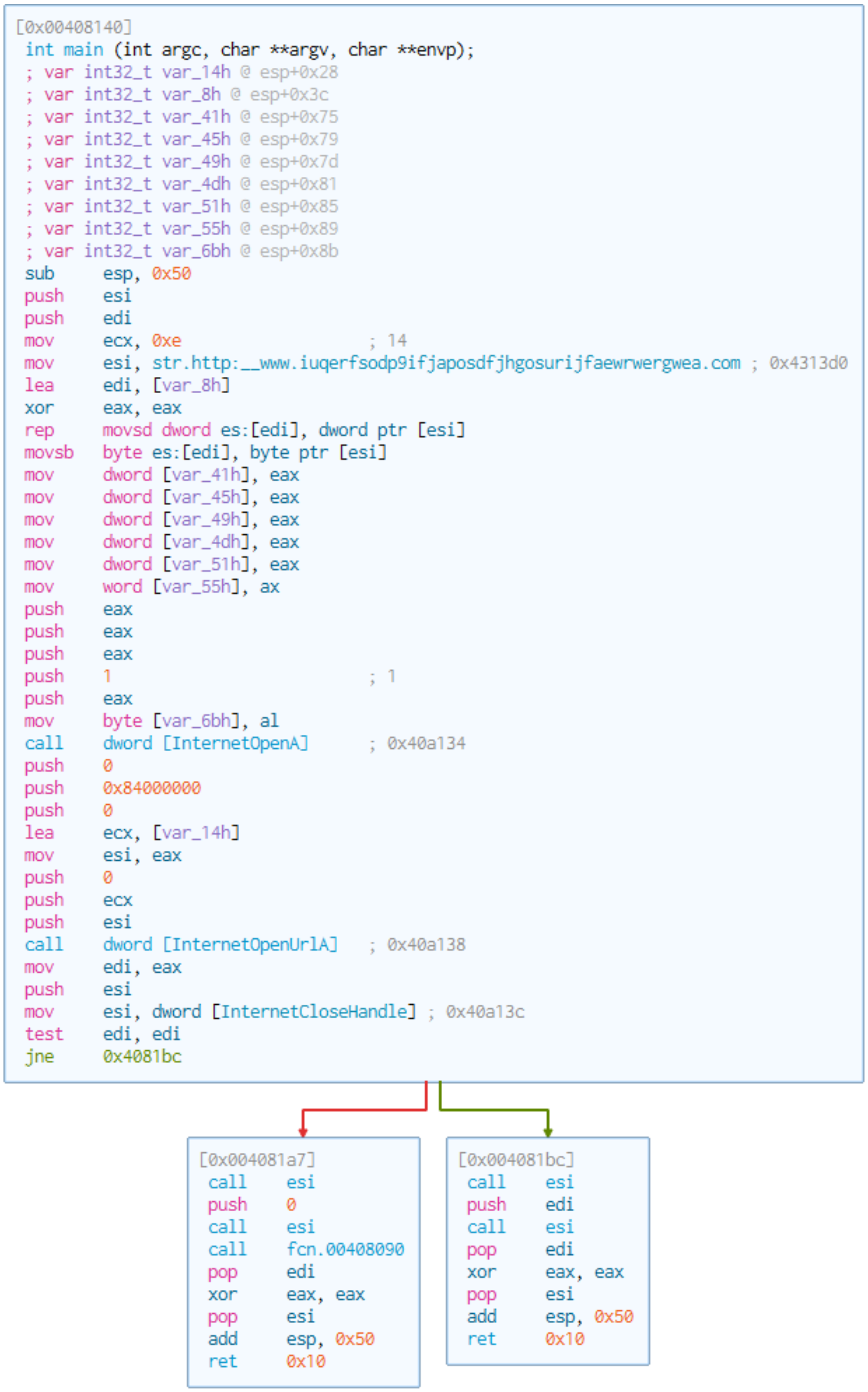
| imports (91) | flag (28) | first-thunk-original (INT) | first-thunk (IAT) | hint | group (10) | technique (8) | type (1) | ordinal (13) | library (7) |
|---|---|---|---|---|---|---|---|---|---|
| EnterCriticalSection | - | 0x0000A4A6 | 0x0000A4A6 | 152 (0x0098) | synchronization | - | implicit | - | KERNEL32.dll |
| LeaveCriticalSection | - | 0x0000A48E | 0x0000A48E | 593 (0x0251) | synchronization | - | implicit | - | KERNEL32.dll |
| InitializeCriticalSection | - | 0x0000A472 | 0x0000A472 | 547 (0x0223) | synchronization | - | implicit | - | KERNEL32.dll |
| StartServiceCtrlDispatcherA | x | 0x0000A6F6 | 0x0000A6F6 | 586 (0x024A) | services | - | implicit | - | ADVAPI32.dll |
| RegisterServiceCtrlHandlerA | - | 0x0000A6D8 | 0x0000A6D8 | 524 (0x020C) | services | Execution through A... | implicit | - | ADVAPI32.dll |
| ChangeServiceConfig2A | x | 0x0000A6C0 | 0x0000A6C0 | 52 (0x0034) | services | System Services | implicit | - | ADVAPI32.dll |
| SetServiceStatus | - | 0x0000A6AC | 0x0000A6AC | 580 (0x0244) | services | Create or Modify Sys... | implicit | - | ADVAPI32.dll |
| OpenSCManagerA | - | 0x0000A69A | 0x0000A69A | 429 (0x01AD) | services | System Services | implicit | - | ADVAPI32.dll |
| CreateServiceA | x | 0x0000A688 | 0x0000A688 | 100 (0x0064) | services | Create or Modify Sys... | implicit | - | ADVAPI32.dll |
| CloseServiceHandle | - | 0x0000A672 | 0x0000A672 | 62 (0x003E) | services | System Services | implicit | - | ADVAPI32.dll |
| StartServiceA | - | 0x0000A662 | 0x0000A662 | 585 (0x0249) | services | System Services | implicit | - | ADVAPI32.dll |
| OpenServiceA | - | 0x0000A714 | 0x0000A714 | 431 (0x01AF) | services | Create or Modify Sys... | implicit | - | ADVAPI32.dll |
| SizeofResource | - | 0x0000A584 | 0x0000A584 | 853 (0x0355) | resource | - | implicit | - | KERNEL32.dll |
| LoadResource | - | 0x0000A5A6 | 0x0000A5A6 | 599 (0x0257) | resource | - | implicit | - | KERNEL32.dll |
| FindResourceA | - | 0x0000A5B6 | 0x0000A5B6 | 227 (0x00E3) | resource | - | implicit | - | KERNEL32.dll |
| LockResource | - | 0x0000A596 | 0x0000A596 | 613 (0x0265) | resource | - | implicit | - | KERNEL32.dll |
| QueryPerformanceFrequency | x | 0x0000A43A | 0x0000A43A | 676 (0x02A4) | reconnaissance | - | implicit | - | KERNEL32.dll |
| QueryPerformanceCounter | - | 0x0000A420 | 0x0000A420 | 675 (0x02A3) | reconnaissance | - | implicit | - | KERNEL32.dll |
| GetTickCount | - | 0x0000A410 | 0x0000A410 | 479 (0x01DF) | reconnaissance | System Time Discov... | implicit | - | KERNEL32.dll |
| GetStartupInfoA | - | 0x0000A97A | 0x0000A97A | 439 (0x01B7) | reconnaissance | - | implicit | - | KERNEL32.dll |
| 3 (closesocket) | x | 0x80000003 | 0x80000003 | 0 (0x0000) | network | - | implicit | x | WS2_32.dll |
| 16 (recv) | x | 0x80000010 | 0x80000010 | 0 (0x0000) | network | - | implicit | x | WS2_32.dll |
| 19 (send) | x | 0x80000013 | 0x80000013 | 0 (0x0000) | network | - | implicit | x | WS2_32.dll |
| 8 (htonl) | x | 0x80000008 | 0x80000008 | 0 (0x0000) | network | - | implicit | x | WS2_32.dll |
| 14 (ntohl) | x | 0x8000000E | 0x8000000E | 0 (0x0000) | network | - | implicit | x | WS2_32.dll |
| 115 (WSAStartup) | x | 0x80000073 | 0x80000073 | 0 (0x0000) | network | - | implicit | x | WS2_32.dll |
| 12 (inet_ntoa) | x | 0x8000000C | 0x8000000C | 0 (0x0000) | network | - | implicit | x | WS2_32.dll |
| 10 (ioctlsocket) | x | 0x8000000A | 0x8000000A | 0 (0x0000) | network | - | implicit | x | WS2_32.dll |
