Practical Malware Analysis & Triage
Malware Analysis Report

WannaCry Ransomware

Course-Final

Table of Contents

Executive Summary

SHA256 hash	24d004a104d4d54034dbcffc2a4b19a11f39008a575aa614ea04703480b1022c

WannaCry is ransomware that struck in May 2017, exploiting a Microsoft SMB vulnerability (EternalBlue) to spread rapidly across networks. It encrypted files and demanded Bitcoin ransom for decryption. Industries like healthcare and finance were hit hard.

The binary is a standalone executable that will first check for connectivity to an external URL that is embedded into it. If a connection is established nothing happens, but when the binary is not able to connect to the hard-coded URL it starts its malicious activity.

Dumps a bunch of malicious scripts and then it starts encrypting each and every single user,system file present in the machine. After which we get the ransom demand screen and instructions to carry out to decrypt the files.

Fig1: WannaCry Ransomware

High-Level Technical Summary

The WannaCry ransomware is a standalone malware binary that, when executed, will encrypt all system and user files on the machine, lock the system, and demand a ransom payment, without which the decryption key will not be provided.

The roots of WannaCry arise from an SMB vulnerability present on Windows systems called EternalBlue. EternalBlue exploits a buffer overflow vulnerability in the SMBv1 (Server Message Block) protocol (CVE-2017-0144).

In the context of WannaCry, EternalBlue was used to propagate the ransomware across networks. Once a machine was compromised, WannaCry injected its payload, encrypted files, and spread laterally to other unpatched machines by exploiting the same SMB vulnerability, without requiring user interaction.

Initially after detonation the malware gets the system prepared to make a internet connection to an external URL (***URL[defanged]: hxxp[://]www[.]iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea[.]com***). If the URL is reachable the malware just closes the connection.

In other cases, the malware starts calling out sub-functions inside of it and starts encrypting the system and user files. After encryption there is a pop-up box that demands for ransom payment in Bitcoins. Post which the victim is provided with a decryption key to decrypt their files.

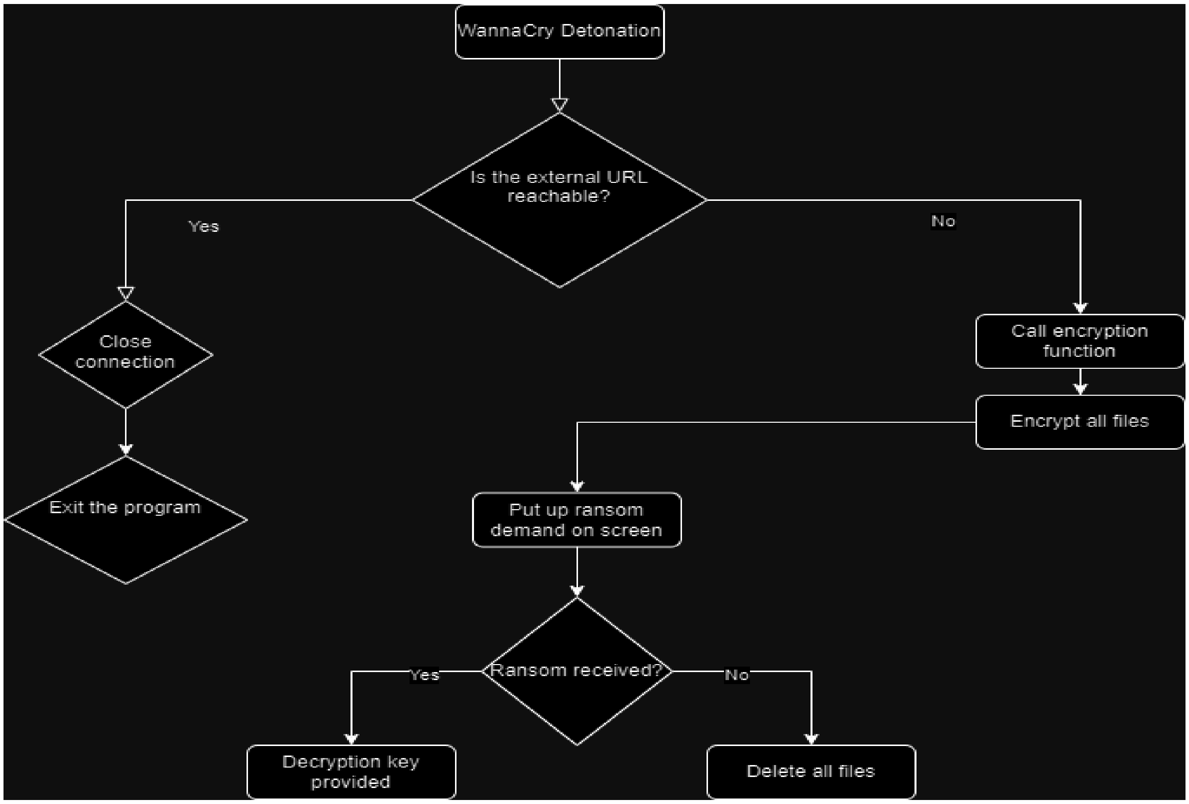


Fig2: Basic Workflow of WannaCry

Malware Composition

WannaCry consists of the following components:

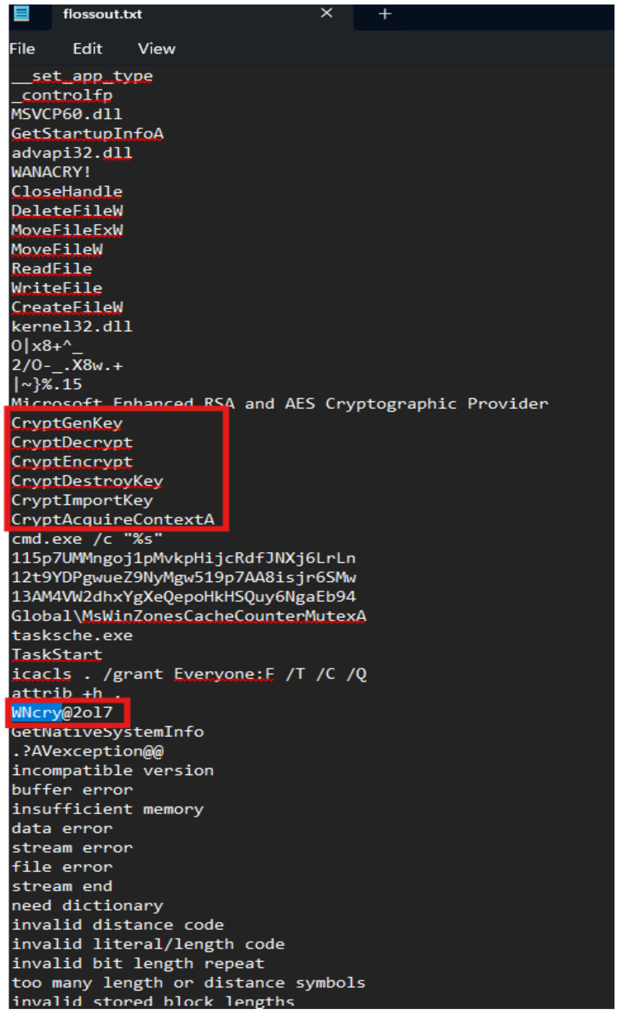
File Name	SHA256 Hash
Ransomware.wannacry.exe	24d004a104d4d54034dbcffc2a4b19a11f39008a575aa614ea04703480b1022c

This malware has a very basic composition. The malware is standalone meaning all of its functionality is packaged into a single binary rather than downloading and executing a second stage.

Though there are some processes that the binary spawns off during its execution. Details of which can be found later in the report.

Basic Static Analysis

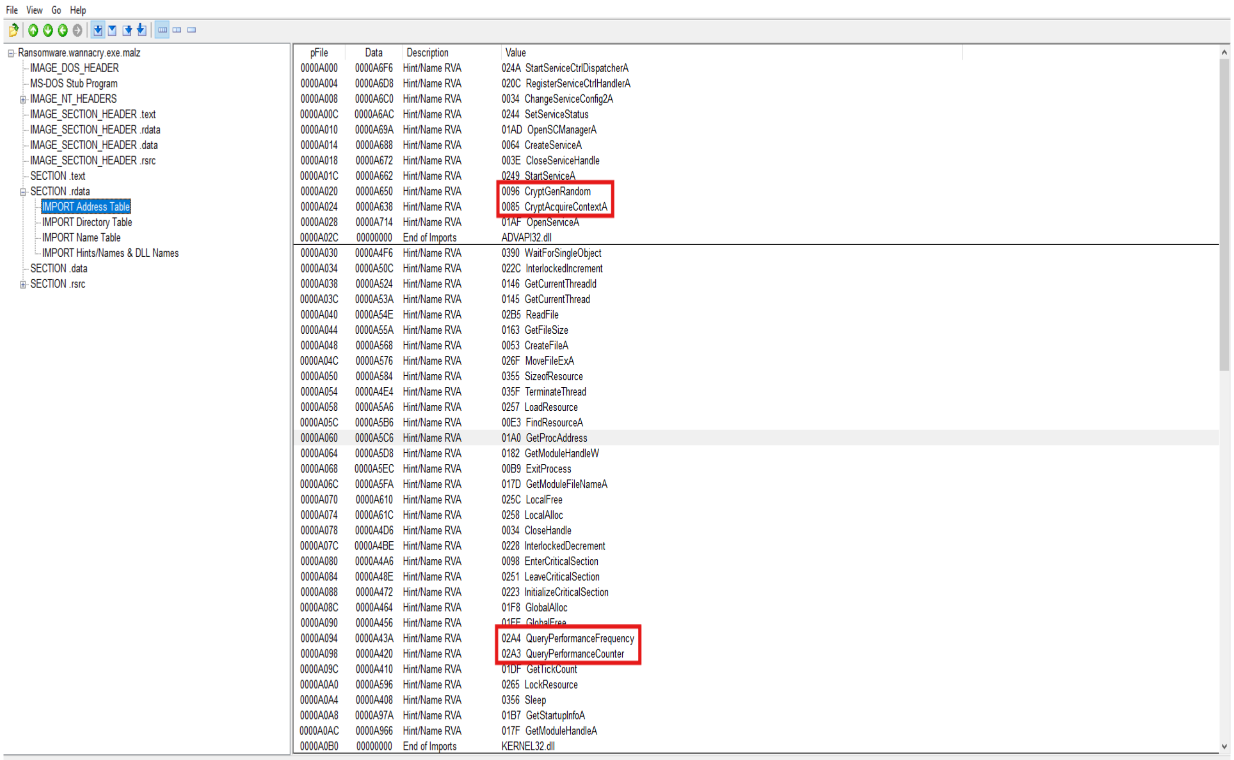
Here are my observations for the binary:



```
flossout.txt
File    Edit    View
__set_app_type
_controlfp
MSVCP60.dll
GetStartupInfoA
advapi32.dll
WANACRY!
CloseHandle
DeleteFileW
MoveFileExW
MoveFileW
ReadFile
WriteFile
CreateFileW
kernel32.dll
O|x8+^_
2/O-_.X8w.+
|~}%.15
Microsoft Enhanced RSA and AES Cryptographic Provider
CryptGenKey
CryptDecrypt
CryptEncrypt
CryptDestroyKey
CryptImportKey
CryptAcquireContextA
cmd.exe /c "%s"
115p7UMMngoj1pMvkpHijcRdfJNXj6LrLn
12t9YDPgwueZ9NyMgw519p7AA8isjr6SMw
13AM4VW2dhxYgXeQepoHkHSQuy6NgaEb94
Global\MsWinZonesCacheCounterMutexA
tasksche.exe
TaskStart
icacls . /grant Everyone:F /T /C /Q
attrib +h .
WNcry@2ol7
GetNativeSystemInfo
.?AVexception@@
incompatible version
buffer error
insufficient memory
data error
stream error
file error
stream end
need dictionary
invalid distance code
invalid literal/length code
invalid bit length repeat
too many length or distance symbols
invalid stored block lengths
```

