Practical Malware Analysis & Triage

Malware Analysis Report

WannaCry Malware

SEP 2022 | Rishank Shah | v1.0

Table of Contents

Executive Summary

SHA256 hash	24D004A104D4D54034DBCFFC2A4B19A11F39008A575AA614EA04703480B1022C

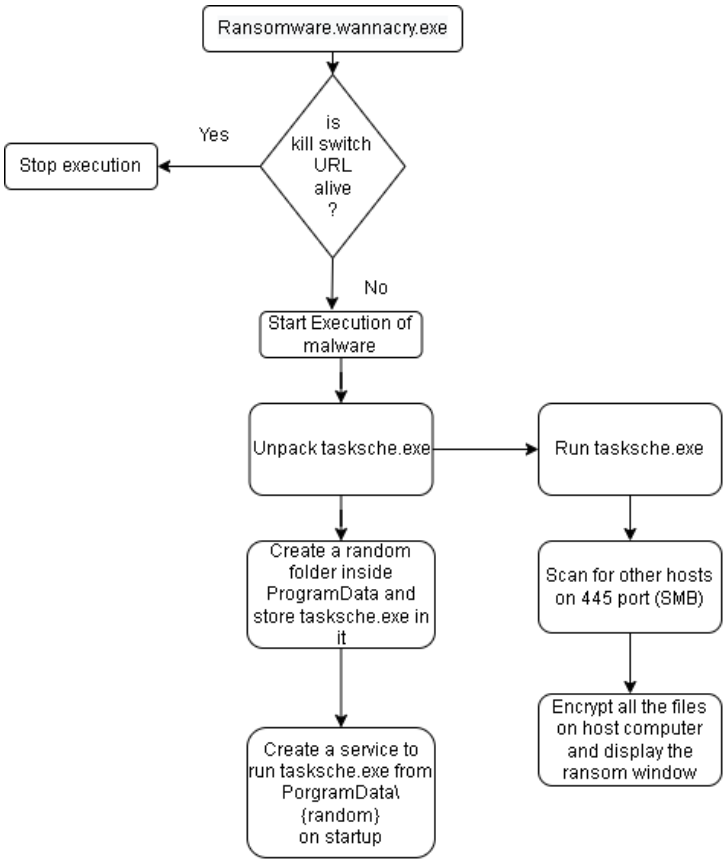
The WannaCry ransomware attack was a global epidemic that took place in May 2017. This ransomware attack spread through computers operating Microsoft Windows. User's files were held hostage, and a Bitcoin ransom was demanded for their return. Were it not for the continued use of outdated computer systems and poor education around the need to update software, the damage caused by this attack could have been avoided.

WannaCry is written in C++ language. On executing the malware it checks for a hardcoded URL, if it successfully pings that URL malware does not execute. If the URL was not found then malware execution takes place. Symptoms of the infection include ransomware payment window popup, encryption of the files, new desktop shortcuts and new services created. After executing the malware it creates a file named "C:\Windows\tasksche.exe" which contains the payloads, and then starts encrypting all the files on computer. WannaCry ransomware also tries to spread to other Windows Computers using the EternalBlue vulnerability.

YARA signature rules are attached in Rules & Signatures. Malware sample and hashes have been submitted to VirusTotal for further examination.

High-Level Technical Summary

WannaCry consists of two parts: stage 0 executable and an unpacked stage 2 encryption and worm program. It first attempts to contact its kill switch URL (hxxps://iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.local). If the URL is alive it does not execute. If the URL is not found then the malware unpacks tasksche.exe and creates a service to start tasksche.exe on startup. This executable encrypts all the files, shows the popup ransom window and changes the background of Desktop. It creates a random folder inside C:\ProgramData to store all the wannacry files. It exploits the EternalBlue vulnerability on port 445 to spread to other computers.



Malware Composition

WannaCry consists of the following components:

File Name	SHA256 Hash
Ranswomware.wannacry.exe	24D004A104D4D54034DBCFFC2A4B19A11F39008A575AA614EA04703480B1022C
tasksche.exe	ED01EBFBC9EB5BBEA545AF4D01BF5F1071661840480439C6E5BABE8E080E41AA

Ransomware.wannacry.exe

The initial executable that runs and checks the kill switch URL. If alive don't run else unpack tasksche.exe.

tasksche.exe:

This is used for persistence. It creates a random folder for wannacry staging area inside ProgramData. After execution of malware on host computer it tries to spread itself on other windows computers using SMB port 445. It starts encrypting all the files and after that it displays the ransomware popup and message.

Basic Static Analysis

Strings – Extracted using Floss

```
floss -n 6 Ransomware.wannacry.exe.malz > floss.txt
```