| 18 (select) | x | 0x80000012 | 0x80000012 | 0 (0x0000) | network | - | implicit | x | WS2_32.dll |
| 9 (htons) | x | 0x80000009 | 0x80000009 | 0 (0x0000) | network | - | implicit | x | WS2_32.dll |
| 23 (socket) | x | 0x80000017 | 0x80000017 | 0 (0x0000) | network | - | implicit | x | WS2_32.dll |
| 4 (connect) | x | 0x80000004 | 0x80000004 | 0 (0x0000) | network | - | implicit | x | WS2_32.dll |
| 11 (inet_addr) | x | 0x8000000B | 0x8000000B | 0 (0x0000) | network | - | implicit | x | WS2_32.dll |
| GetAdaptersInfo | x | 0x0000A792 | 0x0000A792 | 28 (0x001C) | network | - | implicit | - | iphlpapi.dll |
| GetPerAdapterInfo | - | 0x0000A77E | 0x0000A77E | 64 (0x0040) | network | - | implicit | - | iphlpapi.dll |
| InternetOpenA | x | 0x0000A7DC | 0x0000A7DC | 146 (0x0092) | network | - | implicit | - | WININET.dll |
| InternetOpenUrlA | x | 0x0000A7C8 | 0x0000A7C8 | 147 (0x0093) | network | - | implicit | - | WININET.dll |
| InternetCloseHandle | x | 0x0000A7B2 | 0x0000A7B2 | 105 (0x0069) | network | - | implicit | - | WININET.dll |
| LocalFree | - | 0x0000A610 | 0x0000A610 | 604 (0x025C) | memory | - | implicit | - | KERNEL32.dll |
| LocalAlloc | - | 0x0000A61C | 0x0000A61C | 600 (0x0258) | memory | - | implicit | - | KERNEL32.dll |
| GlobalAlloc | - | 0x0000A464 | 0x0000A464 | 504 (0x01F8) | memory | - | implicit | - | KERNEL32.dll |
| GlobalFree | - | 0x0000A456 | 0x0000A456 | 511 (0x01FF) | memory | - | implicit | - | KERNEL32.dll |
| ReadFile | - | 0x0000A54E | 0x0000A54E | 693 (0x02B5) | file | - | implicit | - | KERNEL32.dll |
| GetFileSize | - | 0x0000A55A | 0x0000A55A | 355 (0x0163) | file | - | implicit | - | KERNEL32.dll |
| CreateFileA | - | 0x0000A568 | 0x0000A568 | 83 (0x0053) | file | - | implicit | - | KERNEL32.dll |
| MoveFileExA | x | 0x0000A576 | 0x0000A576 | 623 (0x026F) | file | Remote File Copy | implicit | - | KERNEL32.dll |
| GetCurrentThreadId | x | 0x0000A524 | 0x0000A524 | 326 (0x0146) | execution | Process Discovery | implicit | - | KERNEL32.dll |
| GetCurrentThread | x | 0x0000A53A | 0x0000A53A | 325 (0x0145) | execution | - | implicit | - | KERNEL32.dll |
| TerminateThread | - | 0x0000A4E4 | 0x0000A4E4 | 863 (0x035F) | execution | - | implicit | - | KERNEL32.dll |
| ExitProcess | - | 0x0000A5EC | 0x0000A5EC | 185 (0x00B9) | execution | - | implicit | - | KERNEL32.dll |
| Sleep | - | 0x0000A408 | 0x0000A408 | 854 (0x0356) | execution | Sandbox Evasion | implicit | - | KERNEL32.dll |
| _endthreadex | - | 0x0000A80A | 0x0000A80A | 197 (0x00C5) | execution | - | implicit | - | MSVCRT.dll |
| _beginthreadex | - | 0x0000A82C | 0x0000A82C | 166 (0x00A6) | execution | - | implicit | - | MSVCRT.dll |
| GetProcAddress | - | 0x0000A5C6 | 0x0000A5C6 | 416 (0x01A0) | dynamic-library | - | implicit | - | KERNEL32.dll |
| GetModuleHandleW | - | 0x0000A5D8 | 0x0000A5D8 | 386 (0x0182) | dynamic-library | - | implicit | - | KERNEL32.dll |
| GetModuleFileNameA | - | 0x0000A5FA | 0x0000A5FA | 381 (0x017D) | dynamic-library | - | implicit | - | KERNEL32.dll |
| GetModuleHandleA | - | 0x0000A966 | 0x0000A966 | 383 (0x017F) | dynamic-library | - | implicit | - | KERNEL32.dll |
| CryptGenRandom | x | 0x0000A650 | 0x0000A650 | 150 (0x0096) | cryptography | Obfuscated Files or I... | implicit | - | ADVAPI32.dll |
| CryptAcquireContextA | x | 0x0000A638 | 0x0000A638 | 133 (0x0085) | cryptography | Obfuscated Files or I... | implicit | - | ADVAPI32.dll |

