Practical Malware Analysis & Triage

Malware Analysis Report

Wannacry Ransomware

August 2023 | KillSwitch | v1.0

Table of Contents

Executive Summary

SHA256 hash	6C382A1C16DBA41B4FF6F0D728E9E92AEBFE2EE0C7FEB30A0E63B15EAF6C4B44

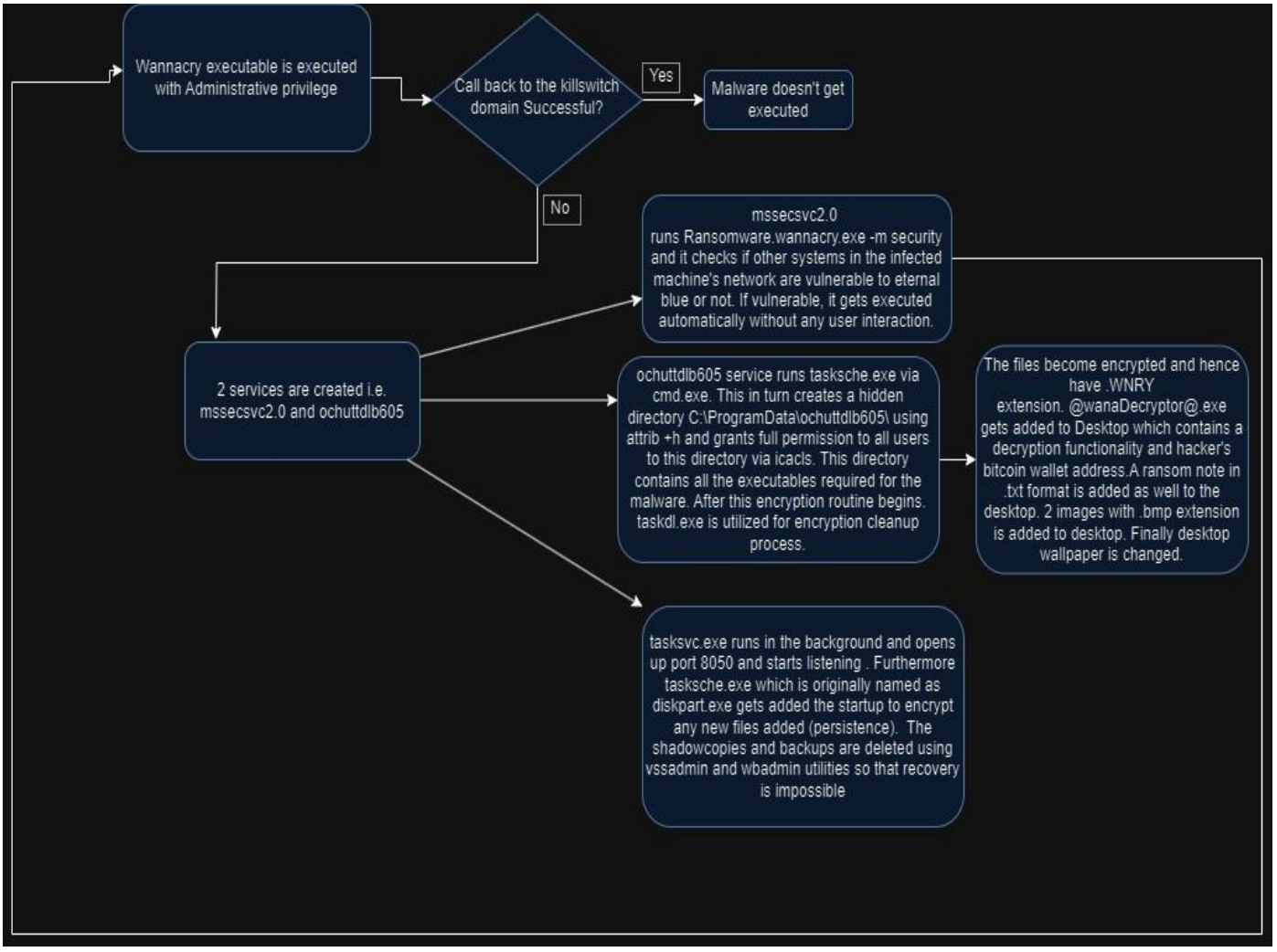
WannaCry, also known as WannaCryptor is a notorious malware that exhibits dual functionality, acting as both ransomware and a worm. It gained worldwide attention in 2017 due to its rapid propagation and impact on organizations and individuals. It is C++ compiled and runs on both x86 and x64 Windows operating 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
Ransomeware.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

Thie 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)

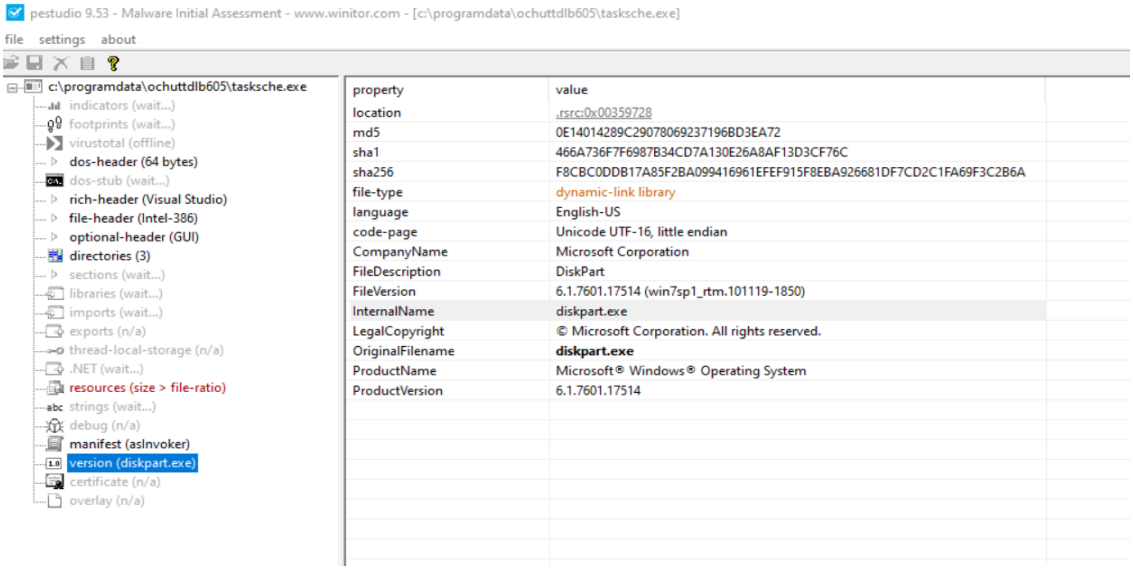


Figure 1: tasksche.exe is originally called diskpart.exe (PEstudio)

Other Files

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

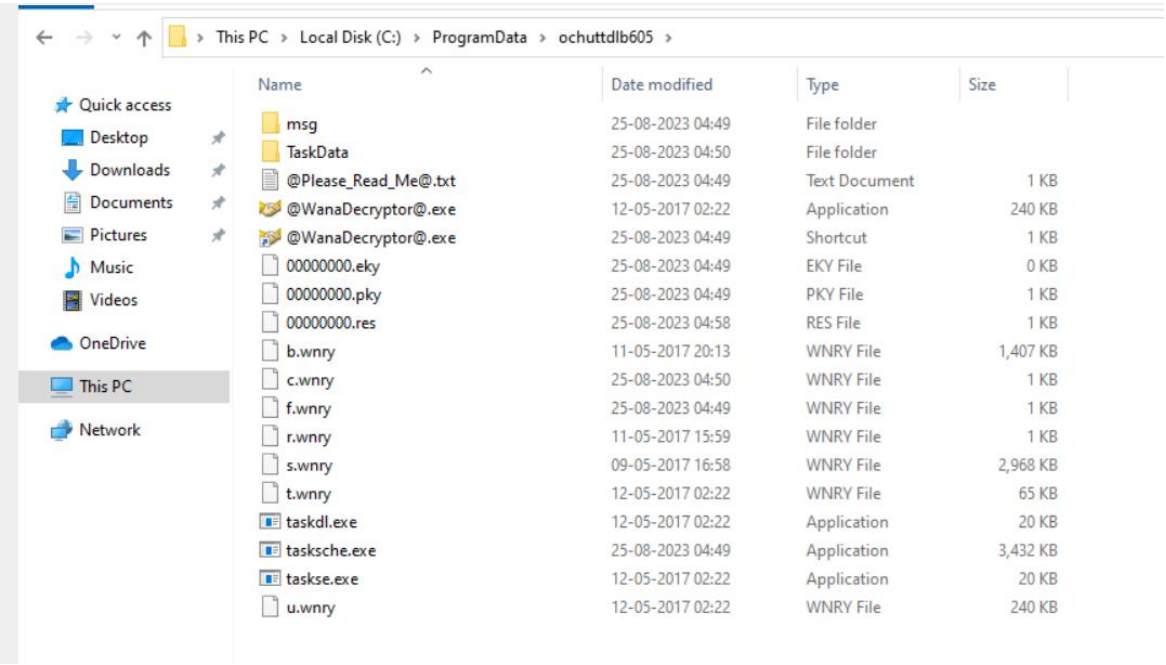


Figure 2: C:\ProgramData\ochuttdlb605

(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:

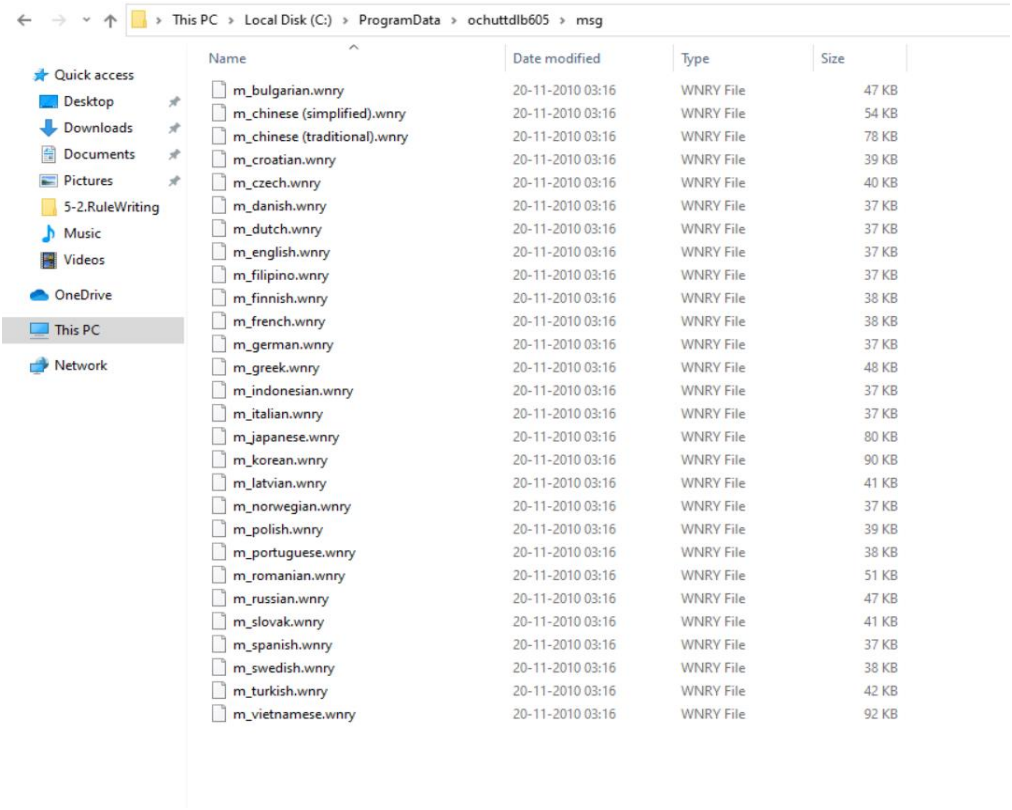


Figure 3: msg folder

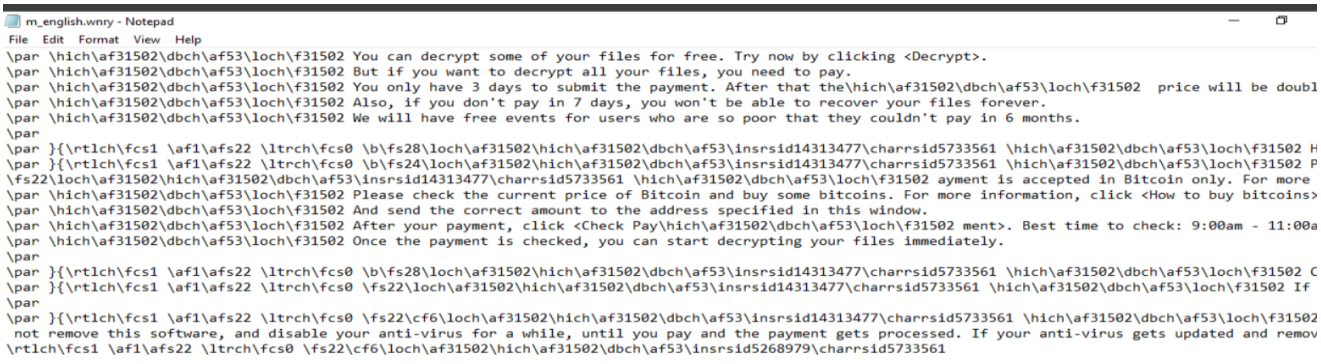


Figure 4: I confirmed that its ransom note message by reading the one in English

(iii) taskdl.exe:

- taskdl.exe is used for cleanup process.

- API like DeleteFileW is used for the cleanup process:

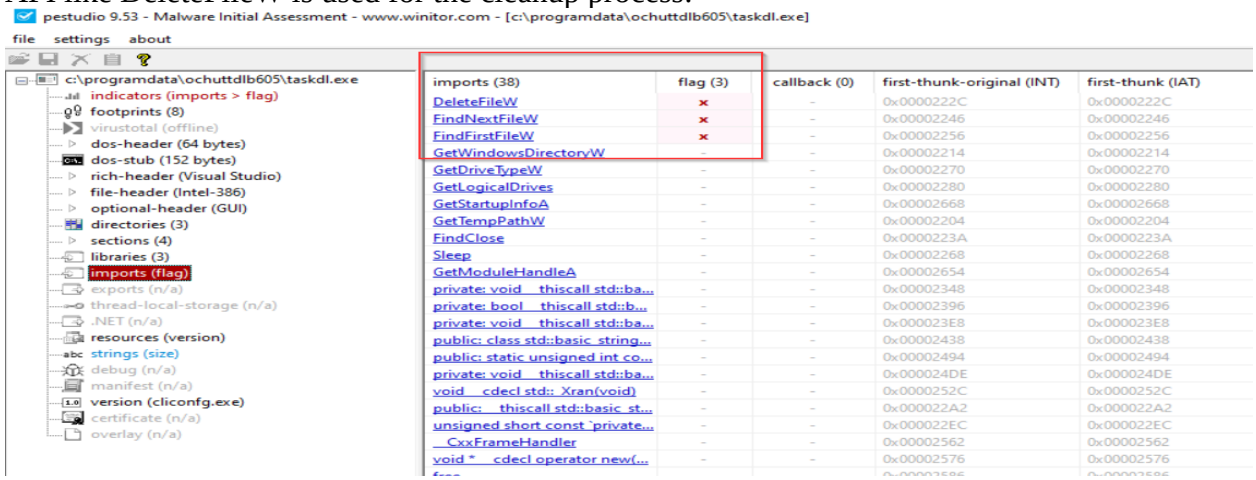


Figure 5: Analysis of taskdl.exe using PEstudio

(iv) taskse.exe:

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

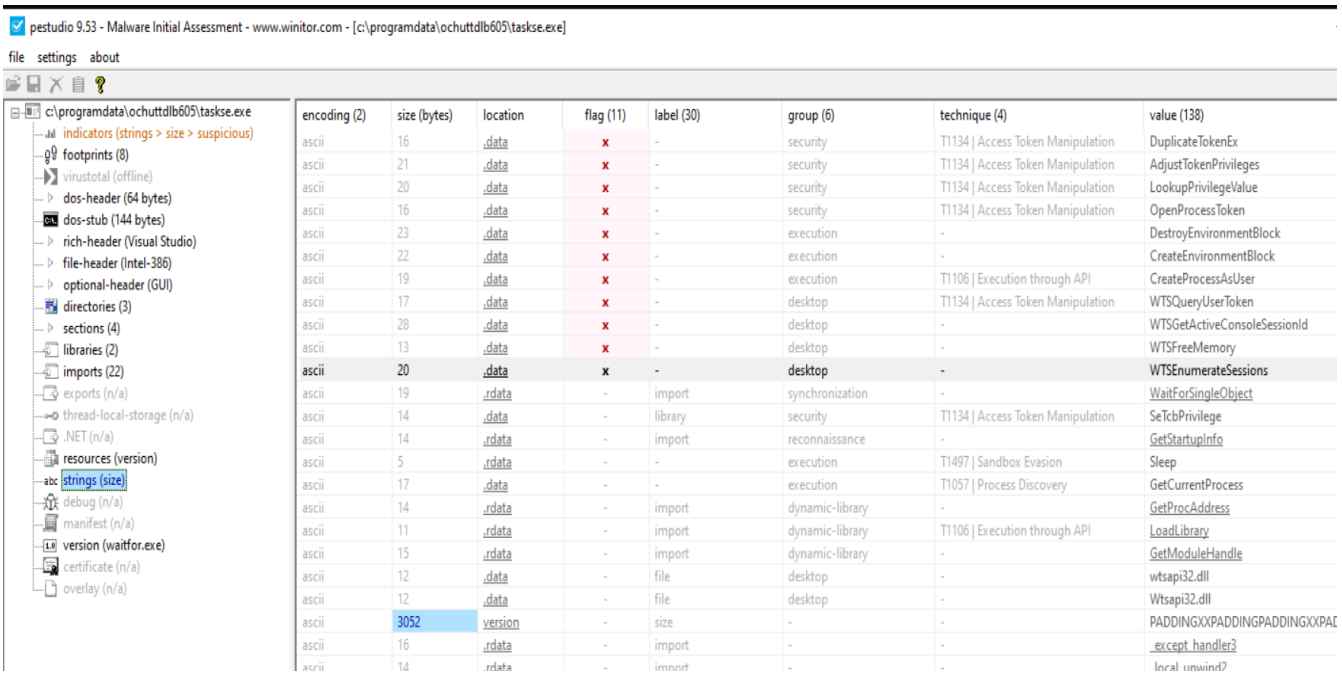


Figure 6: Analysis of taskse.exe using PEstudio

(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.)

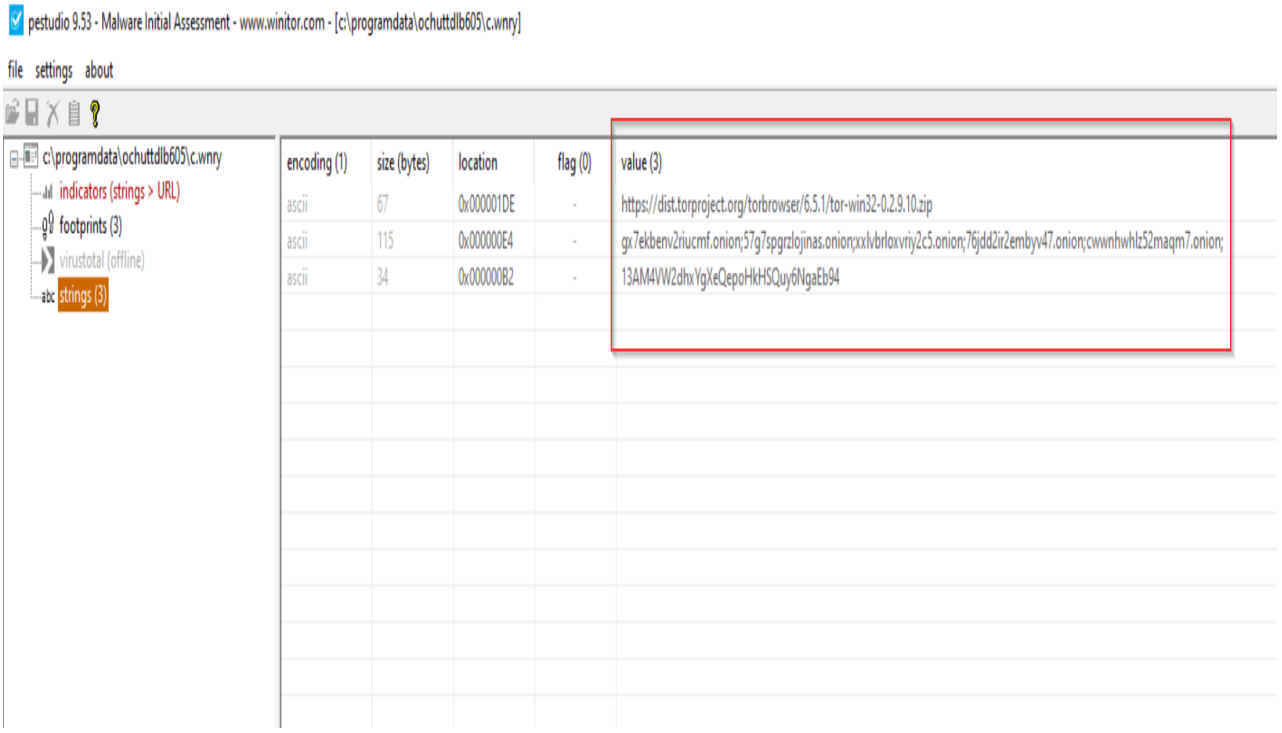


Figure 7: Analysis of c.wnry using PEstudio

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)