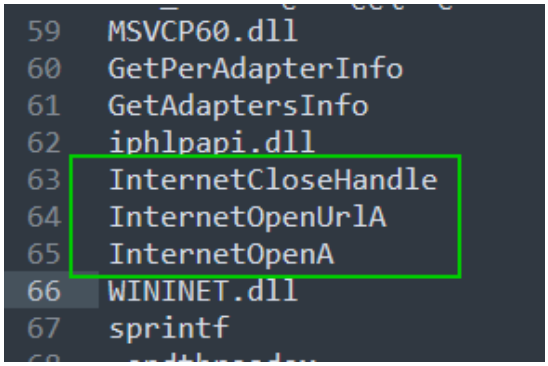


Fig 2: Modules used to open a URL

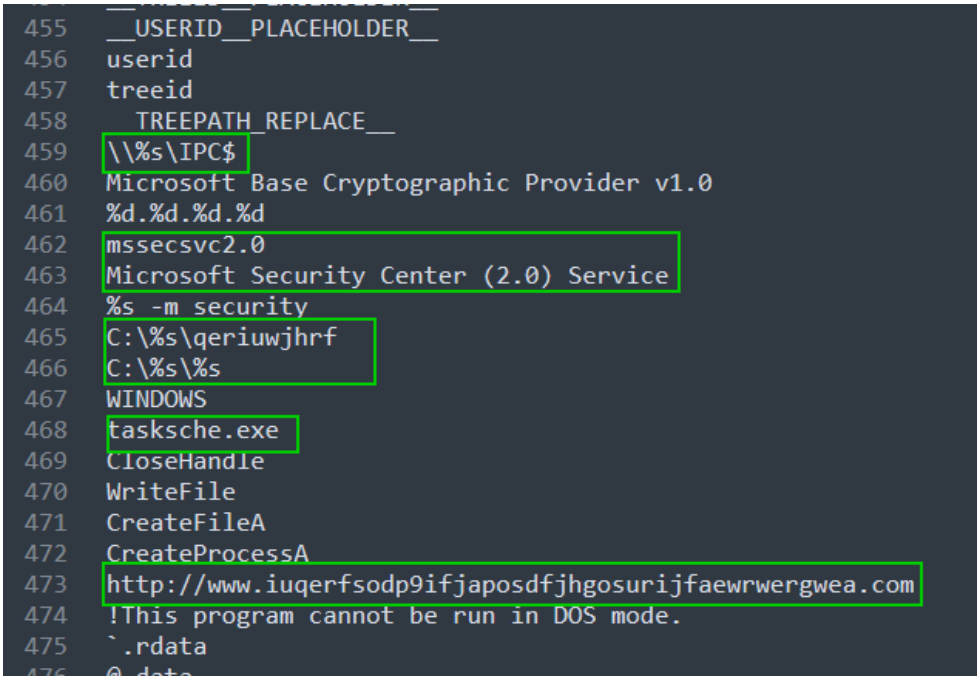


Fig 3: Service names used, Kill Switch URL and random paths

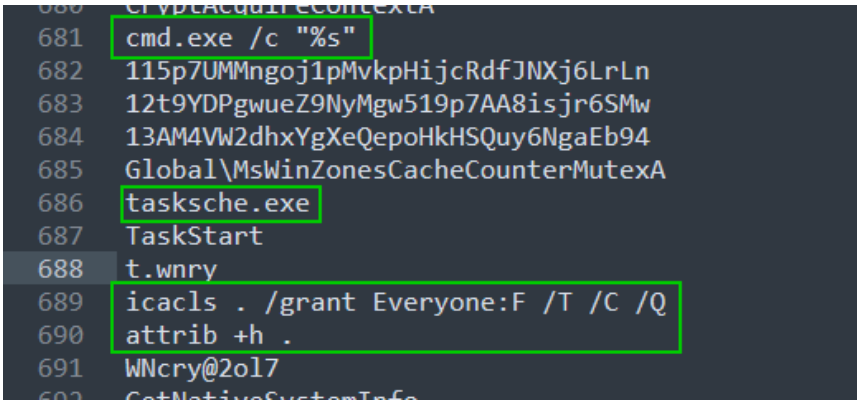
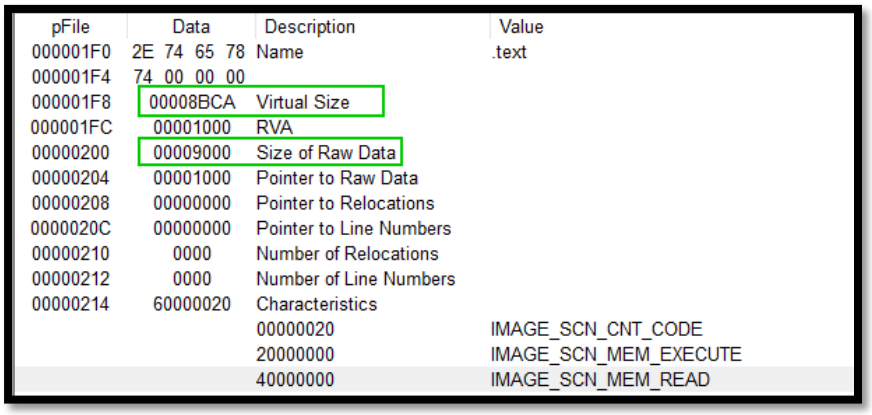


Fig 4: Service names used, random paths

icacls used for modifying access controls on files

attrib +h . used to hide the file attribute

PEview



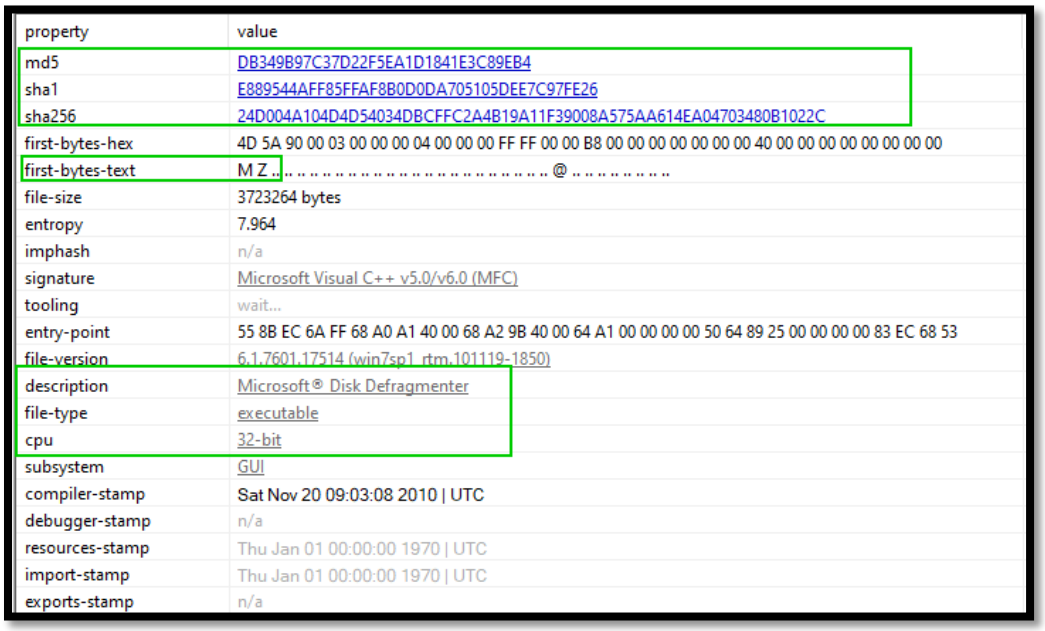
pFile	Data	Description	Value
000001F0	2E 74 65 78	Name	.text
000001F4	74 00 00 00		
000001F8	00008BCA	Virtual Size	
000001FC	00001000	RVA	
00000200	00009000	Size of Raw Data	
00000204	00001000	Pointer to Raw Data	
00000208	00000000	Pointer to Relocations	
0000020C	00000000	Pointer to Line Numbers	
00000210	0000	Number of Relocations	
00000212	0000	Number of Line Numbers	
00000214	60000020	Characteristics	
		00000020	IMAGE_SCN_CNT_CODE
		20000000	IMAGE_SCN_MEM_EXECUTE
		40000000	IMAGE_SCN_MEM_READ