The list of imported APIs reveals that the malware is capable of performing a range of tasks such as modifying the registry, creating directories, and executing other executables.

## Basic Dynamic Analysis

### Cutter

During the analysis of Cutter, the malware was observed to push various payloads or variables into the main argument and load "str.http:__www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com" into memory. The malware then calls InternetOpenA and attempts to connect to the DNS provided. If the DNS responds with 200 OK, the program terminates. However, if the DNS response is not successful, the malware executes the payload and proceeds with the encryption. These observations provide valuable insights into the behavior and functionality of malware.



```
[0x00408140]
 int main (int argc, char **argv, char **envp);
 ; var int32_t var_14h @ esp+0x28
 ; var int32_t var_8h @ esp+0x3c
 ; var int32_t var_41h @ esp+0x75
 ; var int32_t var_45h @ esp+0x79
 ; var int32_t var_49h @ esp+0x7d
 ; var int32_t var_4dh @ esp+0x81
 ; var int32_t var_51h @ esp+0x85
 ; var int32_t var_55h @ esp+0x89
 ; var int32_t var_6bh @ esp+0x8b
 sub     esp, 0x50
 push    esi
 push    edi
 mov     ecx, 0xe                    ; 14
 mov     esi, str.http:__www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com ; 0x4313d0
 lea     edi, [var_8h]
 xor     eax, eax
 rep     movsd dword es:[edi], dword ptr [esi]
 movsb   byte es:[edi], byte ptr [esi]
 mov     dword [var_41h], eax
 mov     dword [var_45h], eax
 mov     dword [var_49h], eax
 mov     dword [var_4dh], eax
 mov     dword [var_51h], eax
 mov     word [var_55h], ax
 push    eax
 push    eax
 push    eax
 push    1                           ; 1
 push    eax
 mov     byte [var_6bh], al
 call    dword [InternetOpenA]       ; 0x40a134
 push    0
 push    0x84000000
 push    0
 lea     ecx, [var_14h]
 mov     esi, eax
 push    0
 push    ecx
 push    esi
 call    dword [InternetOpenUrlA]    ; 0x40a138
 mov     edi, eax
 push    esi
 mov     esi, dword [InternetCloseHandle] ; 0x40a13c
 test    edi, edi
 jne     0x4081bc
```

```
[0x004081a7]
 call    esi
 push    0
 call    esi
 call    fcn.00408090
 pop     edi
 xor     eax, eax
 pop     esi
 add     esp, 0x50
 ret     0x10
```

```
[0x004081bc]
 call    esi
 push    edi
 call    esi
 pop     edi
 xor     eax, eax
 pop     esi
 add     esp, 0x50
 ret     0x10
```

Refer to the image for more detailed information

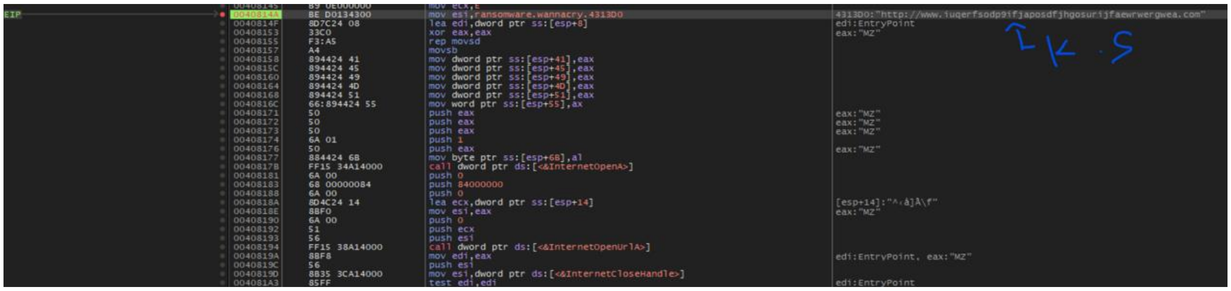
# ADVANCED ANALYSIS

## Static & Dynamic Analysis

### Xdb32

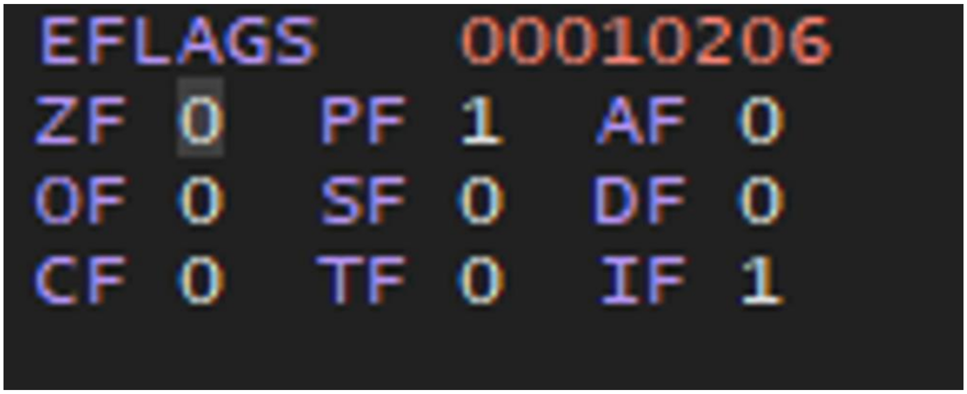
During the use of XDB32 debugger:

The memory address 00409B4A was identified as a critical component of the malware's payload execution, as a process was created before it was called. This indicates that the last part of the memory is the trigger point for the payload, which ultimately activates the ransomware and initiates the encryption process.



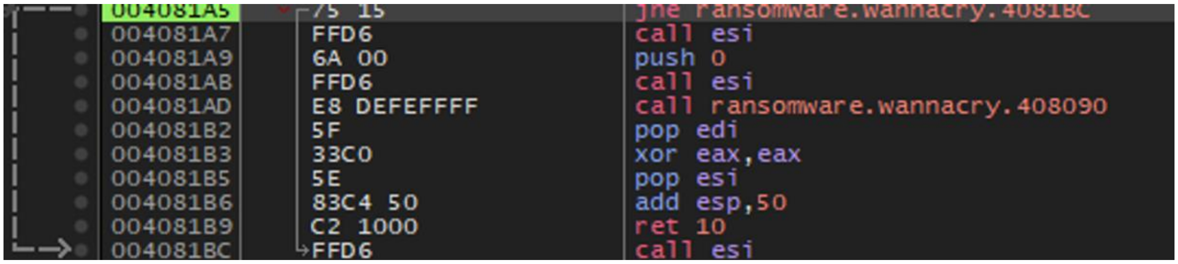
```
     00408145   B9 0E000000      mov ecx,E
EIP  0040814A   BE D0134300      mov esi,ransomware.wannacry.4313D0            4313D0:"http://www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com"
     0040814F   8D7C24 08        lea edi,dword ptr ss:[esp+8]                  edi:EntryPoint
     00408153   33C0             xor eax,eax                                   eax:"MZ"
     00408155   F3:A5            rep movsd
     00408157   A4               movsb
     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                                      eax:"MZ"
     00408172   50               push eax                                      eax:"MZ"
     00408173   50               push eax                                      eax:"MZ"
     00408174   6A 01            push 1
     00408176   50               push eax                                      eax:"MZ"
     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]                 [esp+14]:"^‹åJÅ\f"
     0040818E   8BF0             mov esi,eax                                   eax:"MZ"
     00408190   6A 00            push 0
     00408192   51               push ecx
     00408193   56               push esi
     00408194   FF15 38A14000    call dword ptr ds:[<&InternetOpenUrlA>]
     0040819A   8BF8             mov edi,eax                                   edi:EntryPoint, eax:"MZ"
     0040819C   56               push esi
     0040819D   8B35 3CA14000    mov esi,dword ptr ds:[<&InternetCloseHandle>]
     004081A3   85FF             test edi,edi                                  edi:EntryPoint
```

During the Basic Dynamic Analysis, it was identified that the malware would execute if the DNS did not respond. This particular section of the memory, as shown in the attached image below, indicates that if the flag is set to 0, it will skip this part of the code. However, if the flag is set to 1, it will trigger the payload and continue with the execution of the malware.



```
EFLAGS    00010206
ZF  0   PF  1   AF  0
OF  0   SF  0   DF  0
CF  0   TF  0   IF  1
```

As presented if we change this ZF flag to 1, the program below will continue to execute.