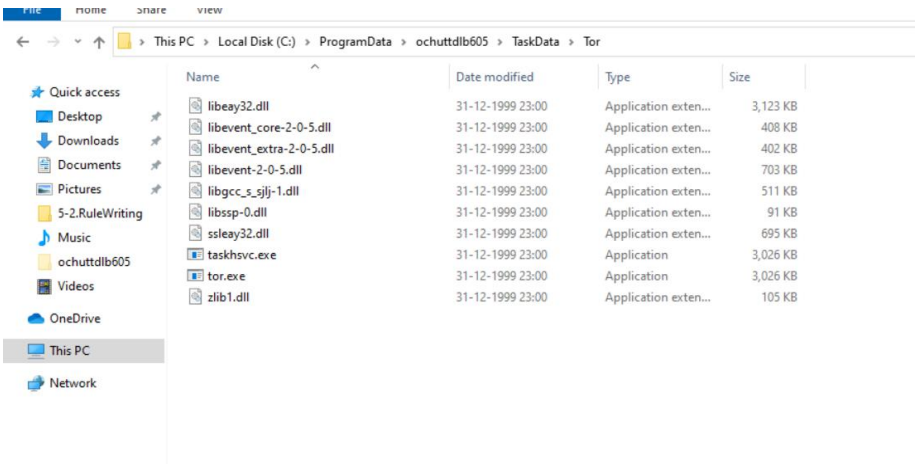


Figure 8: Tor folder

- 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

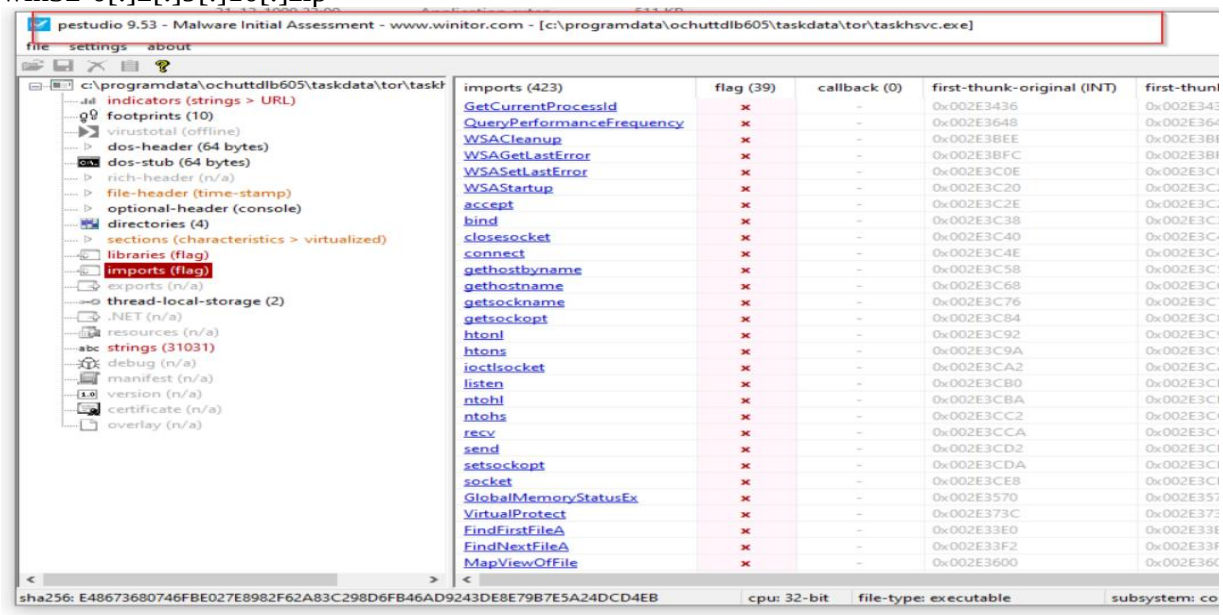


Figure 9: Both taskhsvc.exe and tor.exe have same API's and file hash confirming our theory that both are same files (PEstudio)

(vii) b.wnry:

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

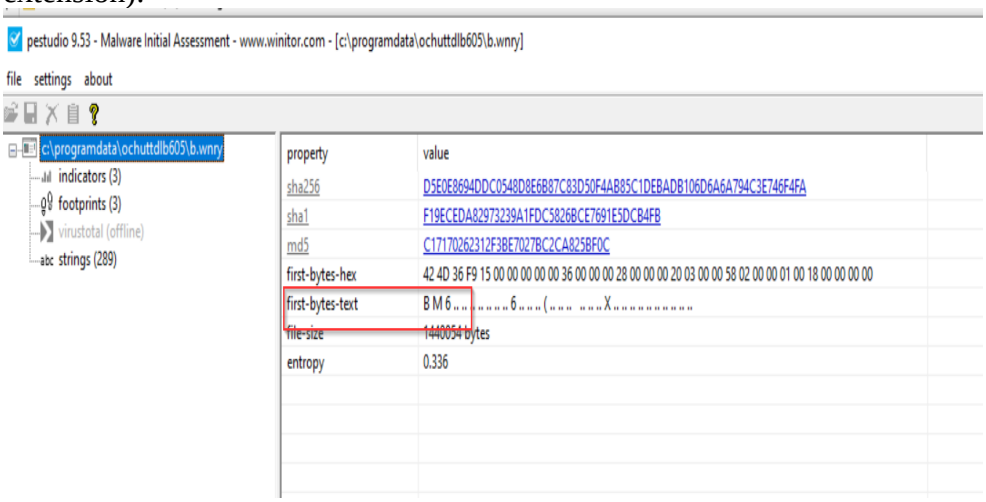


Figure 10: b.wnry file signature (PEstudio)

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

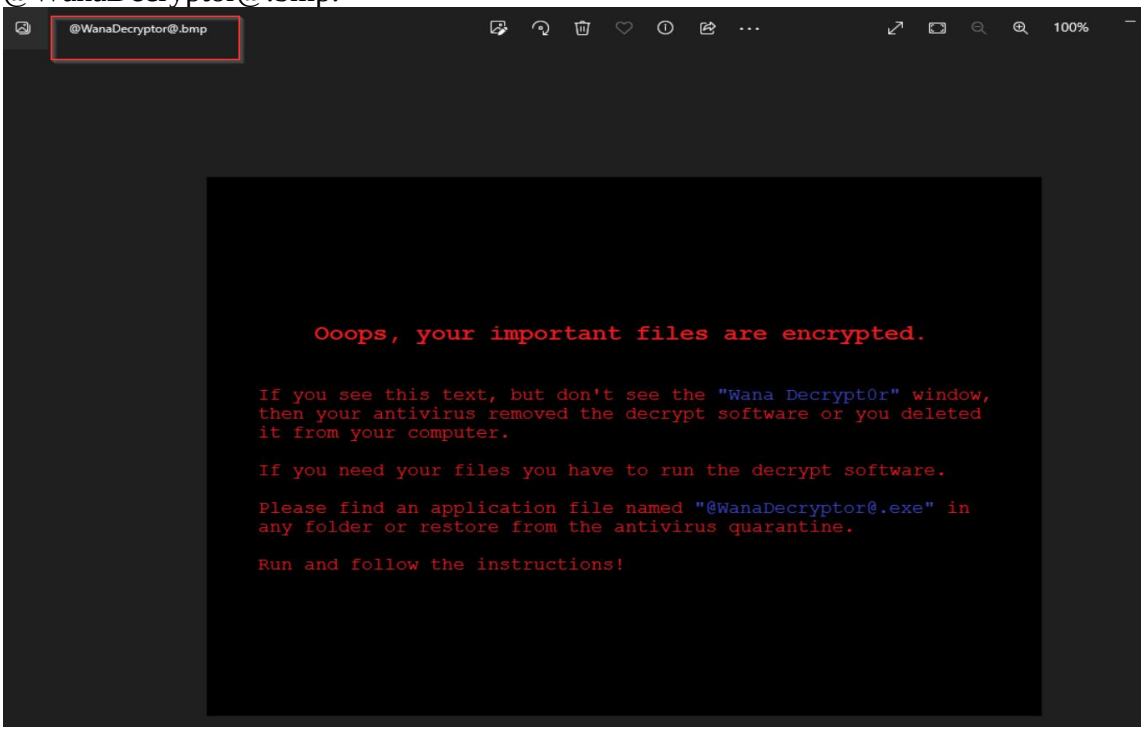


Figure 11: b.wnry is @WanaDecryptor@.bmp

(viii) r.wnry:

- r.wnry contains the ransom note:

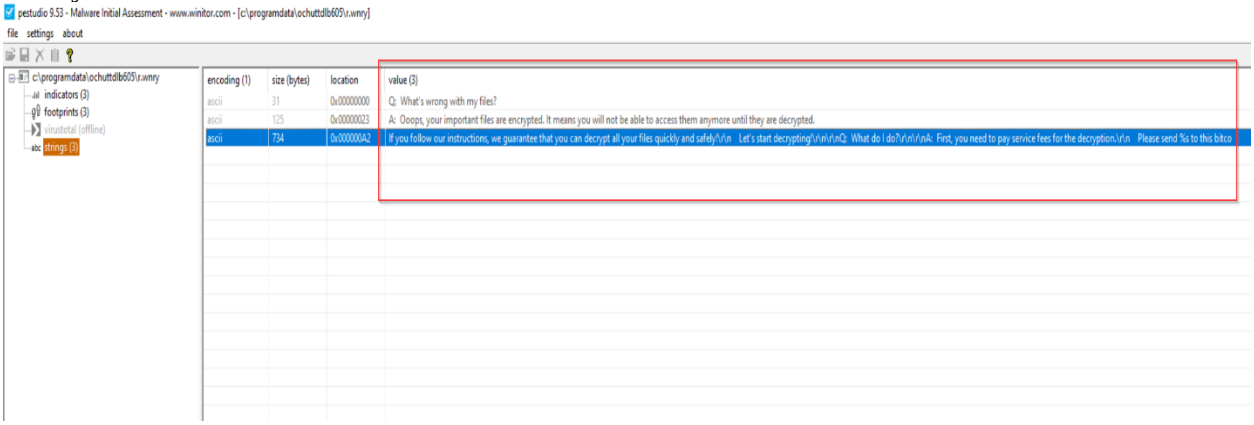


Figure 12: r.wnry analyzed using PEStudio

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

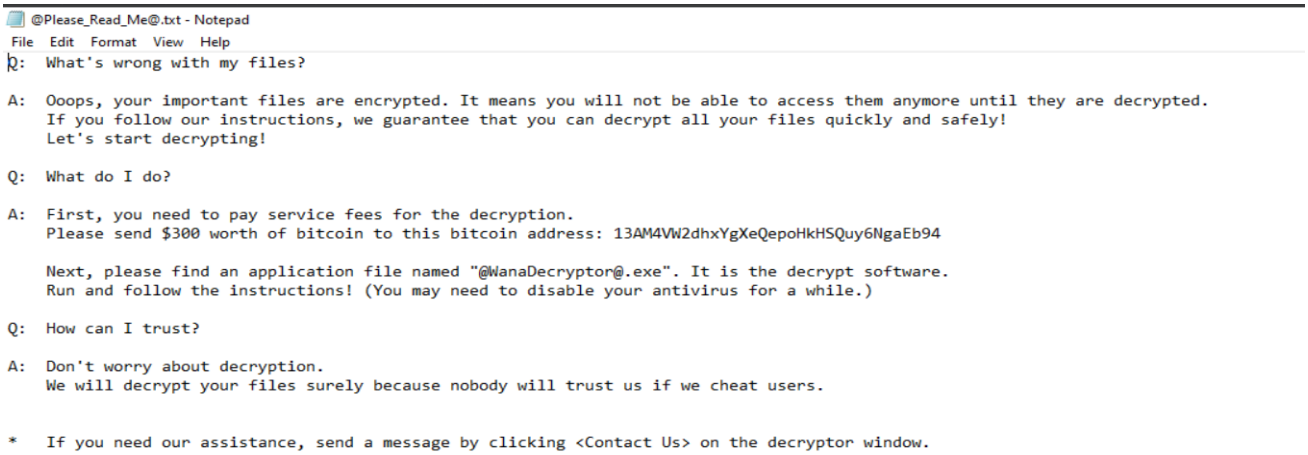


Figure 13: r.wnry is @Please_Read_Me@.txt (ransom note in desktop)