Fig 5: IMAGE_SECTION_HEADER.text

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
0000A0B4	0000A73E	Hint/Name RVA	010B ??1_Lockit@std@@QAE@XZ
0000A0B8	0000A758	Hint/Name RVA	00A2 ??0_Lockit@std@@QAE@XZ
0000A0BC	00000000	End of Imports	MSVCP60.dll
0000A0C0	0000A932	Hint/Name RVA	0081 __set_app_type
0000A0C4	0000A98C	Hint/Name RVA	01C1 _stricmp
0000A0C8	0000A924	Hint/Name RVA	006F __p__fmode
0000A0CC	0000A914	Hint/Name RVA	006A __p__commode
0000A0D0	0000A944	Hint/Name RVA	00CA _except_handler3
0000A0D4	0000A8F0	Hint/Name RVA	0083 __setusermatherr
0000A0D8	0000A8E4	Hint/Name RVA	010F _initterm
0000A0DC	0000A8D4	Hint/Name RVA	0058 __getmainargs
0000A0E0	0000A8CA	Hint/Name RVA	008F _acmdln
0000A0E4	0000A904	Hint/Name RVA	009D _adjust_fdiv
0000A0E8	0000A958	Hint/Name RVA	00B7 _controlfp
0000A0EC	0000A8C2	Hint/Name RVA	0249 exit
0000A110	0000A81A	Hint/Name RVA	02C1 strncpy
0000A114	0000A81A	Hint/Name RVA	02C1 strncpy
0000A118	0000A824	Hint/Name RVA	02A6 rand
0000A11C	0000A82C	Hint/Name RVA	00A6 _beginthreadex
0000A120	0000A83E	Hint/Name RVA	0049 __CxxFrameHandler
0000A124	0000A852	Hint/Name RVA	02B4 srand
0000A128	0000A85A	Hint/Name RVA	02D0 time
0000A12C	0000A862	Hint/Name RVA	0062 __p___argc
0000A130	00000000	End of Imports	MSVCRT.dll
0000A134	0000A7DC	Hint/Name RVA	0092 InternetOpenA
0000A138	0000A7C8	Hint/Name RVA	0093 InternetOpenUrlA
0000A13C	0000A7B2	Hint/Name RVA	0069 InternetCloseHandle
0000A140	00000000	End of Imports	WININET.dll
0000A144	80000003	Ordinal	0003
0000A148	80000010	Ordinal	0010
0000A14C	80000013	Ordinal	0013
0000A150	80000008	Ordinal	0008
0000A154	8000000E	Ordinal	000E
0000A158	80000073	Ordinal	0073
0000A15C	8000000C	Ordinal	000C
0000A160	8000000A	Ordinal	000A
0000A164	80000012	Ordinal	0012
0000A168	80000009	Ordinal	0009
0000A16C	80000017	Ordinal	0017
0000A170	80000004	Ordinal	0004
0000A174	8000000B	Ordinal	000B
0000A178	00000000	End of Imports	WS2_32.dll
0000A17C	0000A792	Hint/Name RVA	001C GetAdaptersInfo
0000A180	0000A77E	Hint/Name RVA	0040 GetPerAdapterInfo
0000A184	00000000	End of Imports	iphlpapi.dll

Fig 6: Import Address Table

PEStudio



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++ v5.0/v6.0 (MFC)
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	Thu Jan 01 00:00:00 1970 | UTC
import-stamp	Thu Jan 01 00:00:00 1970 | UTC
exports-stamp	n/a

Fig 7: Basic Information about the executable

Basic Dynamic Analysis

Analysis with inetsim turned on

When the malware is executed with inetsim turned on, the malware does not execute. It tries to connect to “hxxp://www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com”. On successful connection it does not infect the system.