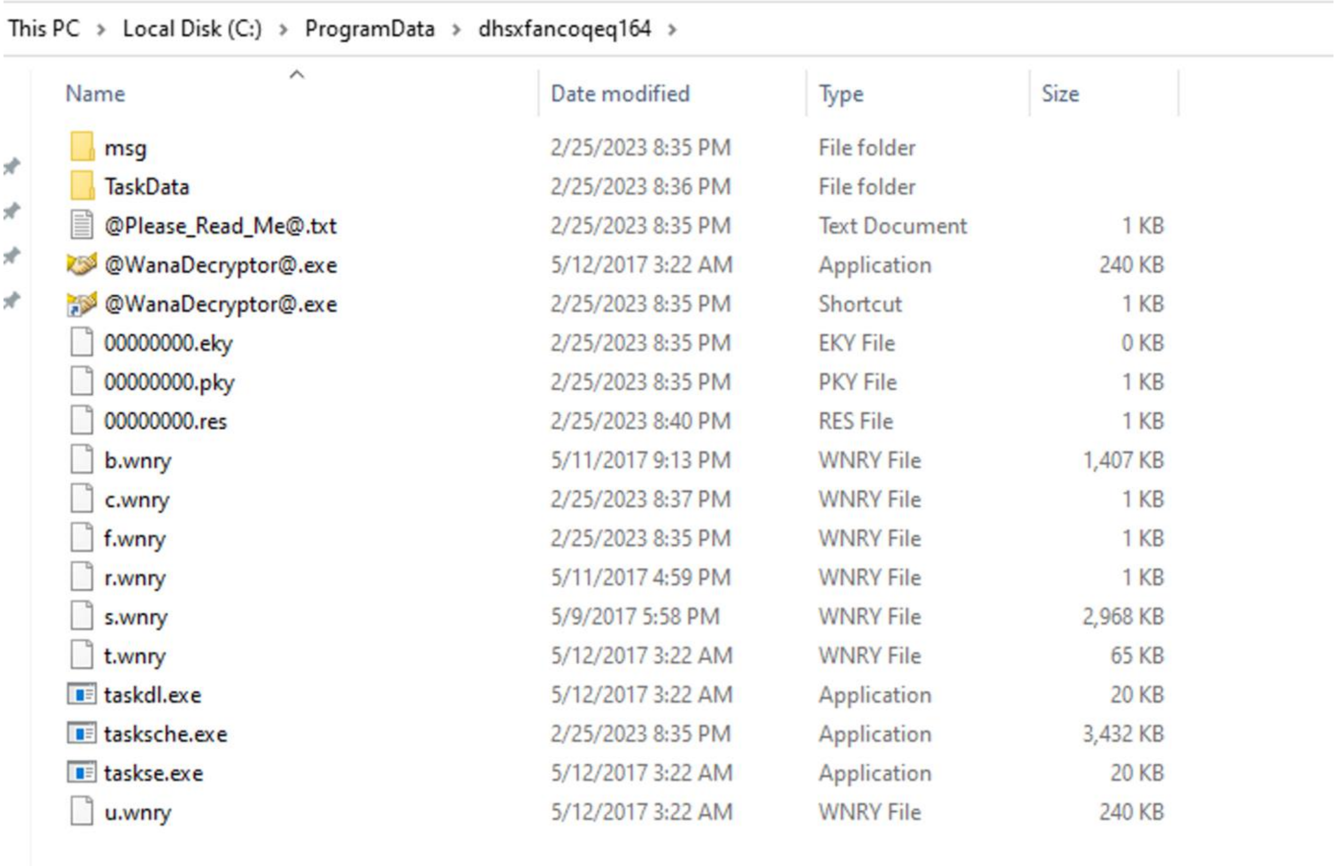
```
004081A5    75 15          jne ransomware.wannacry.4081BC
004081A7    FFD6           call esi
004081A9    6A 00          push 0
004081AB    FFD6           call esi
004081AD    E8 DEFEFFFF    call ransomware.wannacry.408090
004081B2    5F             pop edi
004081B3    33C0           xor eax,eax
004081B5    5E             pop esi
004081B6    83C4 50        add esp,50
004081B9    C2 1000        ret 10
004081BC    FFD6           call esi
```

To gain further insight into the advanced analysis, we can examine the indicators of compromise that reveal the actions taken by the malware once it was triggered.

# INDICATORS OF COMPROMISE

## Host Based Analysis

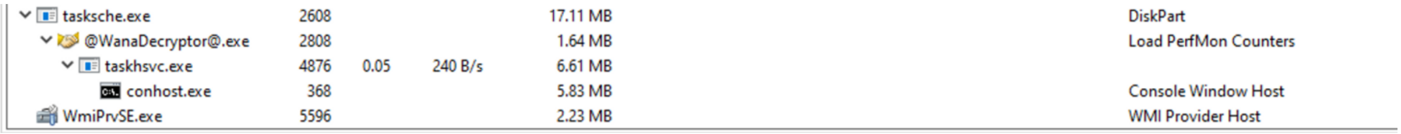
As seen in the directory it is identified that this directory was the staging directory of the executed malware.