Fig3: Floss output from the binary

The strings highlighted in the image raise an eyebrow, as the function calls printed out seem to perform some kind of cryptographic function. Adding to this a similar set of strings were found while analyzing the binary in PEview.



pFile	Data	Description	Value
0000A000	0000A6F6	Hint/Name RVA	024A StartServiceCtrlDispatcherA
0000A004	0000A6D8	Hint/Name RVA	020C RegisterServiceCtrlHandlerA
0000A008	0000A6C0	Hint/Name RVA	0034 ChangeServiceConfig2A
0000A00C	0000A6AC	Hint/Name RVA	0244 SetServiceStatus
0000A010	0000A69A	Hint/Name RVA	01AD OpenSCManagerA
0000A014	0000A688	Hint/Name RVA	0064 CreateServiceA
0000A018	0000A672	Hint/Name RVA	003E CloseServiceHandle
0000A01C	0000A662	Hint/Name RVA	0249 StartServiceA
0000A020	0000A650	Hint/Name RVA	0096 CryptGenRandom
0000A024	0000A638	Hint/Name RVA	0085 CryptAcquireContextA
0000A028	0000A714	Hint/Name RVA	01AF OpenServiceA
0000A02C	00000000	End of Imports	ADVAPI32.dll
0000A030	0000A4F6	Hint/Name RVA	0390 WaitForSingleObject
0000A034	0000A50C	Hint/Name RVA	022C InterlockedIncrement
0000A038	0000A524	Hint/Name RVA	0146 GetCurrentThreadId
0000A03C	0000A53A	Hint/Name RVA	0145 GetCurrentThread
0000A040	0000A54E	Hint/Name RVA	02B5 ReadFile
0000A044	0000A55A	Hint/Name RVA	0163 GetFileSize
0000A048	0000A568	Hint/Name RVA	0053 CreateFileA
0000A04C	0000A576	Hint/Name RVA	026F MoveFileExA
0000A050	0000A584	Hint/Name RVA	0355 SizeofResource
0000A054	0000A4E4	Hint/Name RVA	035F TerminateThread
0000A058	0000A5A6	Hint/Name RVA	0257 LoadResource
0000A05C	0000A5B6	Hint/Name RVA	00E3 FindResourceA
0000A060	0000A5C6	Hint/Name RVA	01A0 GetProcAddress
0000A064	0000A5D8	Hint/Name RVA	0182 GetModuleHandleW
0000A068	0000A5EC	Hint/Name RVA	00B9 ExitProcess
0000A06C	0000A5FA	Hint/Name RVA	017D GetModuleFileNameA
0000A070	0000A610	Hint/Name RVA	025C LocalFree
0000A074	0000A61C	Hint/Name RVA	0258 LocalAlloc
0000A078	0000A4D6	Hint/Name RVA	0034 CloseHandle
0000A07C	0000A4BE	Hint/Name RVA	0228 InterlockedDecrement
0000A080	0000A4A6	Hint/Name RVA	0098 EnterCriticalSection
0000A084	0000A48E	Hint/Name RVA	0251 LeaveCriticalSection
0000A088	0000A472	Hint/Name RVA	0223 InitializeCriticalSection
0000A08C	0000A464	Hint/Name RVA	01F8 GlobalAlloc
0000A090	0000A456	Hint/Name RVA	01FF GlobalFree
0000A094	0000A43A	Hint/Name RVA	02A4 QueryPerformanceFrequency
0000A098	0000A420	Hint/Name RVA	02A3 QueryPerformanceCounter
0000A09C	0000A410	Hint/Name RVA	01DF GetTickCount
0000A0A0	0000A596	Hint/Name RVA	0265 LockResource
0000A0A4	0000A408	Hint/Name RVA	0356 Sleep
0000A0A8	0000A97A	Hint/Name RVA	01B7 GetStartupInfoA
0000A0AC	0000A966	Hint/Name RVA	017F GetModuleHandleA
0000A0B0	00000000	End of Imports	KERNEL32.dll