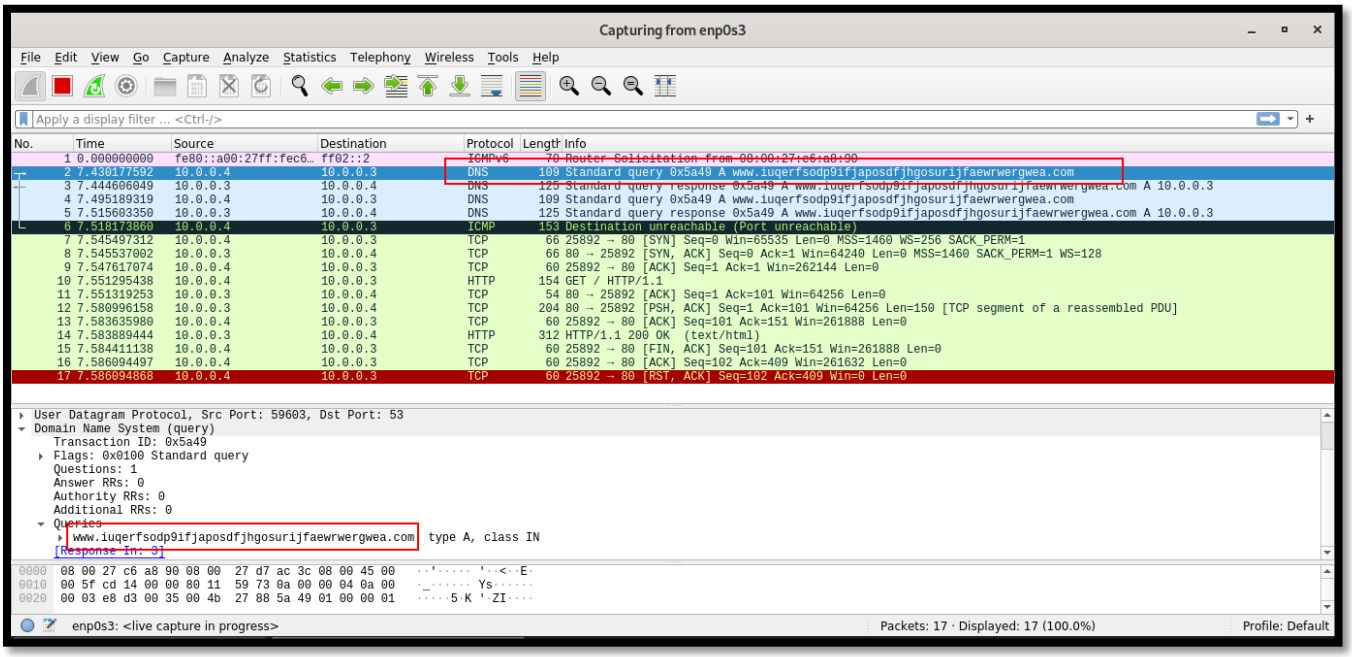


Fig 8: Network traffic when malware is executed

Analysis with inetsim turned off

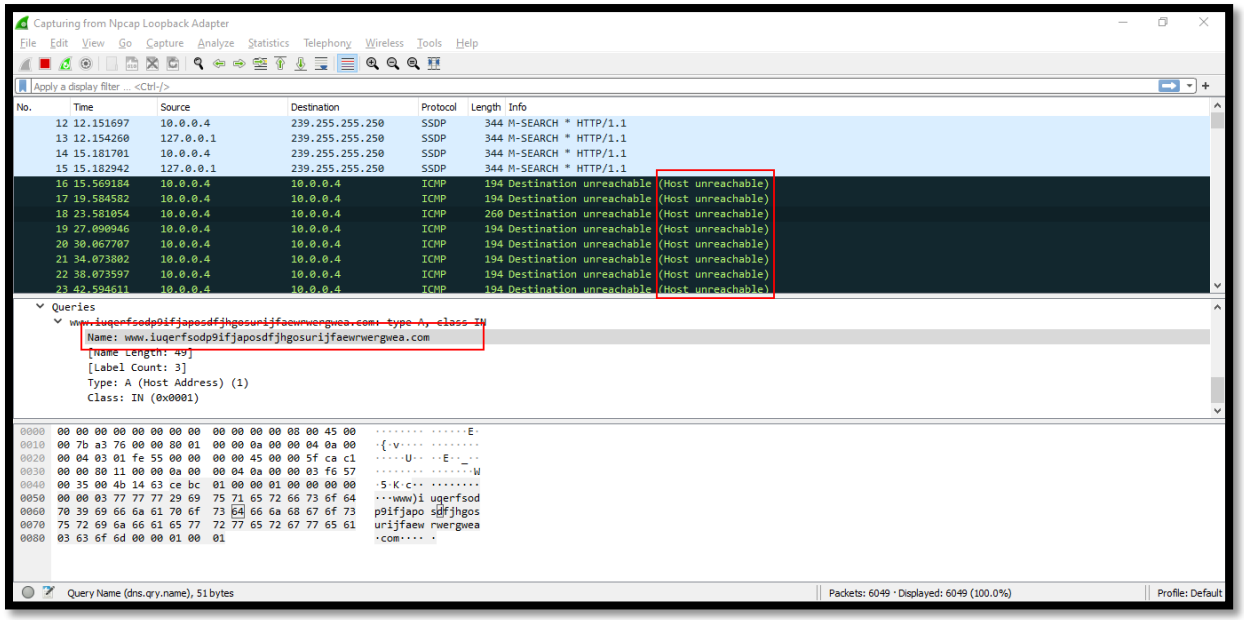


Fig 9: Network traffic when malware is executed. The requests are unreachable because inetsim is turned off

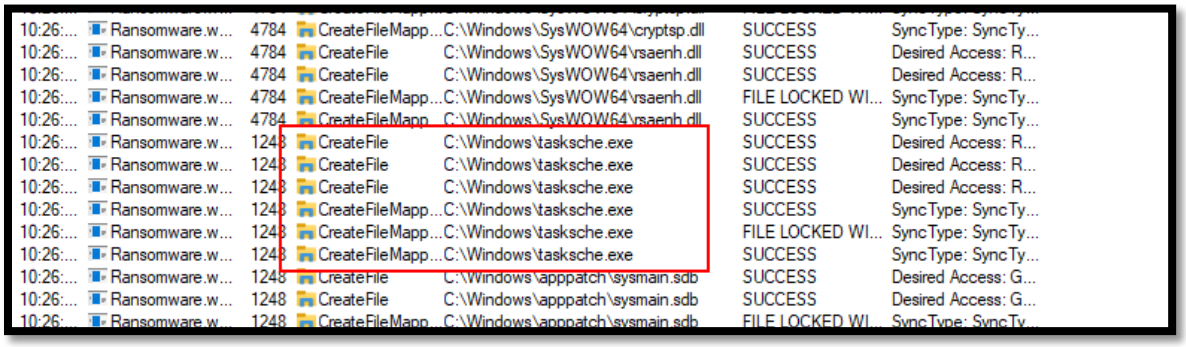


Fig 10: Procmon analysis. Creation of tasksche.exe file

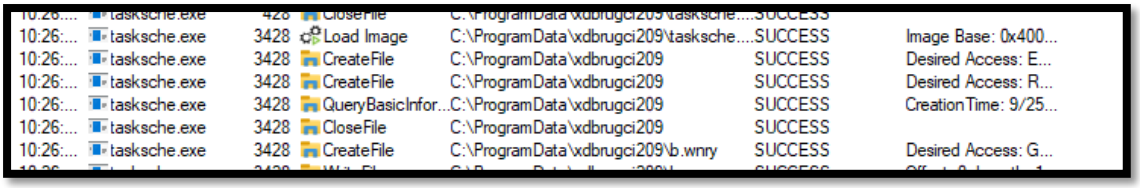


Fig 11: Wannacry creates tasksche.exe and executes it. Tasksche.exe creates a file with a random name in C:\ProgramData\{random name}. This folder is a staging area for wannacry ransomware