This PC › Local Disk (C:) › ProgramData › dhsxfancoqeq164 ›

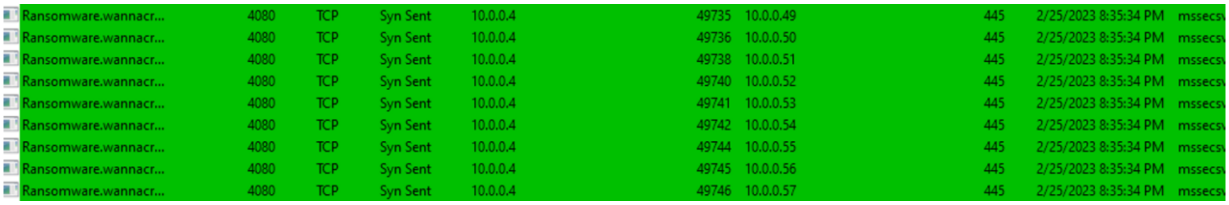
| Name | Date modified | Type | Size |
|---|---|---|---|
| msg | 2/25/2023 8:35 PM | File folder | |
| TaskData | 2/25/2023 8:36 PM | File folder | |
| @Please_Read_Me@.txt | 2/25/2023 8:35 PM | Text Document | 1 KB |
| @WanaDecryptor@.exe | 5/12/2017 3:22 AM | Application | 240 KB |
| @WanaDecryptor@.exe | 2/25/2023 8:35 PM | Shortcut | 1 KB |
| 00000000.eky | 2/25/2023 8:35 PM | EKY File | 0 KB |
| 00000000.pky | 2/25/2023 8:35 PM | PKY File | 1 KB |
| 00000000.res | 2/25/2023 8:40 PM | RES File | 1 KB |
| b.wnry | 5/11/2017 9:13 PM | WNRY File | 1,407 KB |
| c.wnry | 2/25/2023 8:37 PM | WNRY File | 1 KB |
| f.wnry | 2/25/2023 8:35 PM | WNRY File | 1 KB |
| r.wnry | 5/11/2017 4:59 PM | WNRY File | 1 KB |
| s.wnry | 5/9/2017 5:58 PM | WNRY File | 2,968 KB |
| t.wnry | 5/12/2017 3:22 AM | WNRY File | 65 KB |
| taskdl.exe | 5/12/2017 3:22 AM | Application | 20 KB |
| tasksche.exe | 2/25/2023 8:35 PM | Application | 3,432 KB |
| taskse.exe | 5/12/2017 3:22 AM | Application | 20 KB |
| u.wnry | 5/12/2017 3:22 AM | WNRY File | 240 KB |

During the malware execution, it was observed that a new file called taskhsvc.exe was created under the taskche.exe. This process was found to be the persistence mechanism of malware.



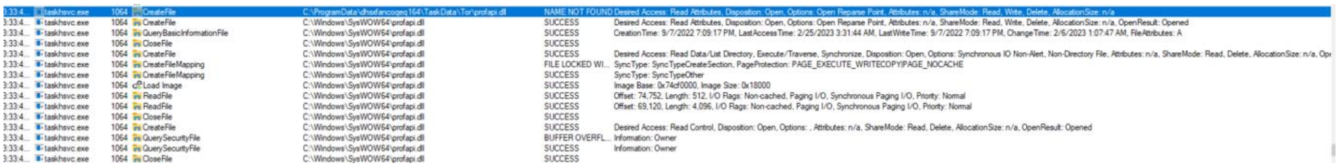
```
taksche.exe               2608                      17.11 MB    DiskPart
  @WanaDecryptor@.exe     2808                       1.64 MB    Load PerfMon Counters
    taskhsvc.exe          4876   0.05   240 B/s      6.61 MB
      conhost.exe          368                       5.83 MB    Console Window Host
WmiPrvSE.exe              5596                       2.23 MB    WMI Provider Host
```

During these processes being active the malware constantly tries to call other hosts within the network using the SMBv1



```
Ransomware.wannacr...  4080  TCP  Syn Sent  10.0.0.4  49735  10.0.0.49  445  2/25/2023 8:35:34 PM  mssecsv
Ransomware.wannacr...  4080  TCP  Syn Sent  10.0.0.4  49736  10.0.0.50  445  2/25/2023 8:35:34 PM  mssecsv
Ransomware.wannacr...  4080  TCP  Syn Sent  10.0.0.4  49738  10.0.0.51  445  2/25/2023 8:35:34 PM  mssecsv
Ransomware.wannacr...  4080  TCP  Syn Sent  10.0.0.4  49740  10.0.0.52  445  2/25/2023 8:35:34 PM  mssecsv
Ransomware.wannacr...  4080  TCP  Syn Sent  10.0.0.4  49741  10.0.0.53  445  2/25/2023 8:35:34 PM  mssecsv
Ransomware.wannacr...  4080  TCP  Syn Sent  10.0.0.4  49742  10.0.0.54  445  2/25/2023 8:35:34 PM  mssecsv
Ransomware.wannacr...  4080  TCP  Syn Sent  10.0.0.4  49744  10.0.0.55  445  2/25/2023 8:35:34 PM  mssecsv
Ransomware.wannacr...  4080  TCP  Syn Sent  10.0.0.4  49745  10.0.0.56  445  2/25/2023 8:35:34 PM  mssecsv
Ransomware.wannacr...  4080  TCP  Syn Sent  10.0.0.4  49746  10.0.0.57  445  2/25/2023 8:35:34 PM  mssecsv
```

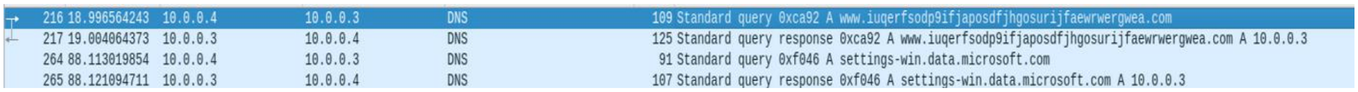
The malware's payload execution is clearly indicated by the many Create File and other operations observed in Procmon. By filtering the captured events, we can see the specific files that were created during a particular stage of the malware's execution, as shown in the image below.



```
3:33:4...  taskhsvc.exe  1064  CreateFile                   C:\ProgramData\dhsxfancoqeq164\TaskData\Tor\profapi.dll  NAME NOT FOUND  Desired Access: Read Attributes, Disposition: Open, Options: Open Reparse Point, Attributes: n/a, ShareMode: Read, Write, Delete, AllocationSize: n/a
3:33:4...  taskhsvc.exe  1064  CreateFile                   C:\Windows\SysWOW64\profapi.dll  SUCCESS          Desired Access: Read Attributes, Disposition: Open, Options: Open Reparse Point, Attributes: n/a, ShareMode: Read, Write, Delete, AllocationSize: n/a, OpenResult: Opened
3:33:4...  taskhsvc.exe  1064  QueryBasicInformationFile    C:\Windows\SysWOW64\profapi.dll  SUCCESS          CreationTime: 9/7/2022 7:09:17 PM, LastAccessTime: 2/25/2023 3:31:44 AM, LastWriteTime: 9/7/2022 7:09:17 PM, ChangeTime: 2/6/2023 1:07:47 AM, FileAttributes: A
3:33:4...  taskhsvc.exe  1064  CloseFile                    C:\Windows\SysWOW64\profapi.dll  SUCCESS
3:33:4...  taskhsvc.exe  1064  CreateFile                   C:\Windows\SysWOW64\profapi.dll  SUCCESS          Desired Access: Read Data/List Directory, Execute/Traverse, Synchronize, Disposition: Open, Options: Synchronous IO Non-Alert, Non-Directory File, Attributes: n/a, ShareMode: Read, Delete, AllocationSize: n/a, Ope
3:33:4...  taskhsvc.exe  1064  CreateFileMapping            C:\Windows\SysWOW64\profapi.dll  FILE LOCKED WI...  SyncType: SyncTypeCreateSection, PageProtection: PAGE_EXECUTE_WRITECOPY|PAGE_NOCACHE
3:33:4...  taskhsvc.exe  1064  CreateFileMapping            C:\Windows\SysWOW64\profapi.dll  SUCCESS          SyncType: SyncTypeOther
3:33:4...  taskhsvc.exe  1064  Load Image                   C:\Windows\SysWOW64\profapi.dll  SUCCESS          Image Base: 0x74cf0000, Image Size: 0x18000
3:33:4...  taskhsvc.exe  1064  ReadFile                     C:\Windows\SysWOW64\profapi.dll  SUCCESS          Offset: 74,752, Length: 512, I/O Flags: Non-cached, Paging I/O, Synchronous Paging I/O, Priority: Normal
3:33:4...  taskhsvc.exe  1064  ReadFile                     C:\Windows\SysWOW64\profapi.dll  SUCCESS          Offset: 69,120, Length: 4,096, I/O Flags: Non-cached, Paging I/O, Synchronous Paging I/O, Priority: Normal
3:33:4...  taskhsvc.exe  1064  CloseFile                    C:\Windows\SysWOW64\profapi.dll  SUCCESS
3:33:4...  taskhsvc.exe  1064  CreateFile                   C:\Windows\SysWOW64\profapi.dll  SUCCESS          Desired Access: Read Control, Disposition: Open, Options: , Attributes: n/a, ShareMode: Read, Delete, AllocationSize: n/a, OpenResult: Opened
3:33:4...  taskhsvc.exe  1064  QuerySecurityFile            C:\Windows\SysWOW64\profapi.dll  BUFFER OVERFL...  Information: Owner
3:33:4...  taskhsvc.exe  1064  QuerySecurityFile            C:\Windows\SysWOW64\profapi.dll  SUCCESS          Information: Owner
3:33:4...  taskhsvc.exe  1064  CloseFile                    C:\Windows\SysWOW64\profapi.dll  SUCCESS
```