Fig4: PEview Import Address Table

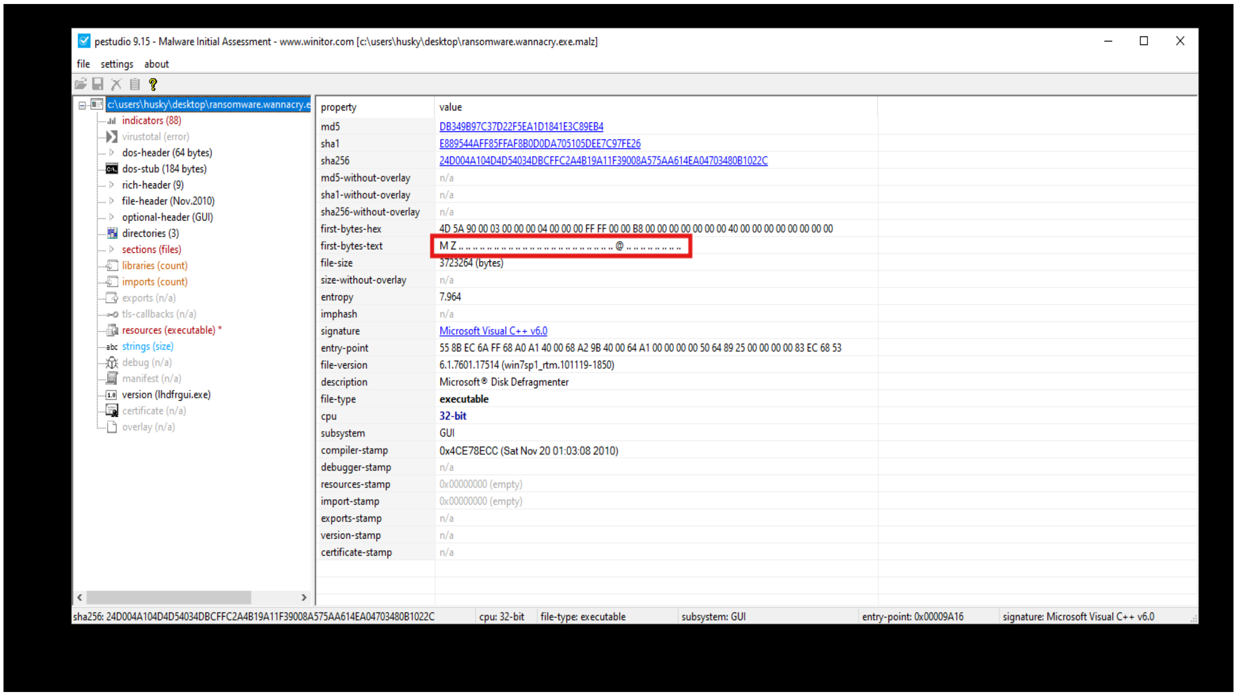


Fig5: PEstudio Output

Basic Dynamic Analysis

With INETsim:

<u>Wireshark Output:-</u>

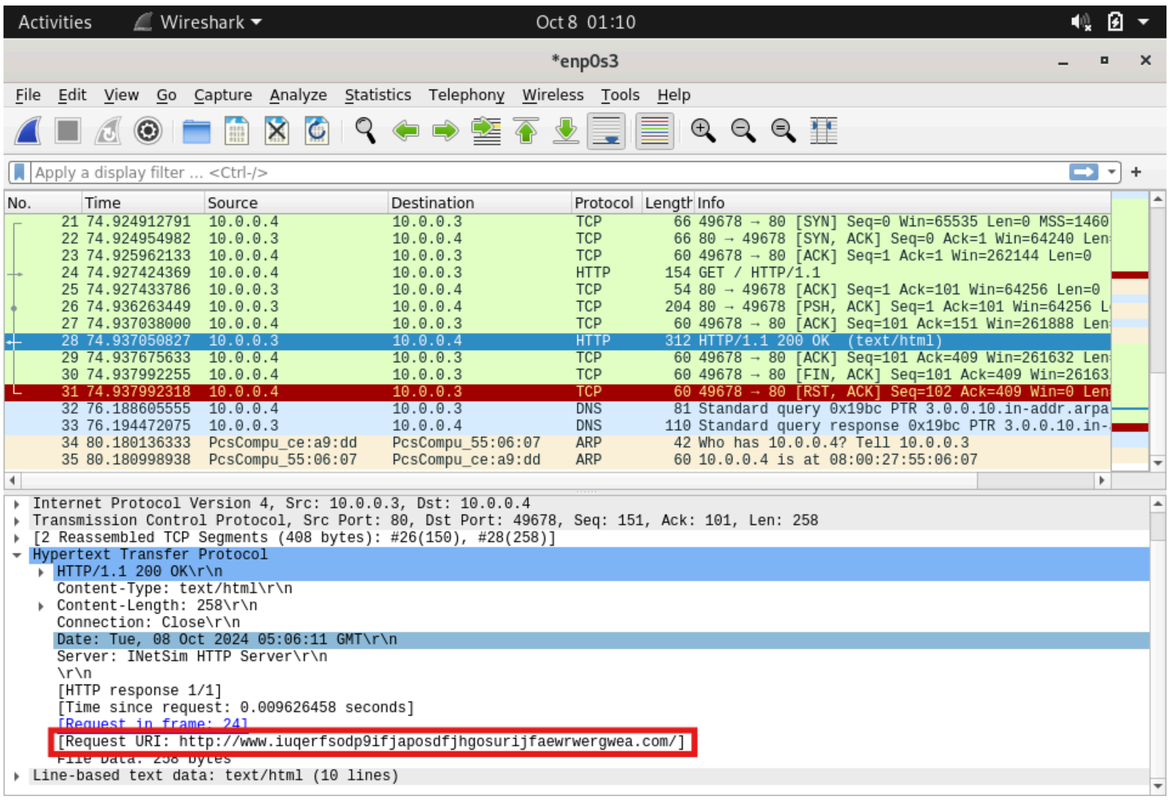


Fig6: Wireshark Output-Connection to external domain

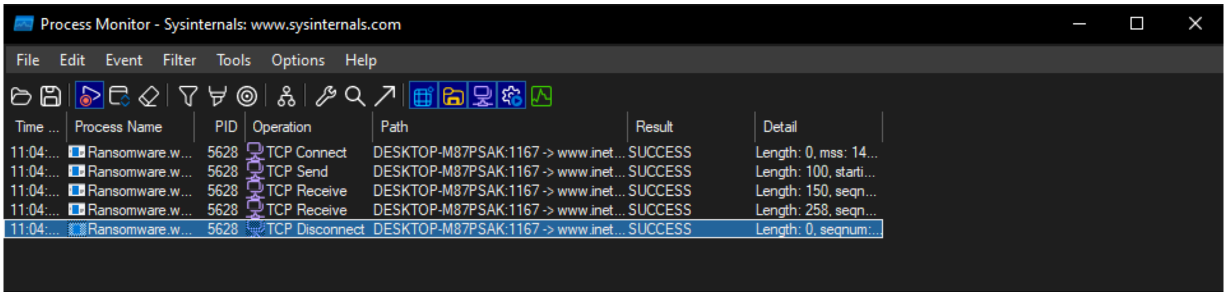


Fig7: Procmon Output for network connectivity

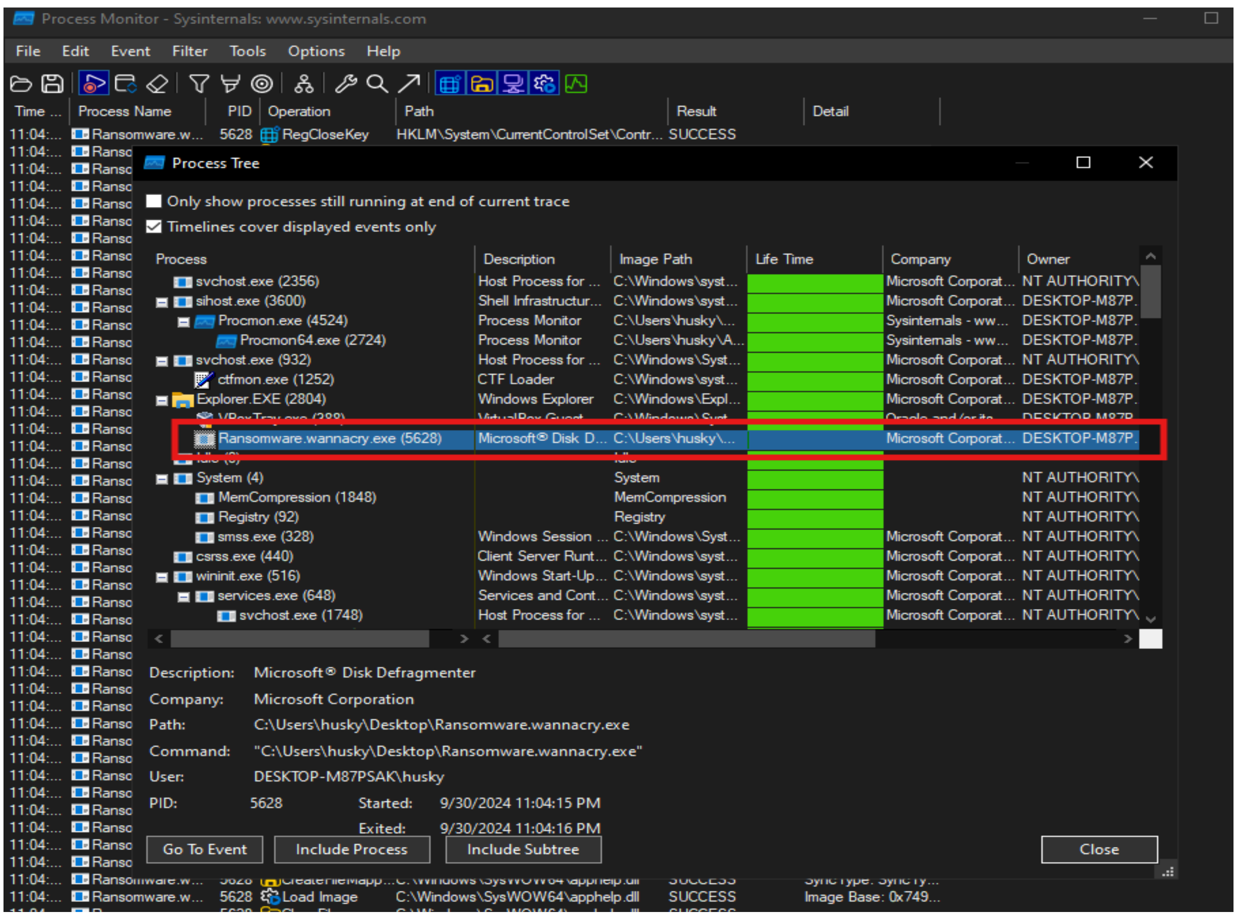


Fig8: Procmon Process Tree Output

The Procmon Process tree also gives out details about some other subtasks that are spawned and run by WannaCry. Details of which can be found in the Indicators of compromise section.