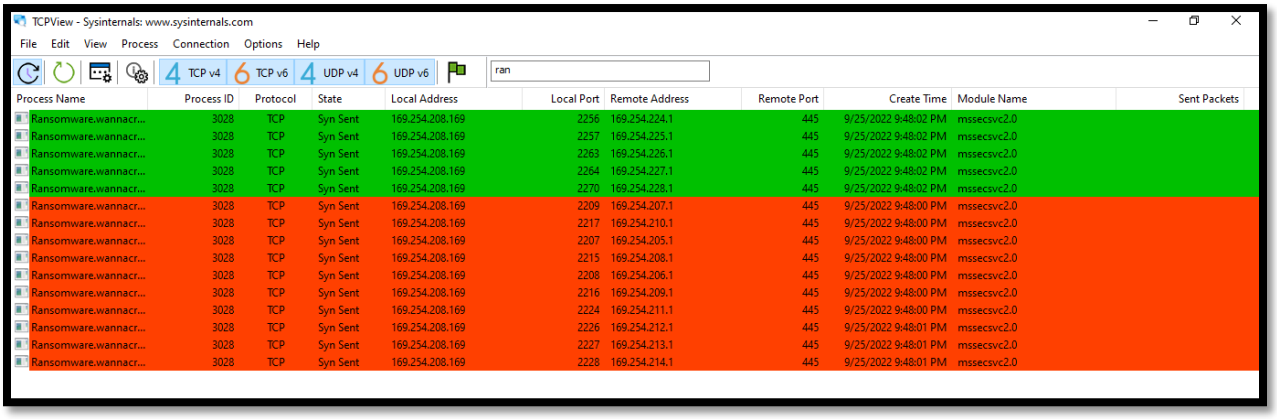


Fig 12: Tasksche.exe tries to locate and infect computers using port 445 (SMB)

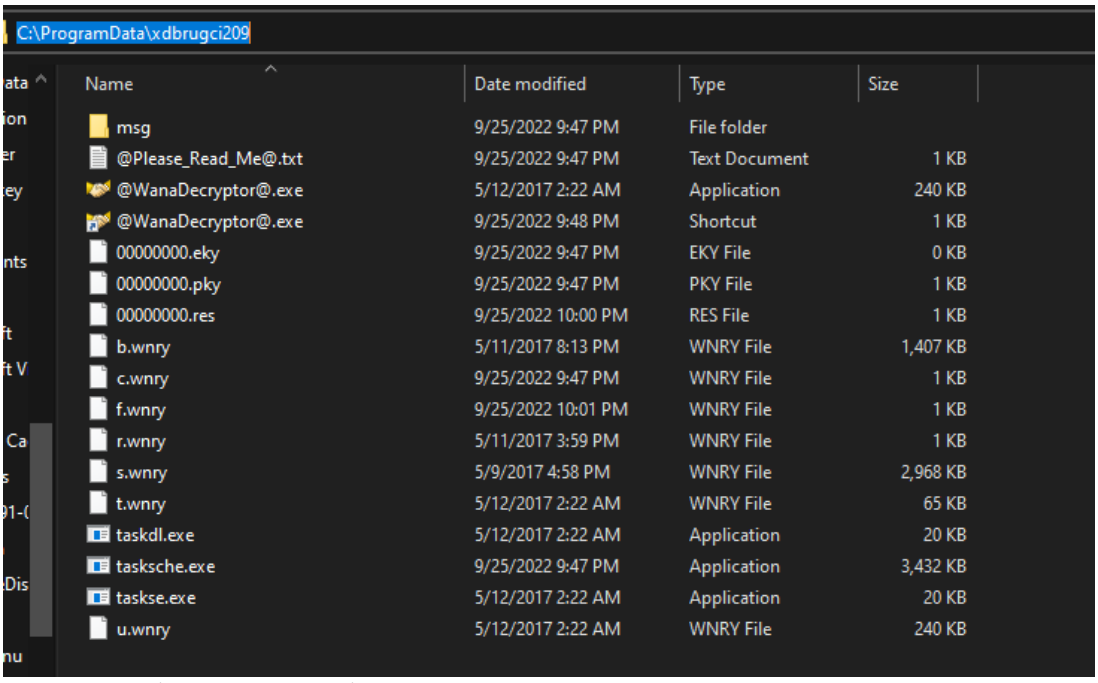


Fig 13: C:\ProgramData\{random name} folder which is staging area for wannacry

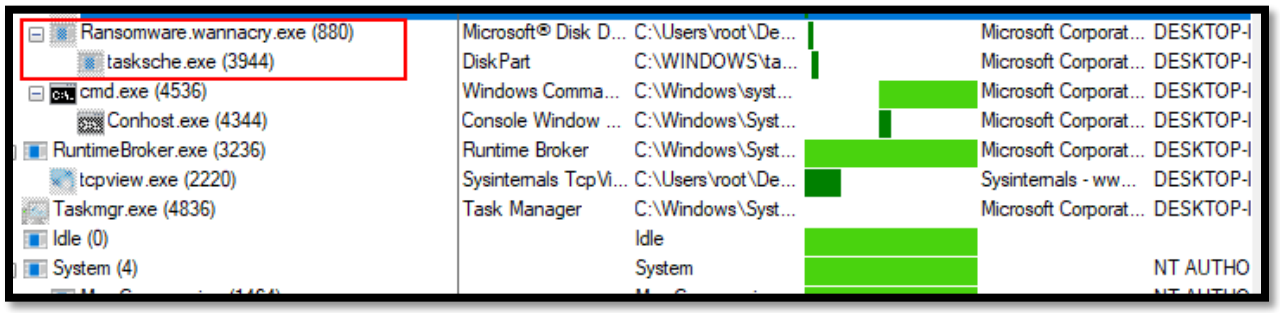


Fig 14: Procmon process tree

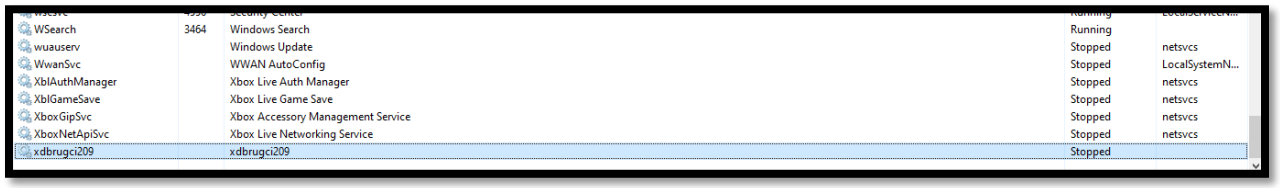


Fig 15: Task Manager. Service name is same as the random file name created by tasksche.exe