(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:

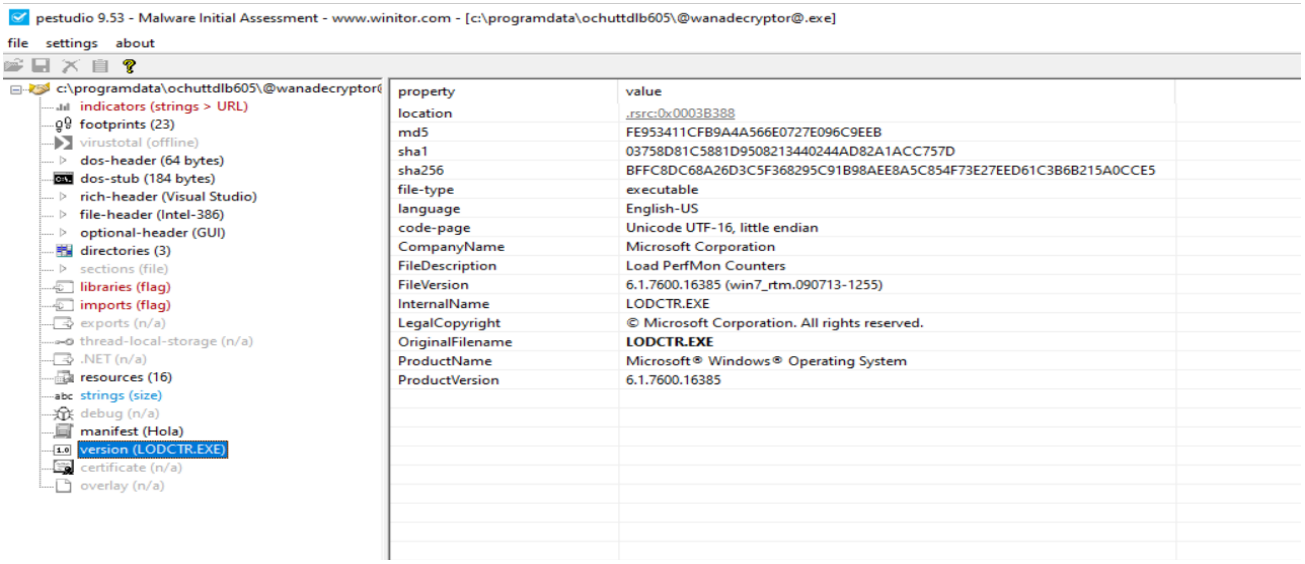


Figure 14: u.wnry is @wanaDecryptor@.exe (PEstudio)

Basic Static Analysis

1. Analysis using floss

- Important strings highlighted:

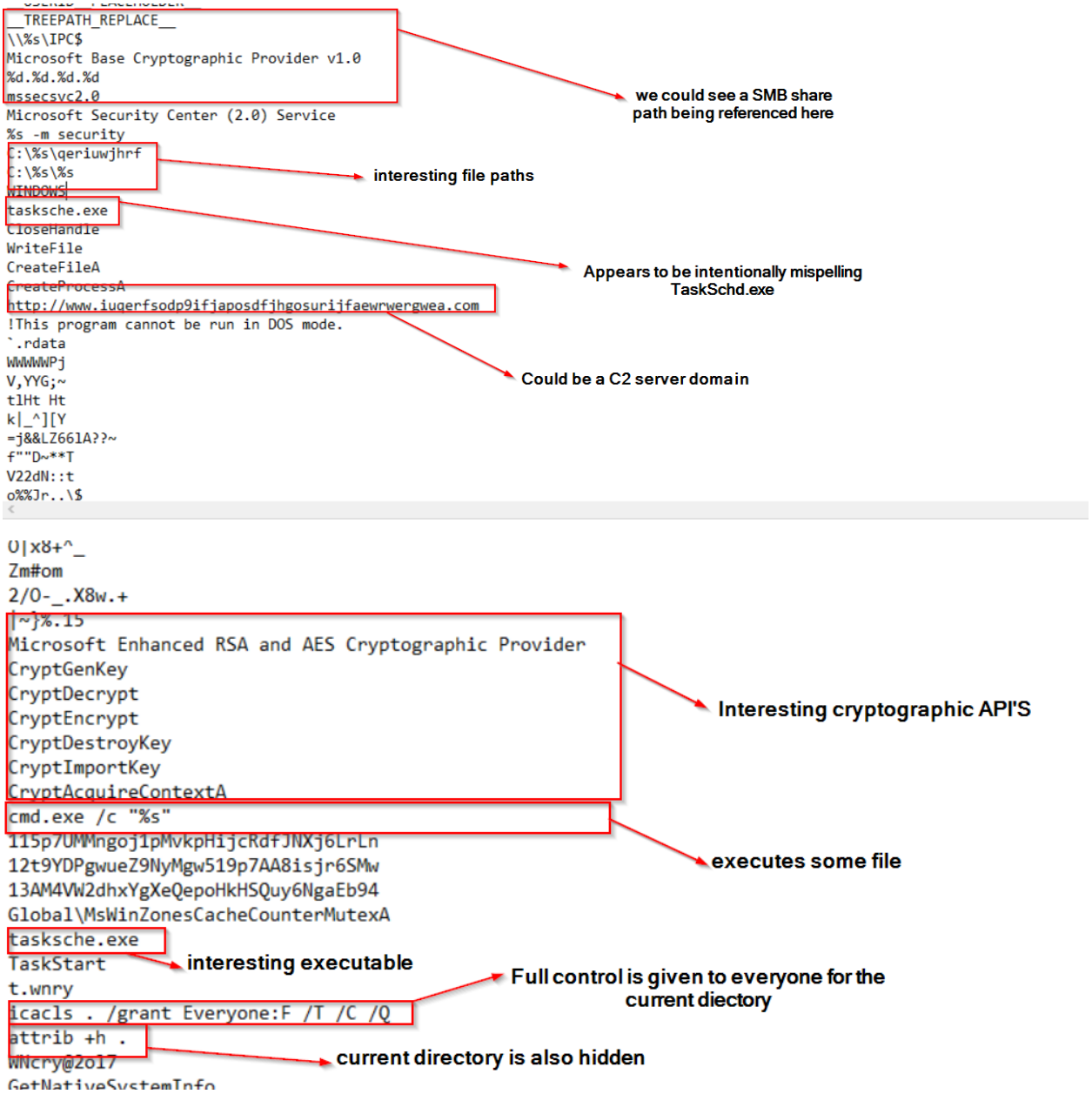


Figure 15: floss output

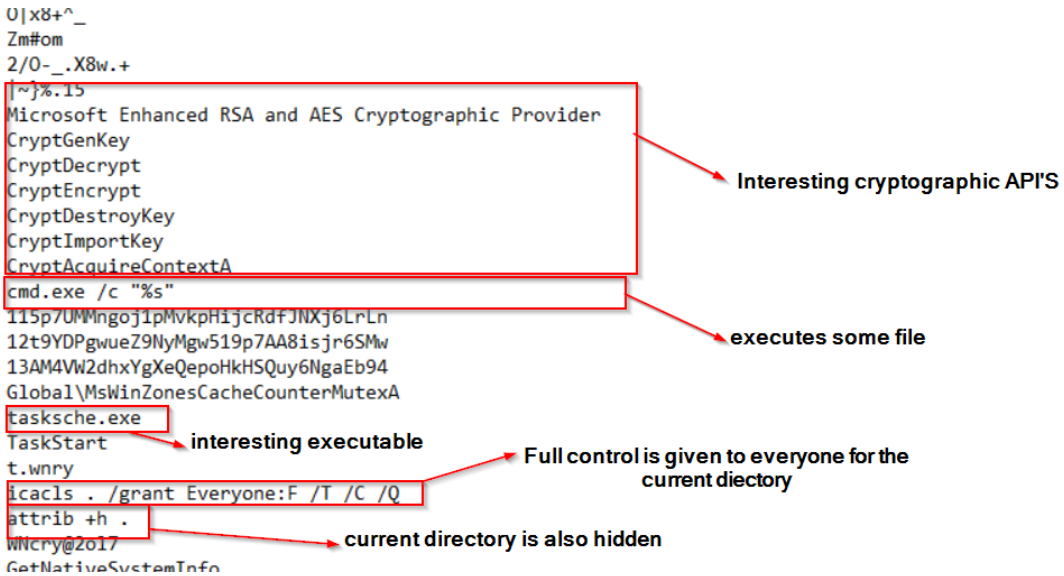


Figure 16: floss output

2. Analysis using PEStudio

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

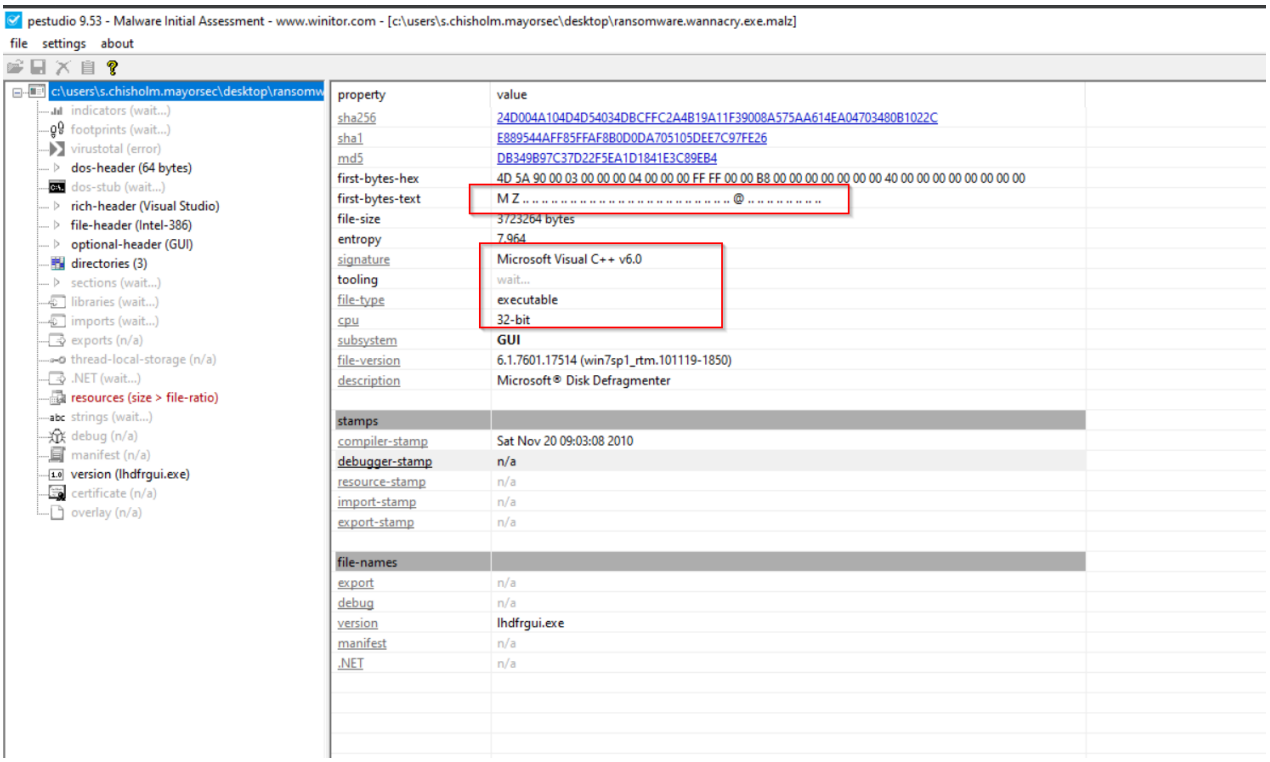


Figure 17: PE studio output

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

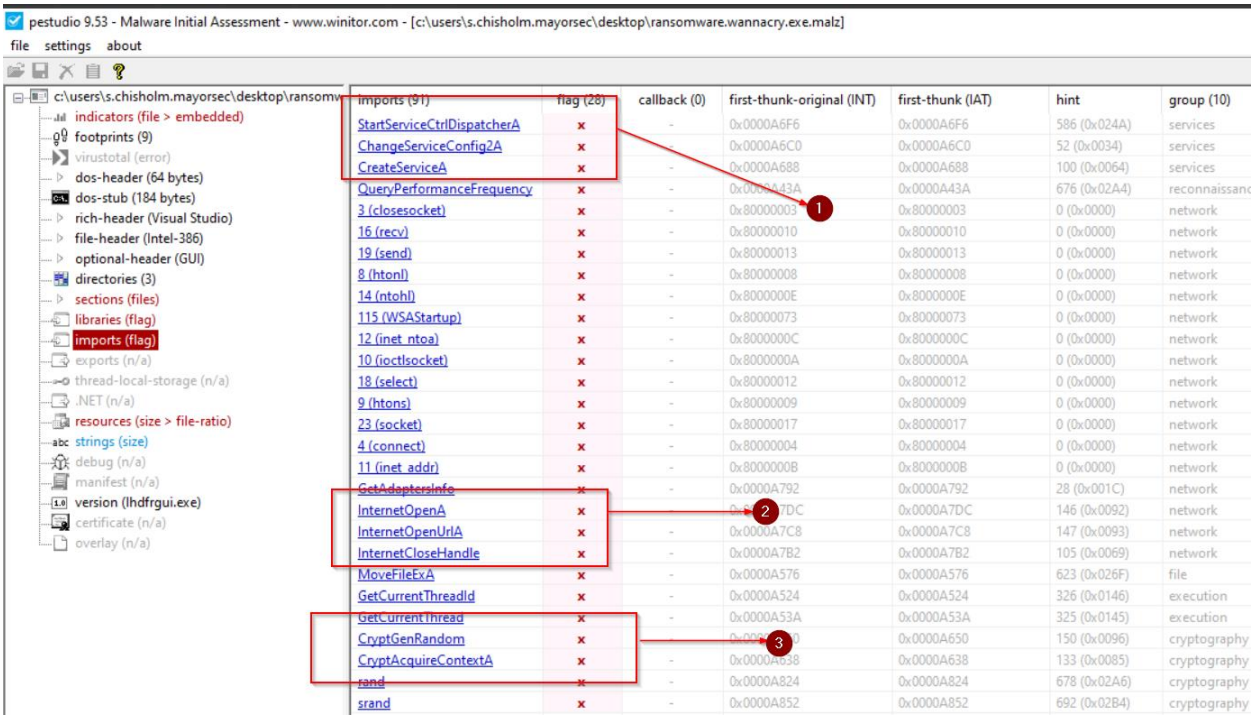


Figure 18: Analyzing APIs used by malware using PE studio

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[.]iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea[.]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.

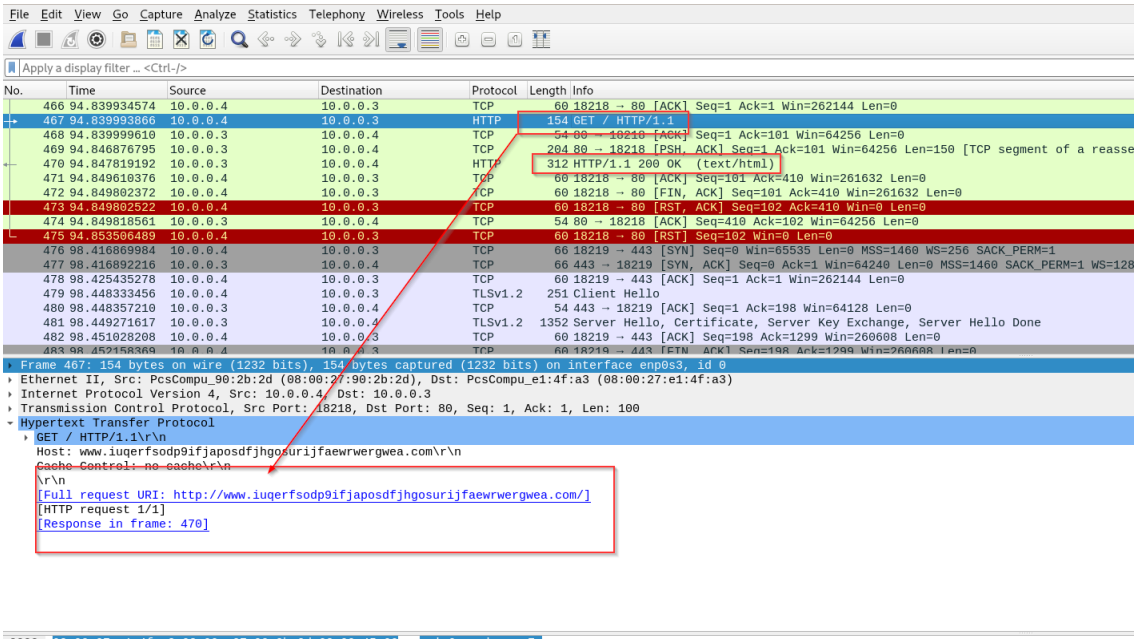


Figure 19: Wireshark output (from Remnux VM)

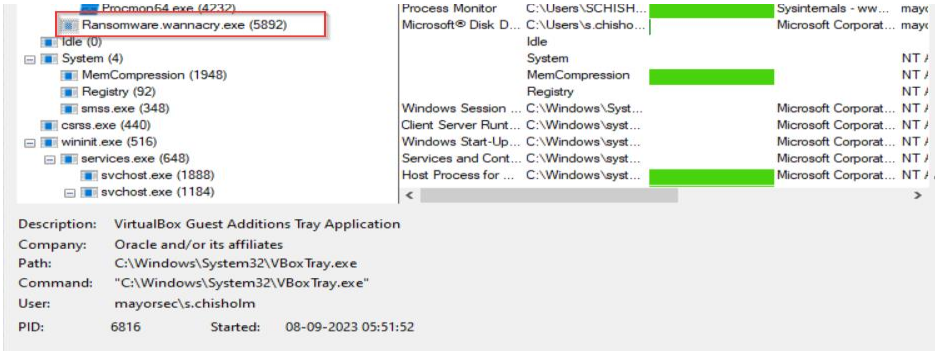


Figure 20: Prcomon's processtree (shows that the malware didn't execute further)

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

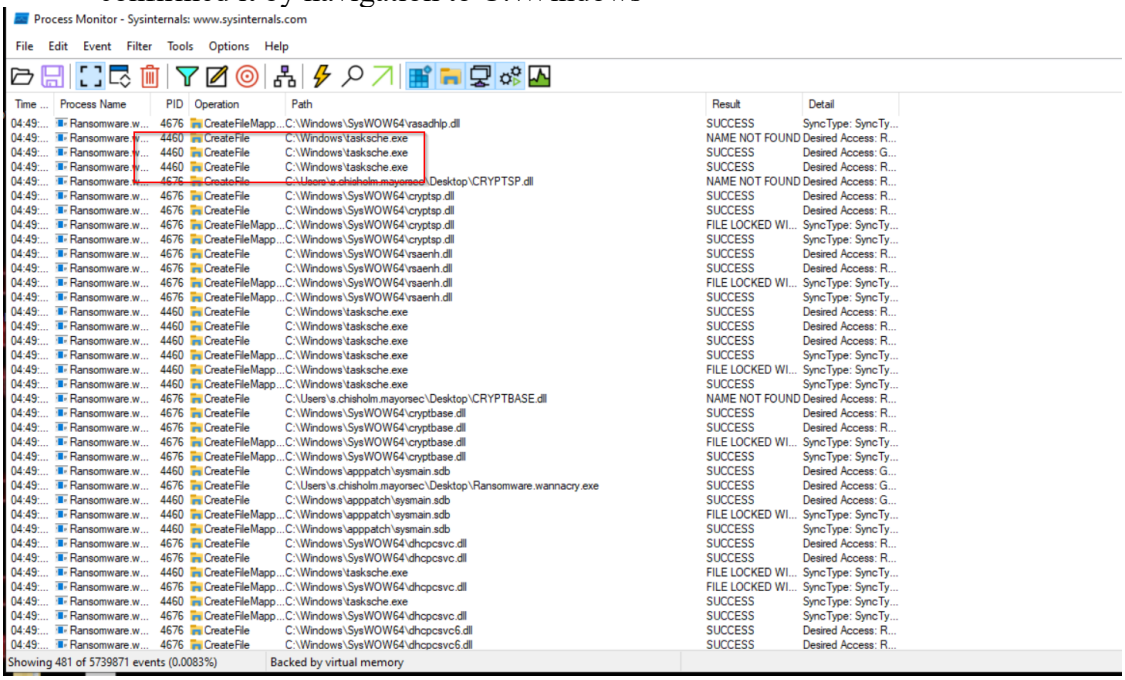


Figure 22: Creation of a file(tasksche.exe) (Procmon output)

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

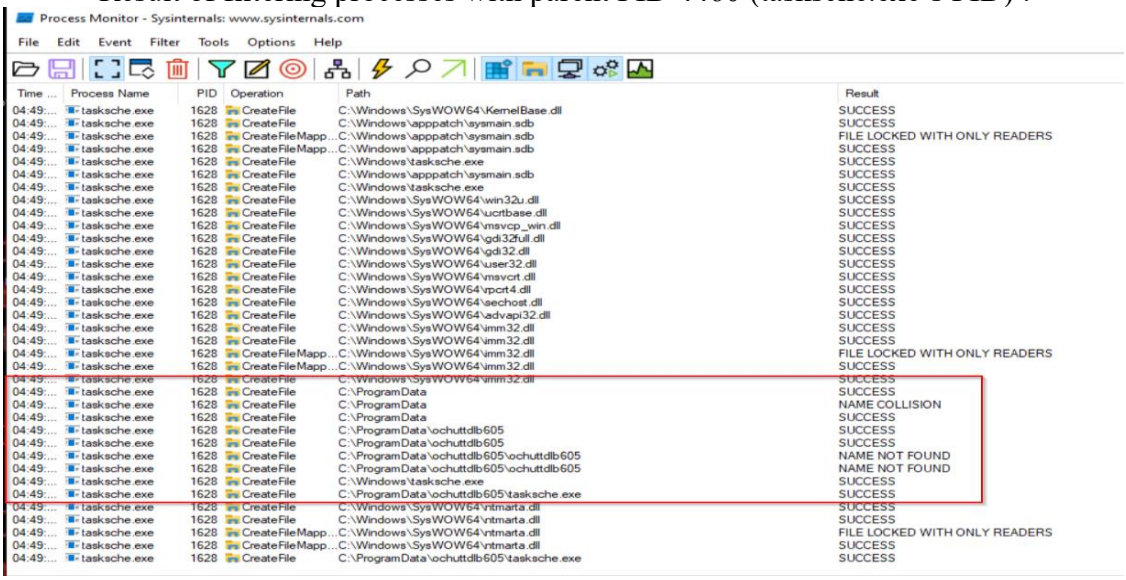


Figure 23: tasksche.exe creates C:\ProgramData\ochuttdlb605 (hidden directory) (Procmon outpit)