Advanced Static Analysis

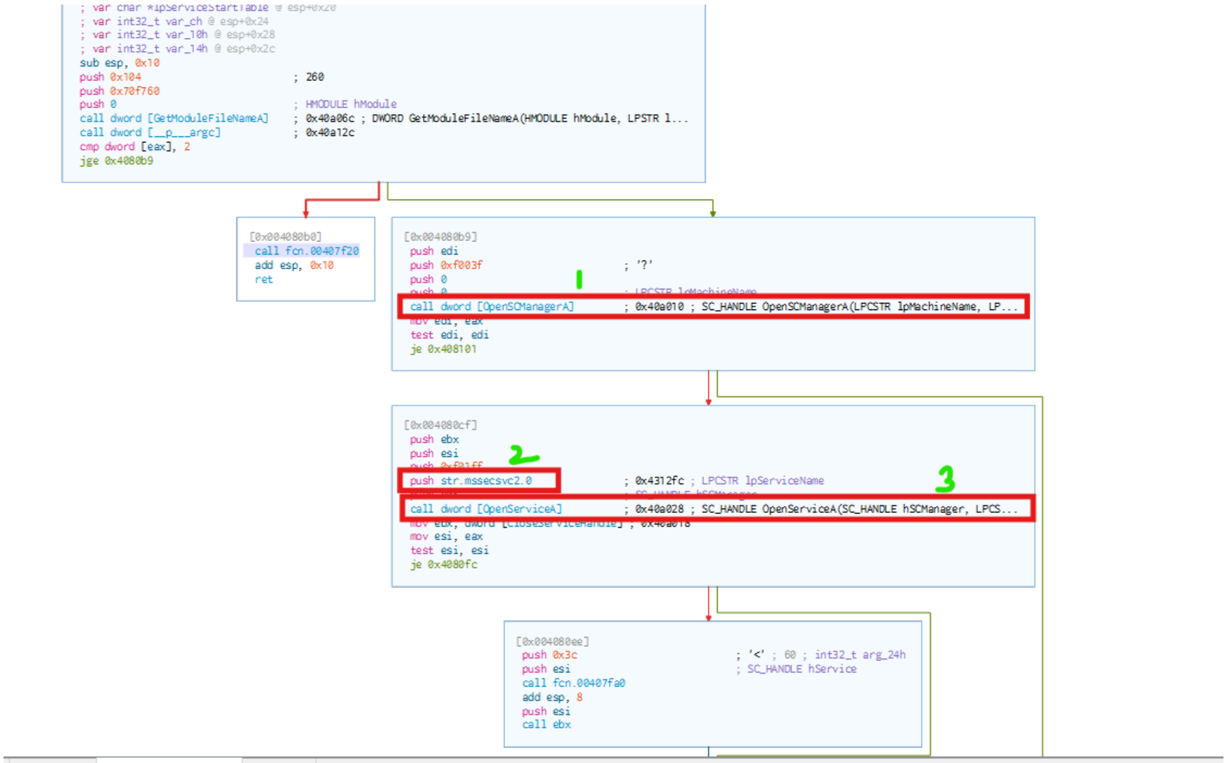


Fig9: Cutter Output

Advanced Dynamic Analysis

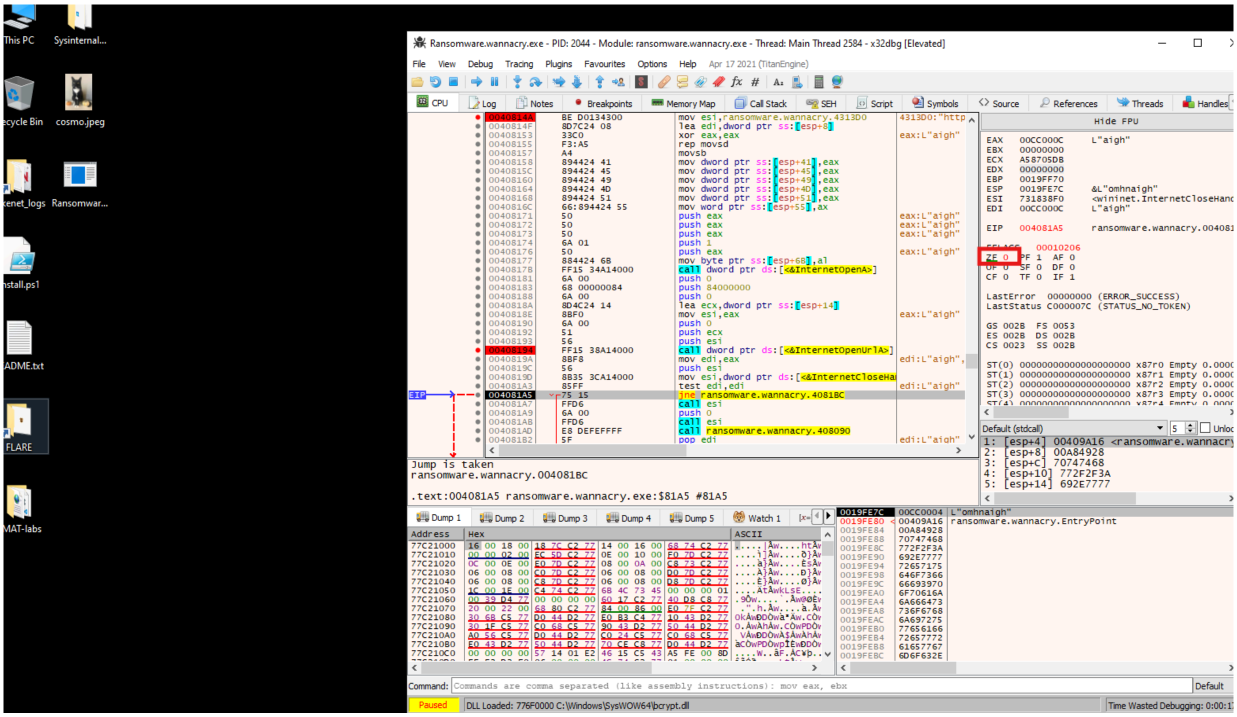


Fig10: Executing binary without patching

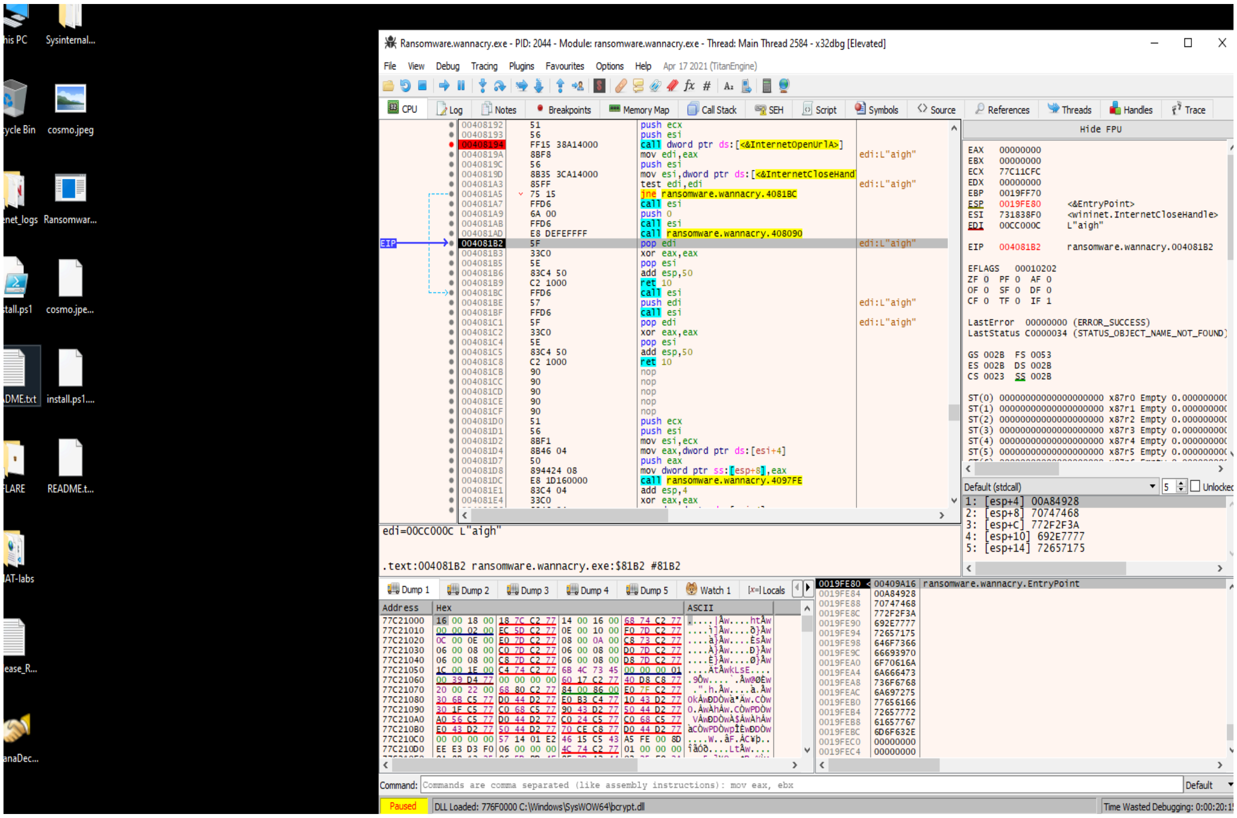


Fig11: Executing after patching binary

Indicators of Compromise

Network Indicators

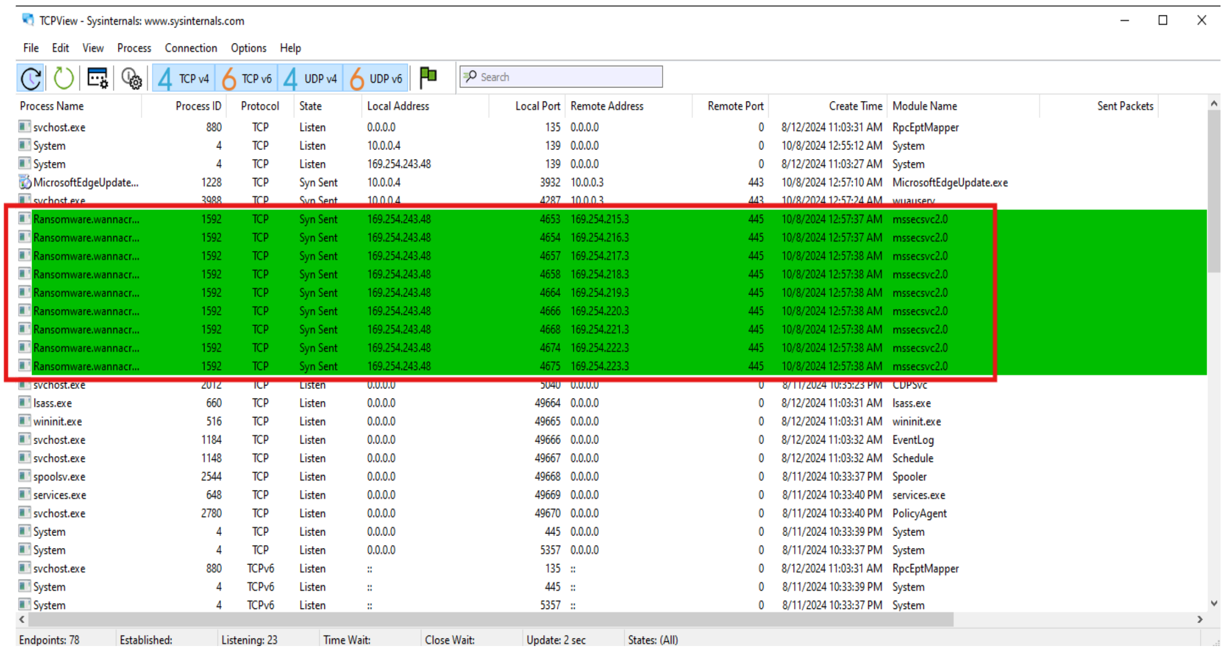


Fig12: TCPview Output displaying network indicators

Host-based Indicators

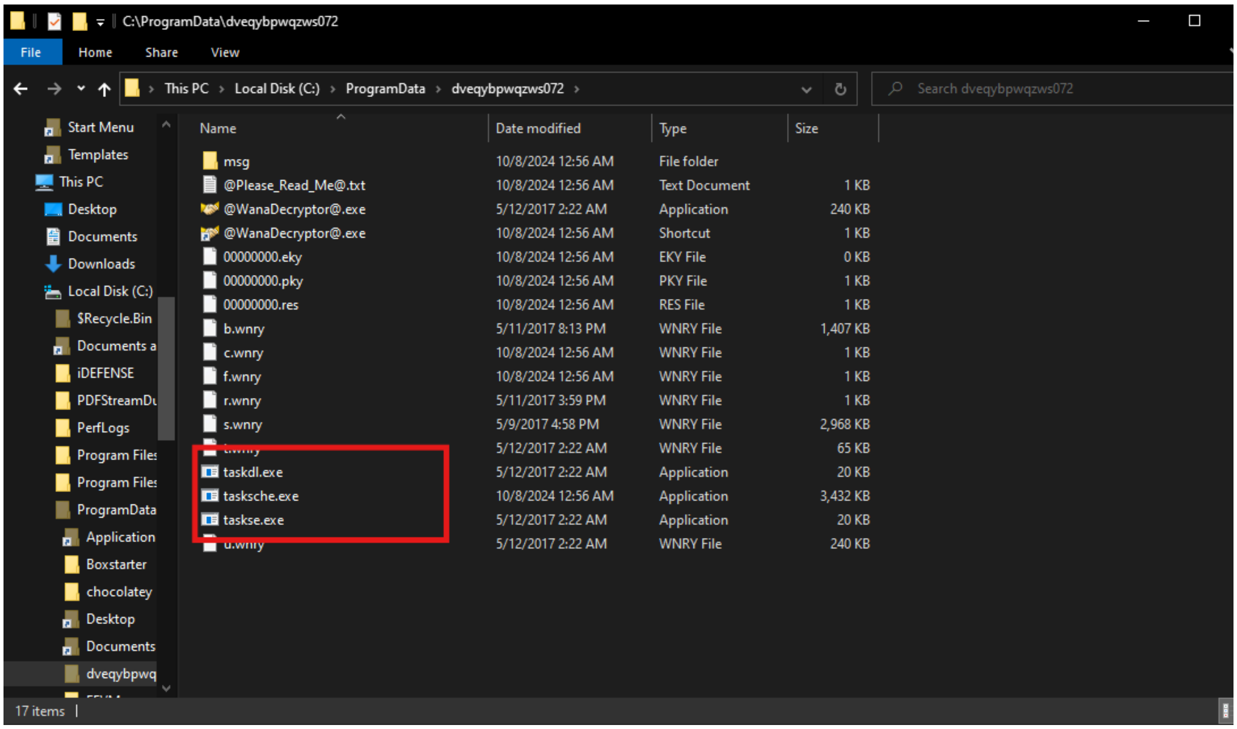


Fig13: Creation of hidden directory with malicious processes.

Note:-

The highlighted executables are the malicious processes that the sample spawns off and executes.

Rules & Signatures

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