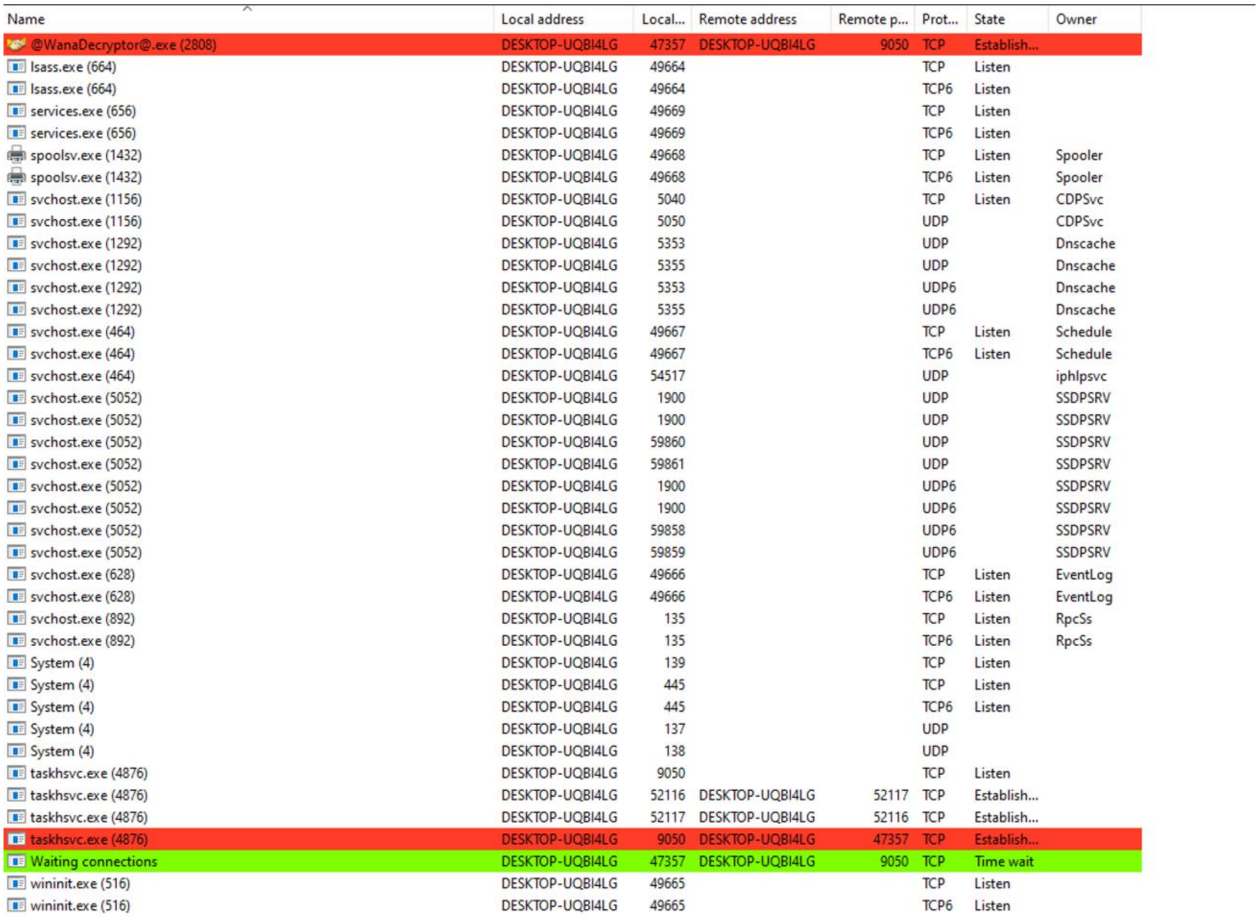
## Network Indicators

One of the few indicators that the malware is attempting to call for the DNS attempting to check if it will return 200 OK.



```
216 18.996564243  10.0.0.4   10.0.0.3   DNS   109 Standard query 0xca92 A www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com
217 19.004064373  10.0.0.3   10.0.0.4   DNS   125 Standard query response 0xca92 A www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com A 10.0.0.3
264 88.113019854  10.0.0.4   10.0.0.3   DNS    91 Standard query 0xf046 A settings-win.data.microsoft.com
265 88.121094711  10.0.0.3   10.0.0.4   DNS   107 Standard query response 0xf046 A settings-win.data.microsoft.com A 10.0.0.3
```

During the execution of the malware, it creates multiple processes on the victim's machine, such as changes in the Registry, new payloads, and commands. The malware also performs various actions on TCP ports, including opening and closing connections. Multiple calls are made to different ports, ranging from SMBv1 to nodes that communicate with the malware creators command and control node.