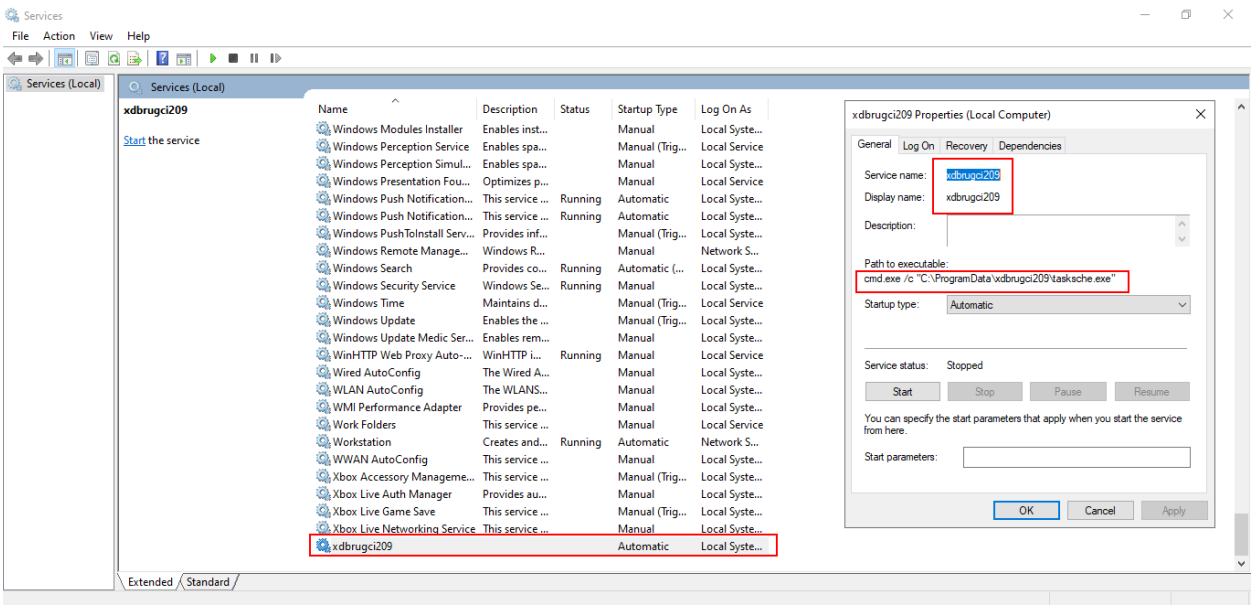


Fig 16: Service. Service name is same as the random file name created by tasksche.exe. This service just invokes the tasksche.exe command on startup.

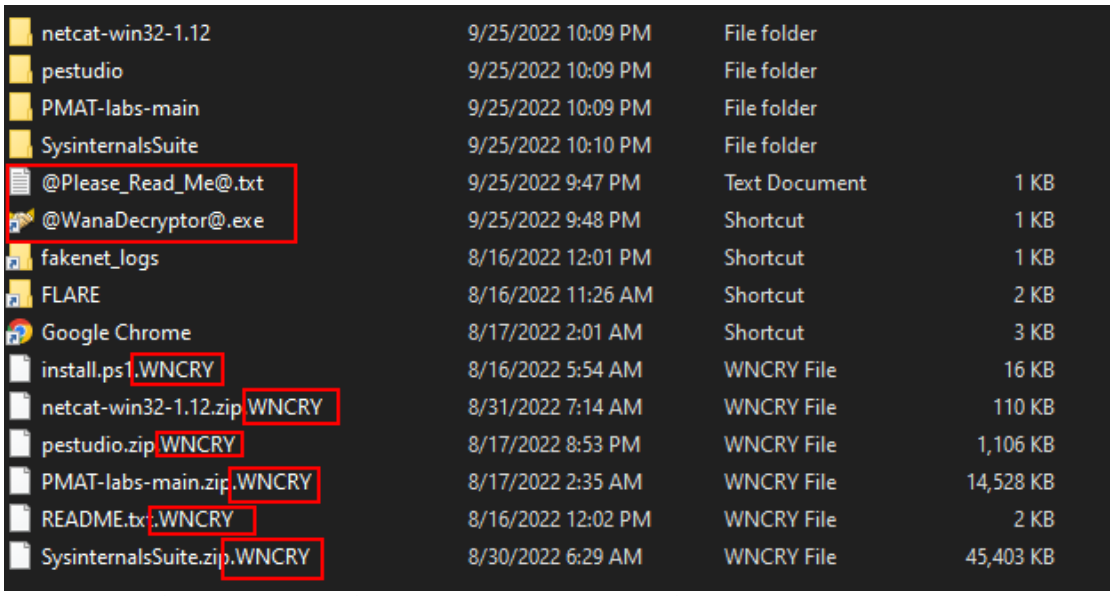


Fig 17: New files added and old files are encrypted.