(iii) Finding Host Based Indicators using Process Hacker 2

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

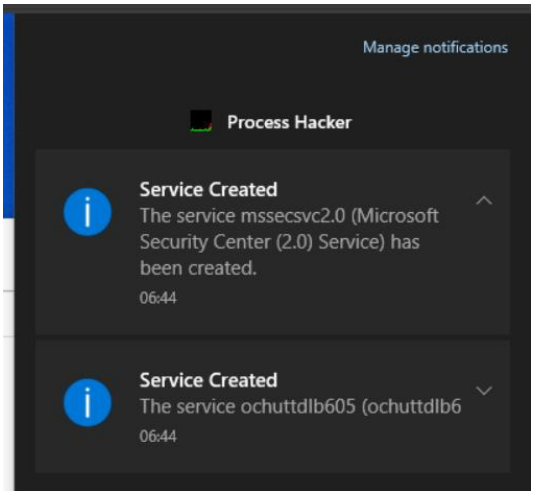


Figure 24: Process Hacker 2 output

- 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)

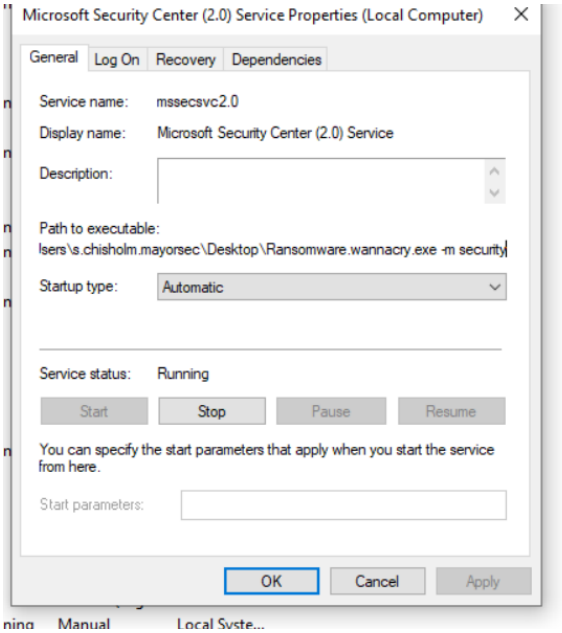


Figure 25: Screenshot of mssecsvc2.0 service (Windows Service)

(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.

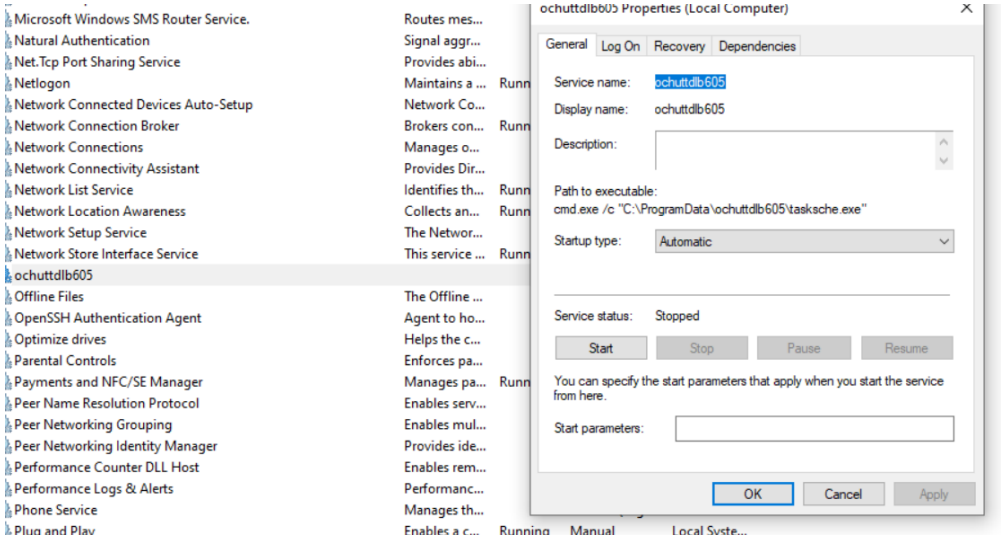


Figure 26: Screenshot of ochuttdlb605 service (Windows Service)

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

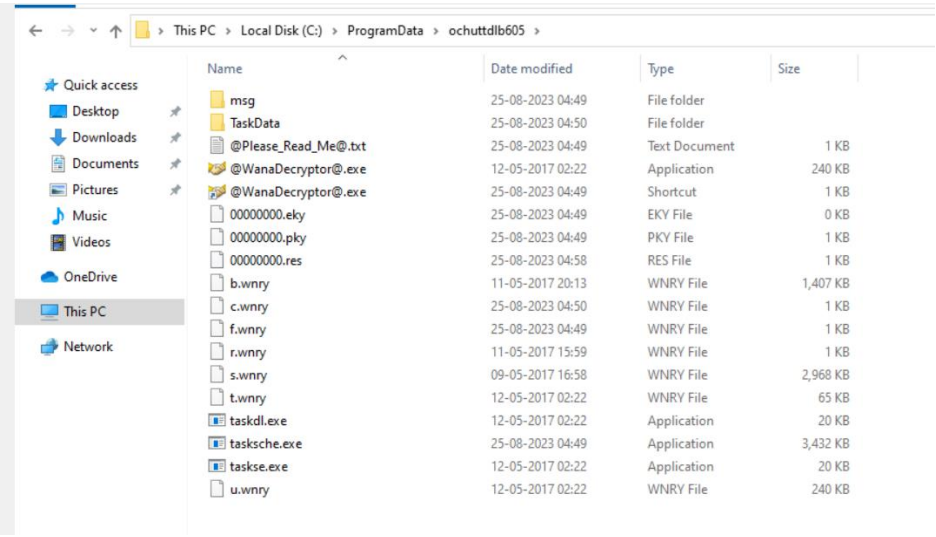


Figure 27: Screenshot of C:\ProgramData\ochuttdlb605 location

- mssecsvc2.0 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)

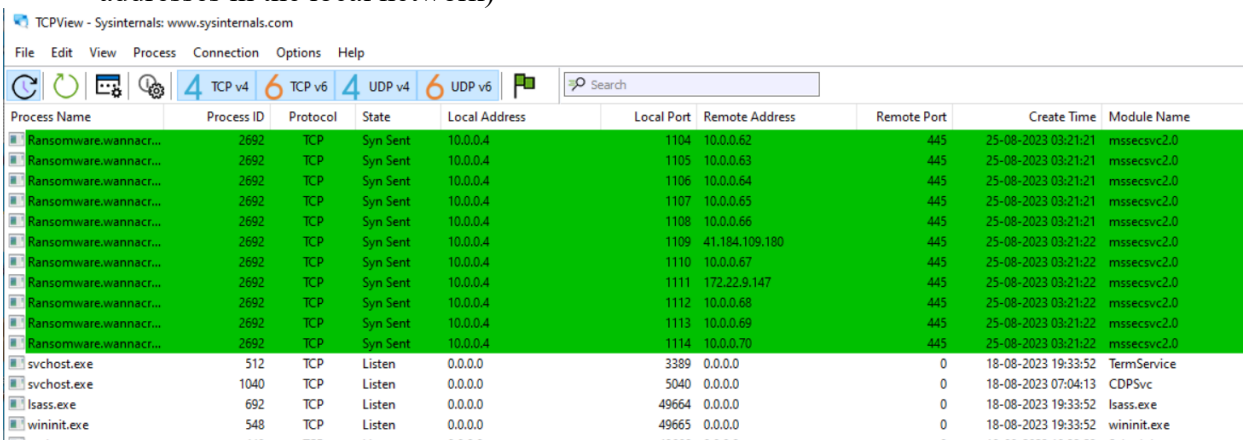


Figure 28: TCPView output

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

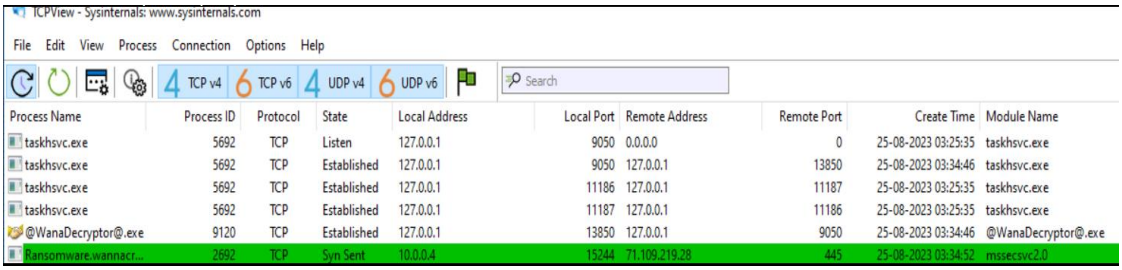


Figure 29: TCPView output (taskhsvc.exe)