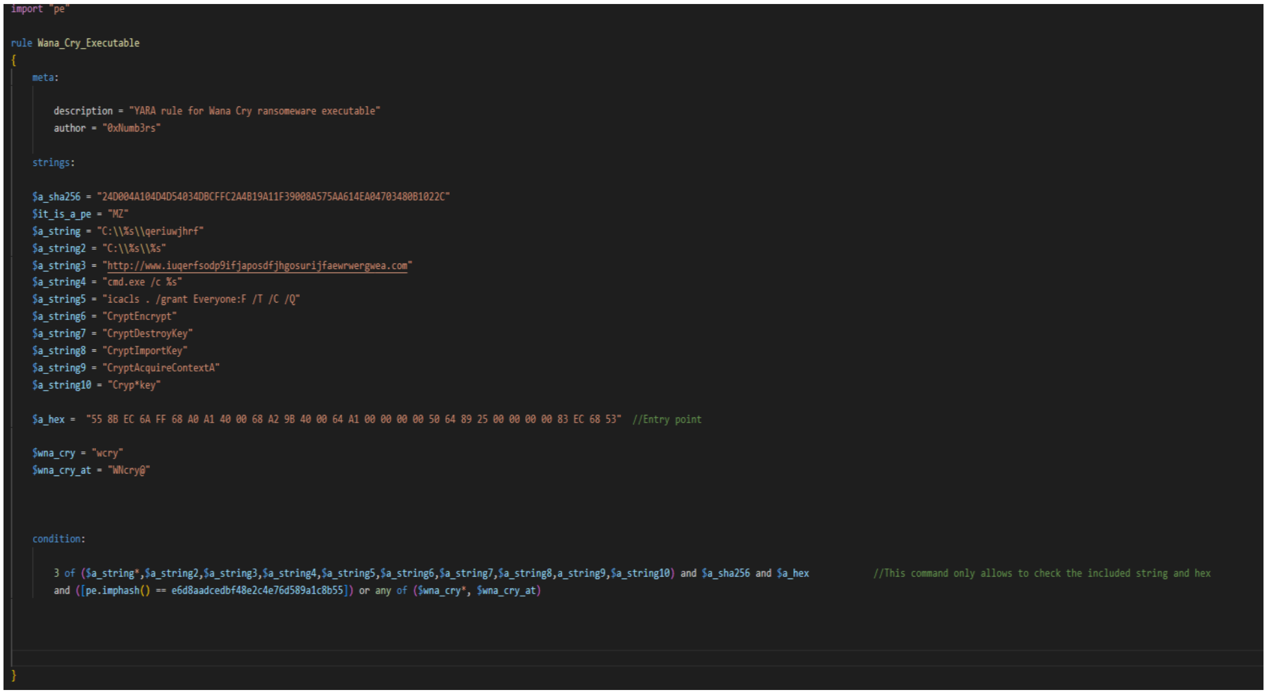
| Name | Local address | Local... | Remote address | Remote p... | Prot... | State | Owner |
|---|---|---|---|---|---|---|---|
| @WanaDecryptor@.exe (2808) | DESKTOP-UQBI4LG | 47357 | DESKTOP-UQBI4LG | 9050 | TCP | Establish... | |
| lsass.exe (664) | DESKTOP-UQBI4LG | 49664 | | | TCP | Listen | |
| lsass.exe (664) | DESKTOP-UQBI4LG | 49664 | | | TCP6 | Listen | |
| services.exe (656) | DESKTOP-UQBI4LG | 49669 | | | TCP | Listen | |
| services.exe (656) | DESKTOP-UQBI4LG | 49669 | | | TCP6 | Listen | |
| spoolsv.exe (1432) | DESKTOP-UQBI4LG | 49668 | | | TCP | Listen | Spooler |
| spoolsv.exe (1432) | DESKTOP-UQBI4LG | 49668 | | | TCP6 | Listen | Spooler |
| svchost.exe (1156) | DESKTOP-UQBI4LG | 5040 | | | TCP | Listen | CDPSvc |
| svchost.exe (1156) | DESKTOP-UQBI4LG | 5050 | | | UDP | | CDPSvc |
| svchost.exe (1292) | DESKTOP-UQBI4LG | 5353 | | | UDP | | Dnscache |
| svchost.exe (1292) | DESKTOP-UQBI4LG | 5355 | | | UDP | | Dnscache |
| svchost.exe (1292) | DESKTOP-UQBI4LG | 5353 | | | UDP6 | | Dnscache |
| svchost.exe (1292) | DESKTOP-UQBI4LG | 5355 | | | UDP6 | | Dnscache |
| svchost.exe (464) | DESKTOP-UQBI4LG | 49667 | | | TCP | Listen | Schedule |
| svchost.exe (464) | DESKTOP-UQBI4LG | 49667 | | | TCP6 | Listen | Schedule |
| svchost.exe (464) | DESKTOP-UQBI4LG | 54517 | | | UDP | | iphlpsvc |
| svchost.exe (5052) | DESKTOP-UQBI4LG | 1900 | | | UDP | | SSDPSRV |
| svchost.exe (5052) | DESKTOP-UQBI4LG | 1900 | | | UDP | | SSDPSRV |
| svchost.exe (5052) | DESKTOP-UQBI4LG | 59860 | | | UDP | | SSDPSRV |
| svchost.exe (5052) | DESKTOP-UQBI4LG | 59861 | | | UDP | | SSDPSRV |
| svchost.exe (5052) | DESKTOP-UQBI4LG | 1900 | | | UDP6 | | SSDPSRV |
| svchost.exe (5052) | DESKTOP-UQBI4LG | 1900 | | | UDP6 | | SSDPSRV |
| svchost.exe (5052) | DESKTOP-UQBI4LG | 59858 | | | UDP6 | | SSDPSRV |
| svchost.exe (5052) | DESKTOP-UQBI4LG | 59859 | | | UDP6 | | SSDPSRV |
| svchost.exe (628) | DESKTOP-UQBI4LG | 49666 | | | TCP | Listen | EventLog |
| svchost.exe (628) | DESKTOP-UQBI4LG | 49666 | | | TCP6 | Listen | EventLog |
| svchost.exe (892) | DESKTOP-UQBI4LG | 135 | | | TCP | Listen | RpcSs |
| svchost.exe (892) | DESKTOP-UQBI4LG | 135 | | | TCP6 | Listen | RpcSs |
| System (4) | DESKTOP-UQBI4LG | 139 | | | TCP | Listen | |
| System (4) | DESKTOP-UQBI4LG | 445 | | | TCP | Listen | |
| System (4) | DESKTOP-UQBI4LG | 445 | | | TCP6 | Listen | |
| System (4) | DESKTOP-UQBI4LG | 137 | | | UDP | | |
| System (4) | DESKTOP-UQBI4LG | 138 | | | UDP | | |
| taskhsvc.exe (4876) | DESKTOP-UQBI4LG | 9050 | | | TCP | Listen | |
| taskhsvc.exe (4876) | DESKTOP-UQBI4LG | 52116 | DESKTOP-UQBI4LG | 52117 | TCP | Establish... | |
| taskhsvc.exe (4876) | DESKTOP-UQBI4LG | 52117 | DESKTOP-UQBI4LG | 52116 | TCP | Establish... | |
| taskhsvc.exe (4876) | DESKTOP-UQBI4LG | 9050 | DESKTOP-UQBI4LG | 47357 | TCP | Establish... | |
| Waiting connections | DESKTOP-UQBI4LG | 47357 | DESKTOP-UQBI4LG | 9050 | TCP | Time wait | |
| wininit.exe (516) | DESKTOP-UQBI4LG | 49665 | | | TCP | Listen | |
| wininit.exe (516) | DESKTOP-UQBI4LG | 49665 | | | TCP6 | Listen | |

# Rules & Signatures

A full set of YARA rules is included in Appendix A.