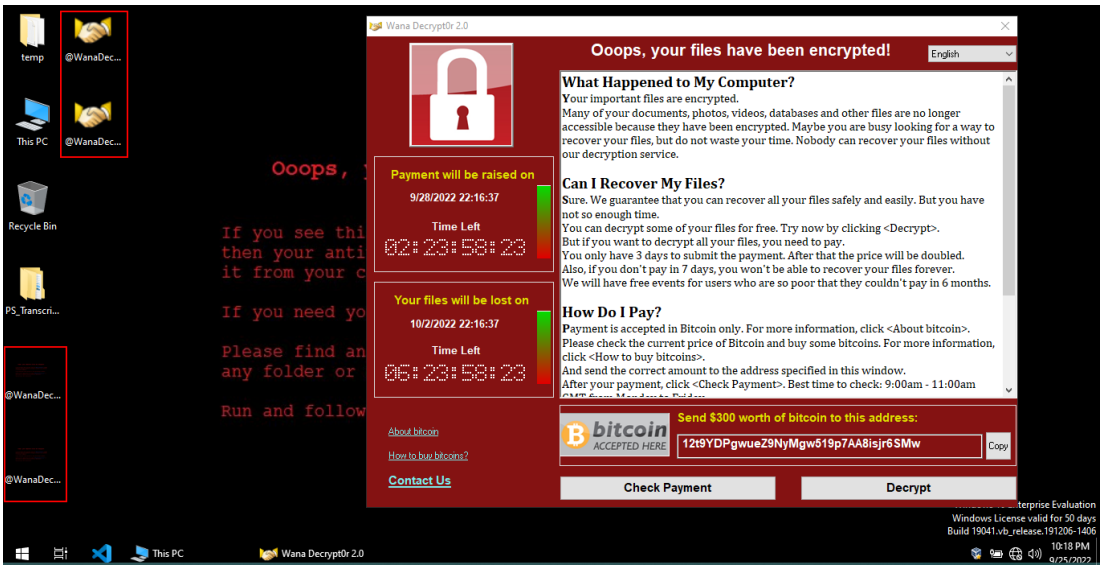


Fig 18: After Infection. New desktop icons and ransom payment popup

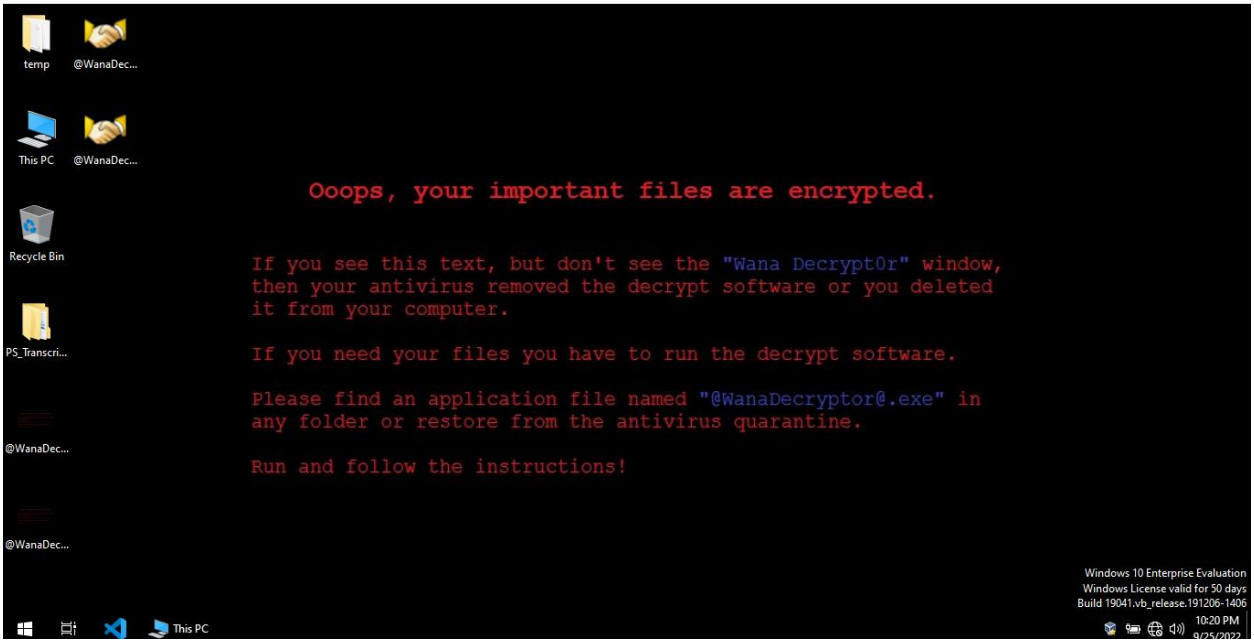


Fig 19: After Infection. Ransom message

Advanced Static Analysis

Cutter

Fig 20: Main function viewed inside cutter graph mode

Advanced Dynamic Analysis

X32dbg

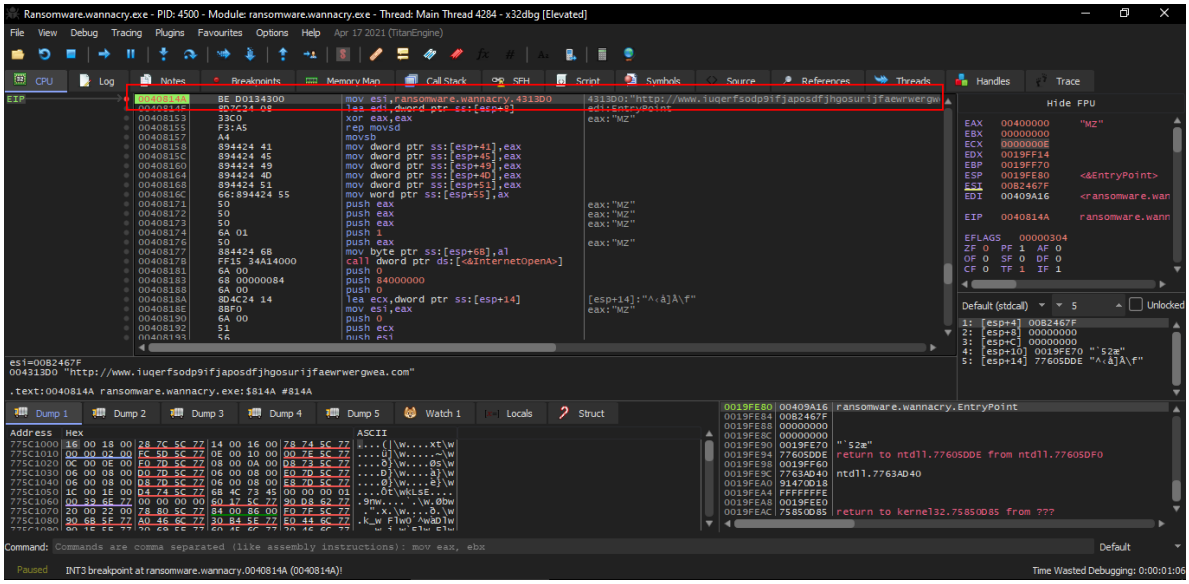


Fig 21: Set a breakpoint on kill switch URL

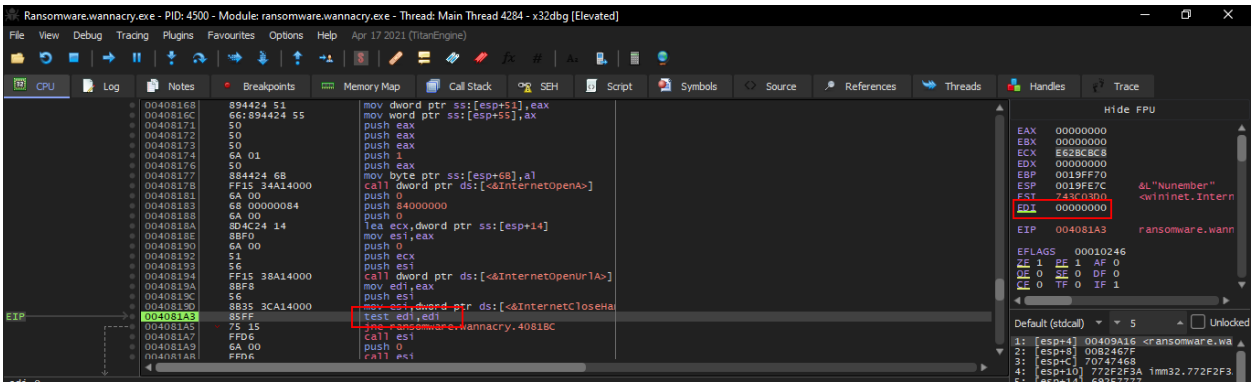