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

1. Firstly two services are created i.e., mssecsvc2.0 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.

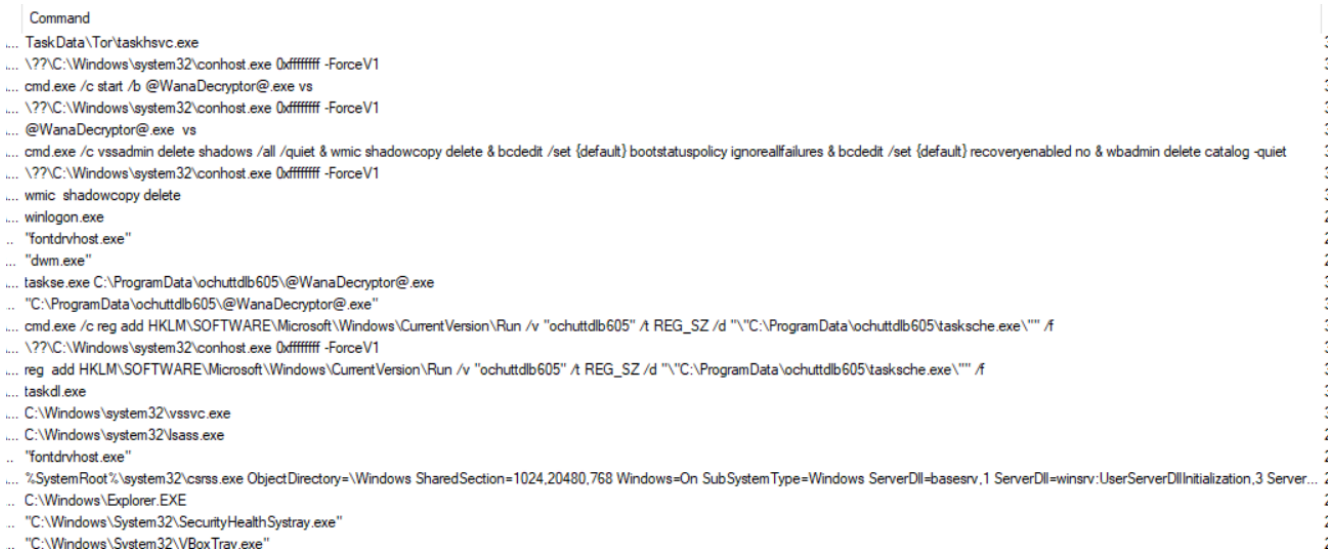


Figure 30: Process tree view (procmon) during and post detonation

```
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
.. taskdl.exe
.. C:\Windows\system32\cmd.exe /c 89381693379808.bat
.. \??\C:\Windows\system32\conhost.exe 0xffffffff -ForceV1
.. cscript.exe  //nologo m.vbs
.. taskdl.exe
.. taskdl.exe
.. taskdl.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 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: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:

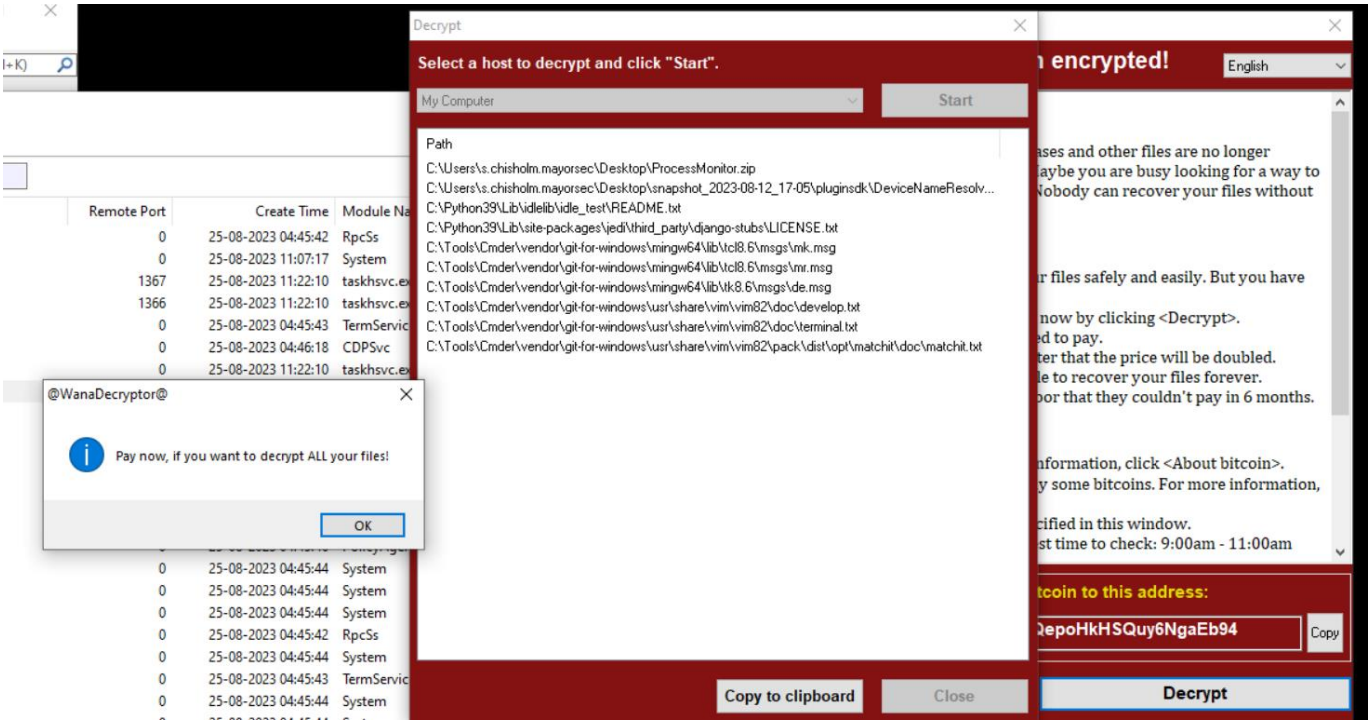


Figure 32: @WanaDecryptor@.exe decrypting certain number of random files as sample

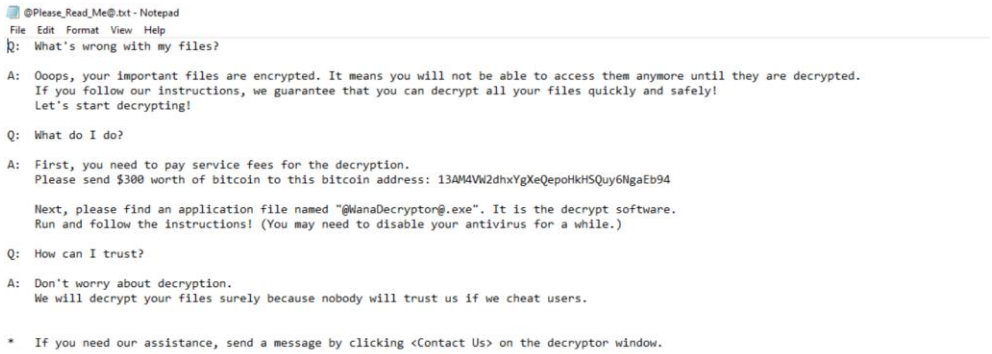


Figure 33: ransom note in the desktop containing the instructions for decryption and ransom payment

- 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)

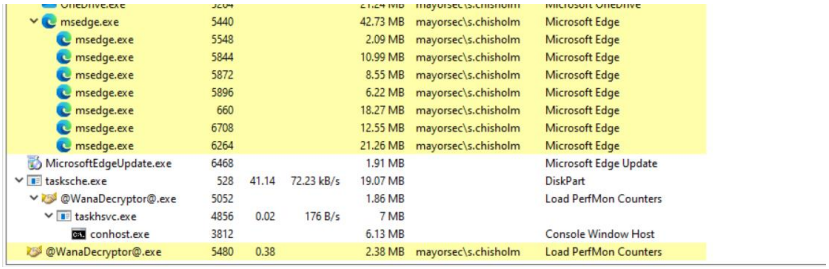


Figure 34: @WanaDecryptor@.exe launching taskhsvc.exe (captured from processs tree of procmon)

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

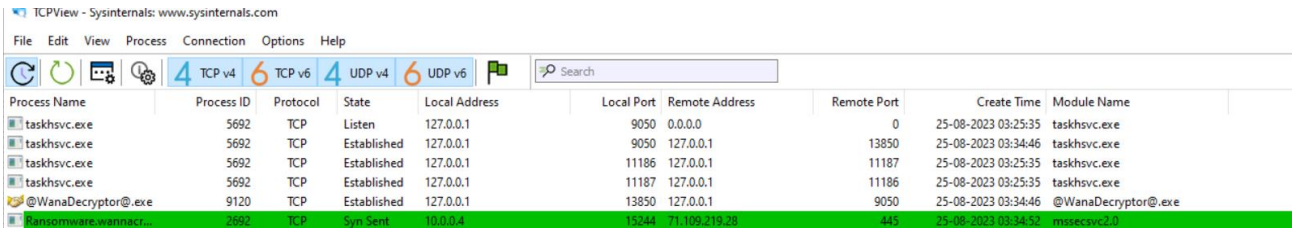


Figure 35: TCPview Output

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[.]iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea[.]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 egister.
- 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)

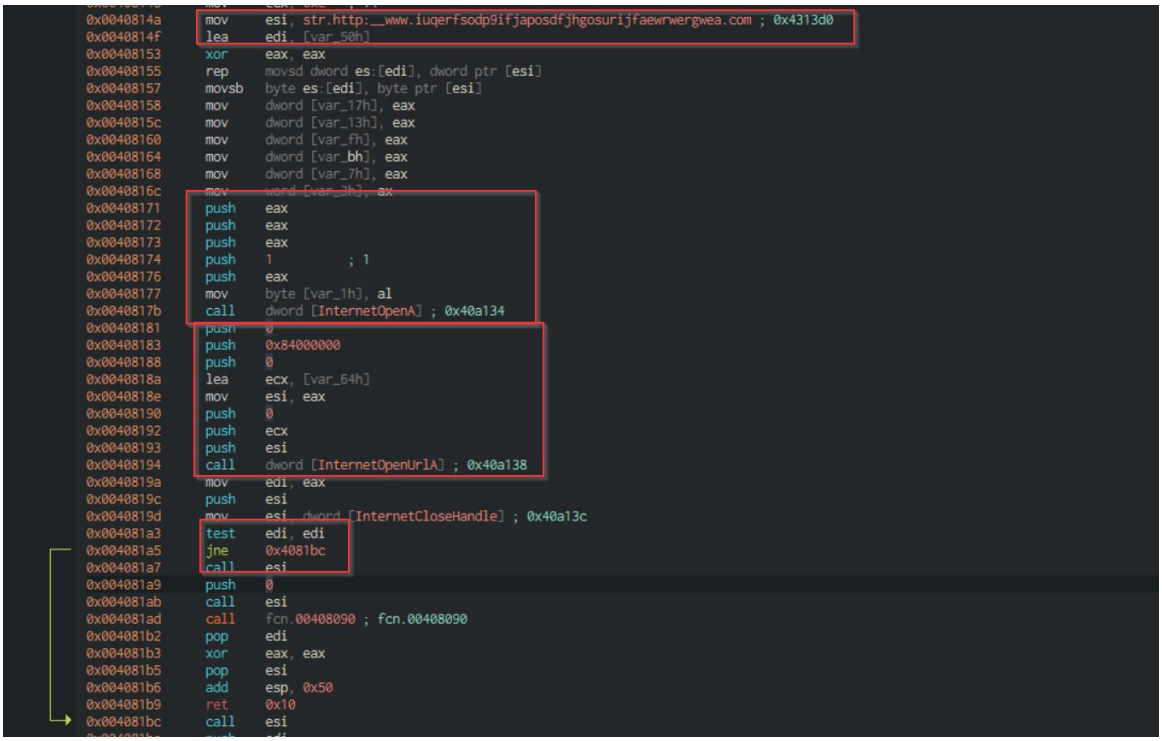


Figure 37: Cutter's graphical view

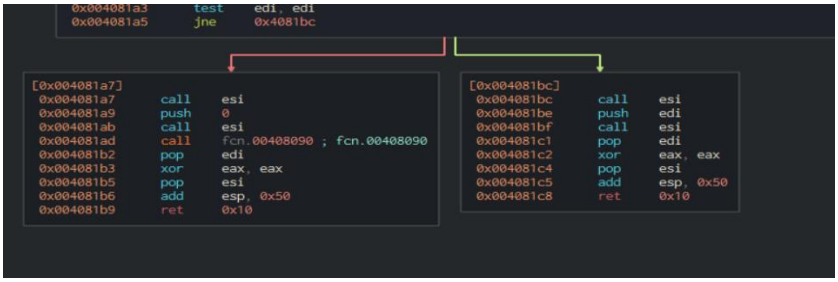


Figure 38: Cutter's graphical vi