- Since this is a windows based binary the PE magic header can be found at the start of the file which can be marked.
- Also the external URL that the malware initially tries to connect with is another such string to look out for.
- We can also look out for the hidden directory that is created once the malware starts its proper functioning.
- Including the processes dropped into that folder.

Appendices

A. Yara Rules

```
rule WannaCry_catcher {

    meta:

        author = "Niyanth Guruprasad"

        Created on = "08-10-2024"

        Last updated on = "08-10-2024"

        Description = "Detection Rule for WannaCry Ransomware,PMAT"


    strings:

        $PE_magic_header_byte = "MZ"

        $External_URL = 
"http://www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com" ascii

        $hidden_directory = "C:\ProgramData\%s"

        $spawned_executables1 = "tasksche.exe"

        $spawned_executbales2 = "taskse.exe"

        $spawned_executables3 = "taskdl.exe"

        $common_file_extension = ".wnry"

    condition:

        $PE_magic_header_byte at 0 and $common_file_extension and $External_URL 
and $hidden_directory and ($spawned_executables1 or $spawned_executables2 or 
$spawned_executables3)

}
```

B. Callback URLs

Domain	Port
hxxp[://]www[.]iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea[.]com	80

C. Decompiled Code Snippets

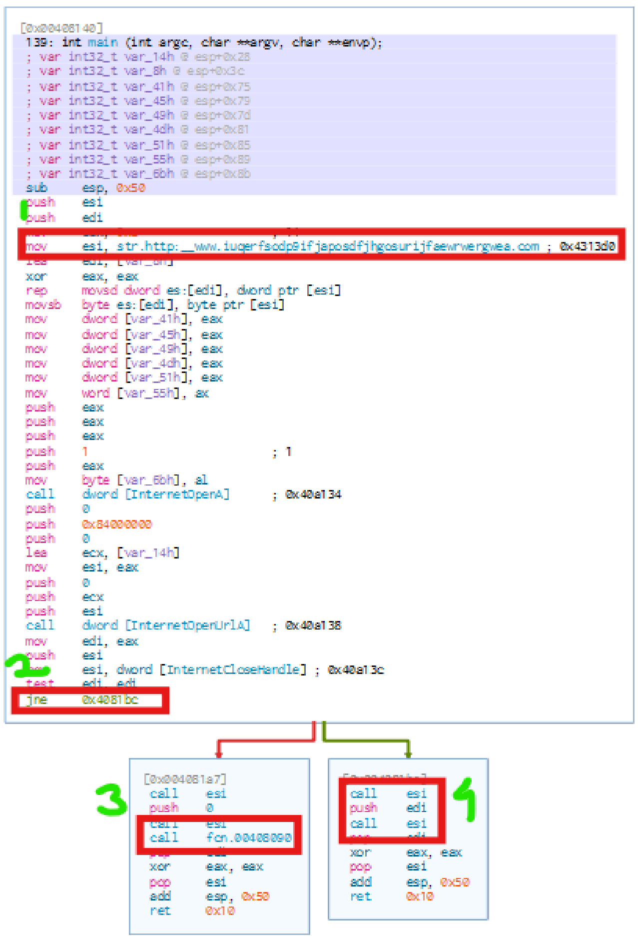


Fig14: Cutter Output of main function