# Appendices

## A. Yara Rules



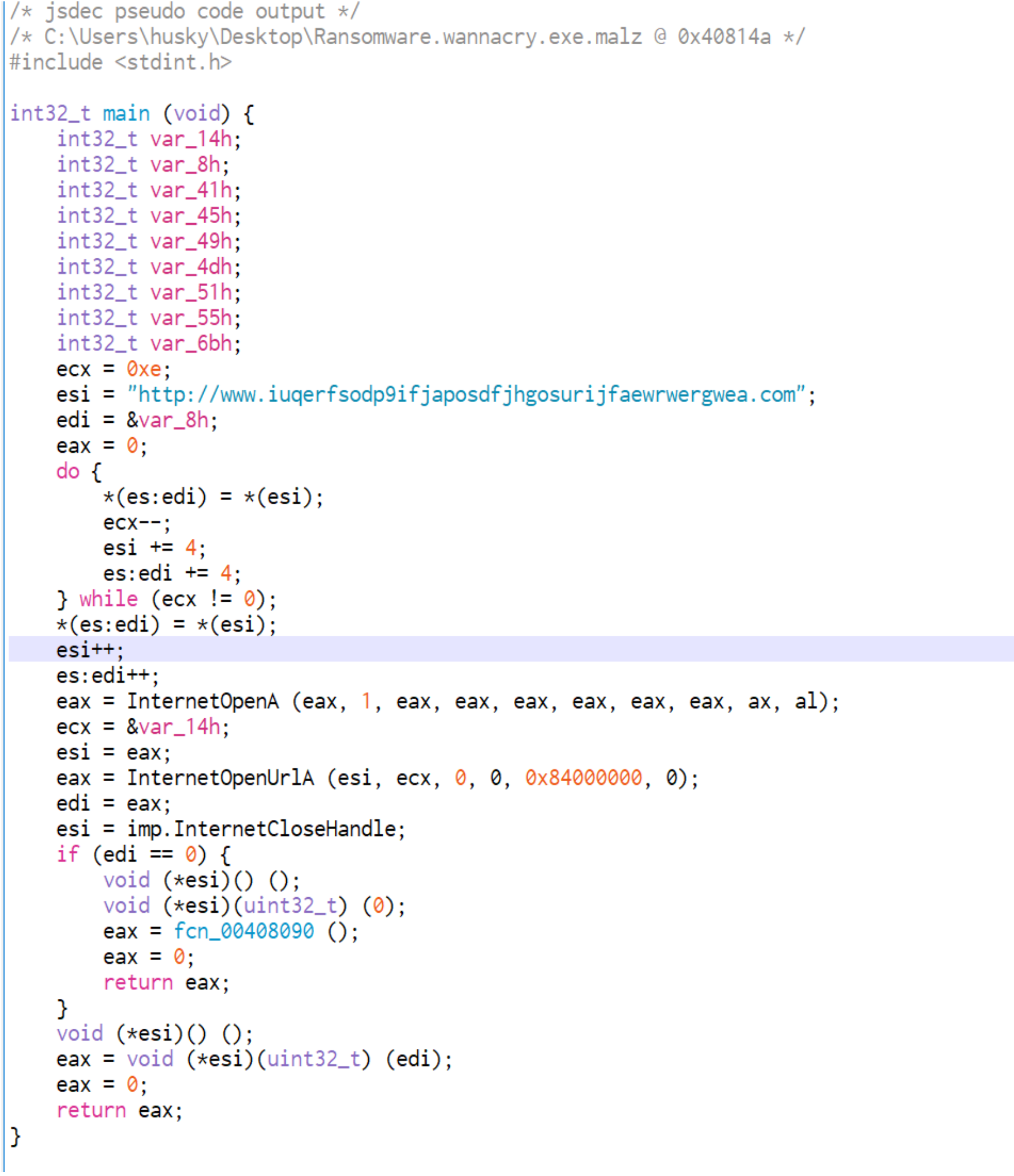
```
import "pe"

rule Wana_Cry_Executable
{
    meta:

        description = "YARA rule for Wana Cry ransomeware executable"
        author = "0xNumb3rs"

    strings:

    $a_sha256 = "24D004A104D4D54034DBCFFC2A4B19A11F39008A575AA614EA04703480B1022C"
    $it_is_a_pe = "MZ"
    $a_string = "C:\\%s\\qeriuwjhrf"
    $a_string2 = "C:\\%s\\%s"
    $a_string3 = "http://www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com"
    $a_string4 = "cmd.exe /c %s"
    $a_string5 = "icacls . /grant Everyone:F /T /C /Q"
    $a_string6 = "CryptEncrypt"
    $a_string7 = "CryptDestroyKey"
    $a_string8 = "CryptImportKey"
    $a_string9 = "CryptAcquireContextA"
    $a_string10 = "Cryp*key"

    $a_hex =  "55 8B EC 6A FF 68 A0 A1 40 00 68 A2 9B 40 00 64 A1 00 00 00 00 50 64 89 25 00 00 00 00 83 EC 68 53"  //Entry point

    $wna_cry = "wcry"
    $wna_cry_at = "WNcry@"



    condition:

        3 of ($a_string*,$a_string2,$a_string3,$a_string4,$a_string5,$a_string6,$a_string7,$a_string8,a_string9,$a_string10) and $a_sha256 and $a_hex          //This command only allows to check the included string and hex
        and ([pe.imphash() == e6d8aadcedbf48e2c4e76d589a1c8b55]) or any of ($wna_cry*, $wna_cry_at)



}
```

[Check The GitHub Link]

## B. Callback URLs

| Domain | Port |
|---|---|
| http://www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com | DNS Call |

## C. Decompiled Code Snippets



```
/* jsdec pseudo code output */
/* C:\Users\husky\Desktop\Ransomware.wannacry.exe.malz @ 0x40814a */
#include <stdint.h>

int32_t main (void) {
    int32_t var_14h;
    int32_t var_8h;
    int32_t var_41h;
    int32_t var_45h;
    int32_t var_49h;
    int32_t var_4dh;
    int32_t var_51h;
    int32_t var_55h;
    int32_t var_6bh;
    ecx = 0xe;
    esi = "http://www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com";
    edi = &var_8h;
    eax = 0;
    do {
        *(es:edi) = *(esi);
        ecx--;
        esi += 4;
        es:edi += 4;
    } while (ecx != 0);
    *(es:edi) = *(esi);
    esi++;
    es:edi++;
    eax = InternetOpenA (eax, 1, eax, eax, eax, eax, eax, eax, ax, al);
    ecx = &var_14h;
    esi = eax;
    eax = InternetOpenUrlA (esi, ecx, 0, 0, 0x84000000, 0);
    edi = eax;
    esi = imp.InternetCloseHandle;
    if (edi == 0) {
        void (*esi)() ();
        void (*esi)(uint32_t) (0);
        eax = fcn_00408090 ();
        eax = 0;
        return eax;
    }
    void (*esi)() ();
    eax = void (*esi)(uint32_t) (edi);
    eax = 0;
    return eax;
}
```

*Fig 5: Process of Main Routine in Cutter*