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[.]iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea[.]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 (ZE=1) so that the jump is not taken: (ZE=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)

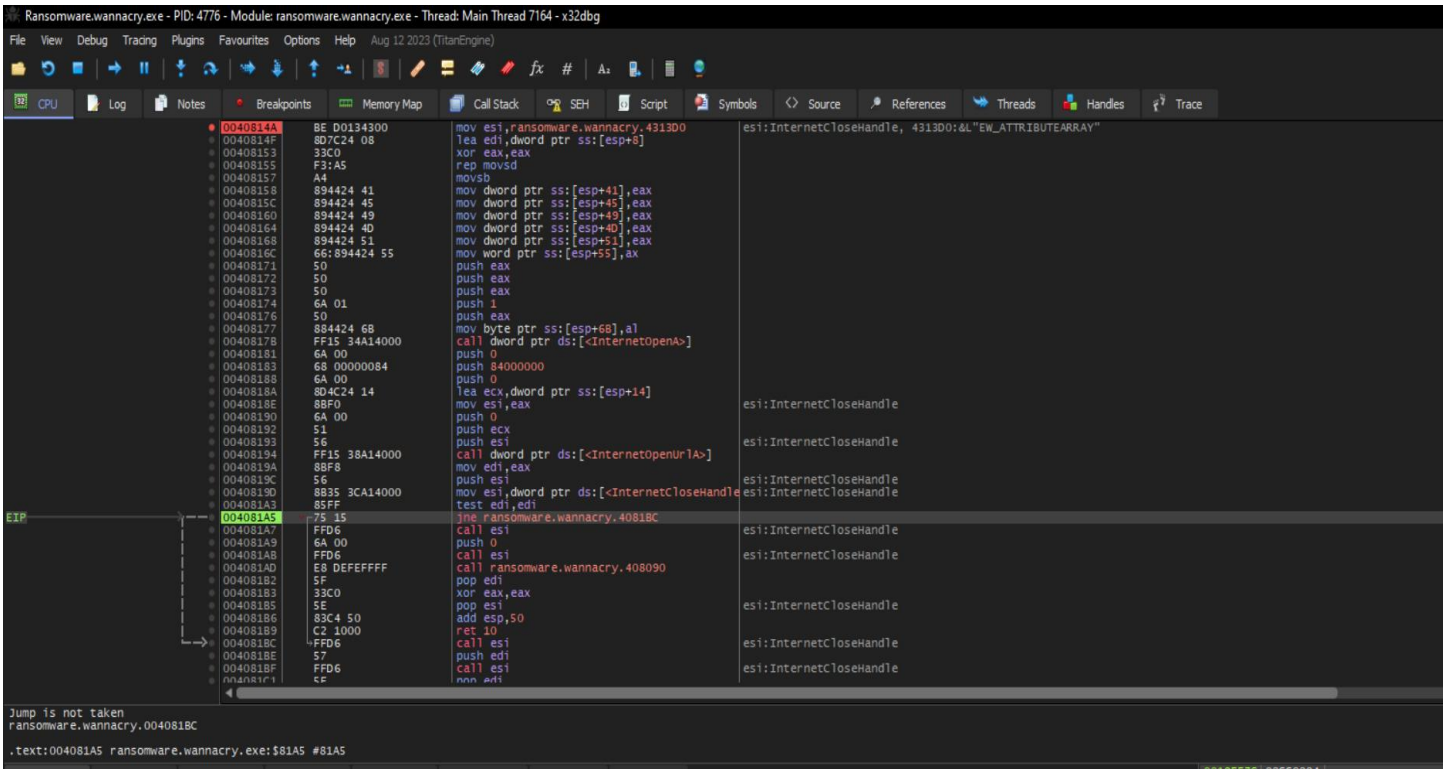


Figure 39: Setting the zero flag in debugger (x32Dbg.exe so jump won't be taken)

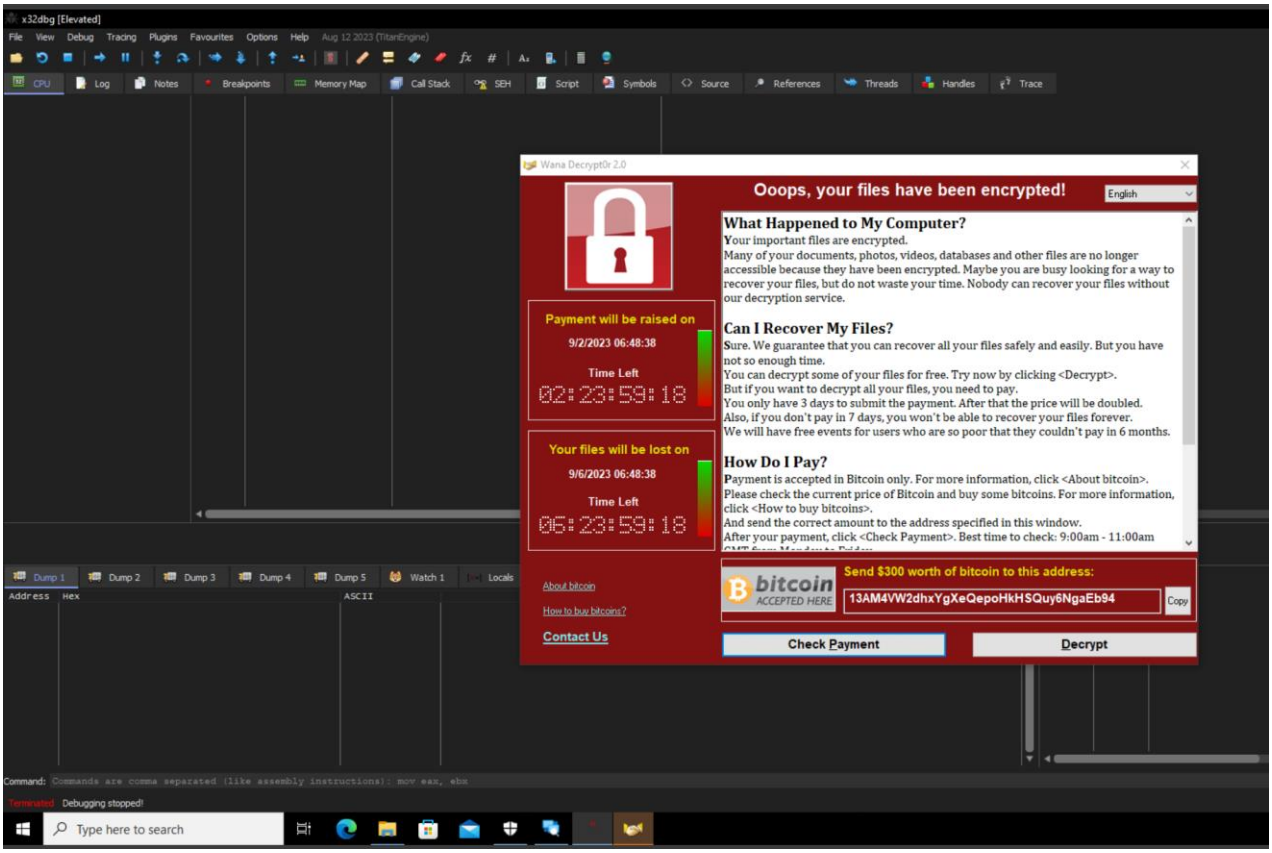


Figure 40: Malware getting executed after setting zero flag to 1 while connected to inetsim

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[.]iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea[.]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 taskshsvc.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 wnry when we analyzed the strings present in the binary using floss. These strings too indicate that the executable we have is wannacry ransomware.

Appendices

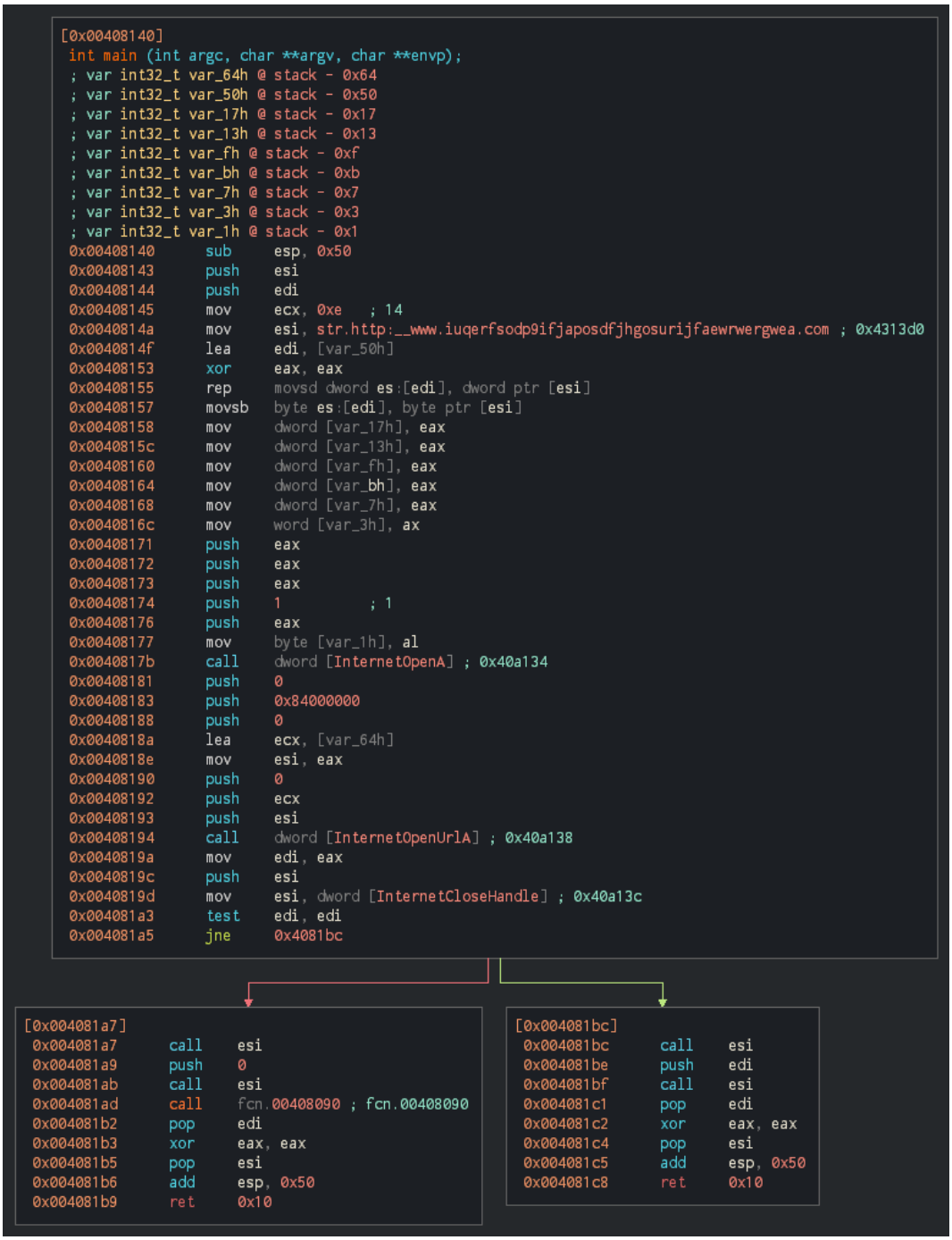
A. Yara Rules

```
rule Wannacry_Ransomware {
    meta:
        last_updated = "2023-09-09"
        author = "killswitch"
        description = "A sample Yara rule for Wannacry Ransomware"
        Hash = "6C382A1C16DBA41B4FF6F0D728E9E92AEBFE2EE0C7FEB30A0E63B15EAF6C4B44"

    strings:
        $PE_magic_byte = "MZ"
        $string1 = "http://www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com"
        $string2="tasksche.exe"
        $string3="wnry"
        $string4 = "\\\\%s\\IPC$"


    condition:
        $PE_magic_byte at 0 and $string1 and $string2 and $string3 and $string4
}
```

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.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.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
```