Fig 22: The kill switch URL was not found therefore the EDI has value 0

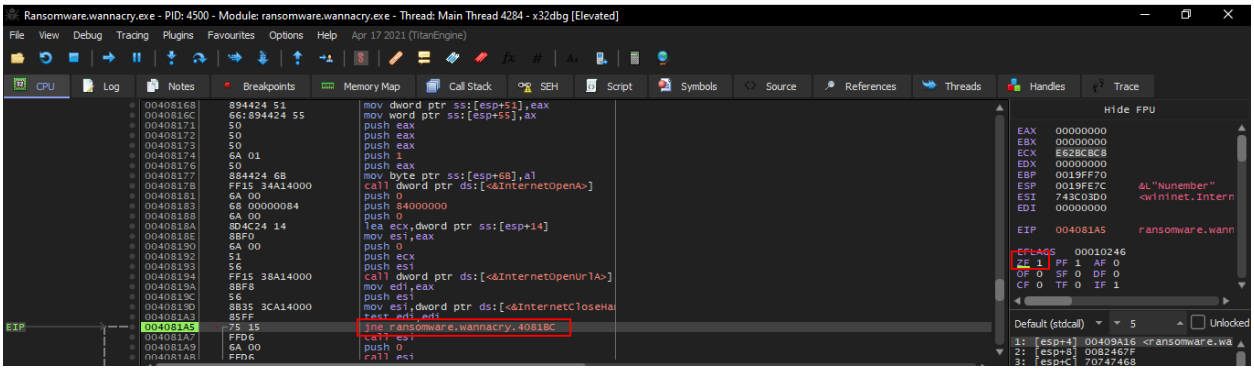


Fig 23: The zero flag is evaluated to 1 but we change it to 0

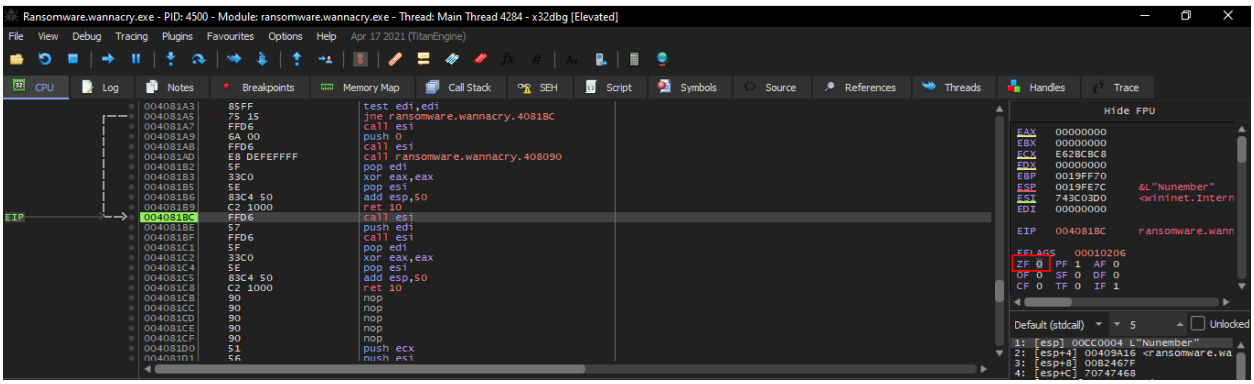


Fig 24: Changing the zero flag to 0. This makes the program to take the jump call and the malware is not executed.

Indicators of Compromise

Network Indicators

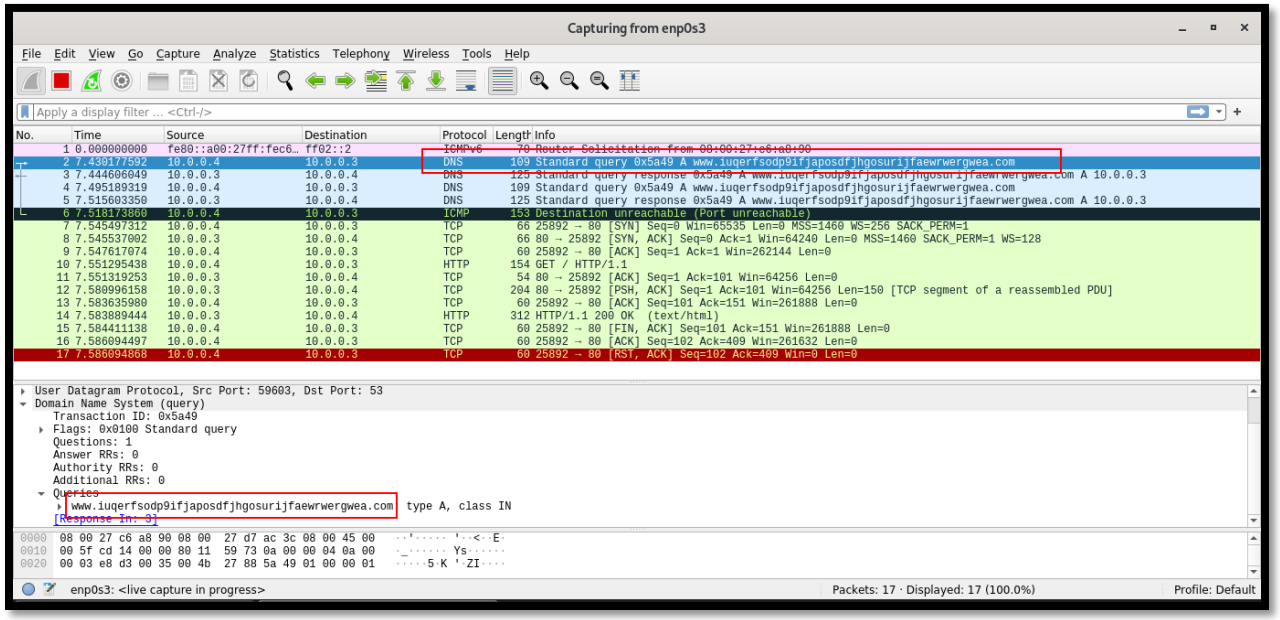


Fig 25: Initial connection to kill switch URL.

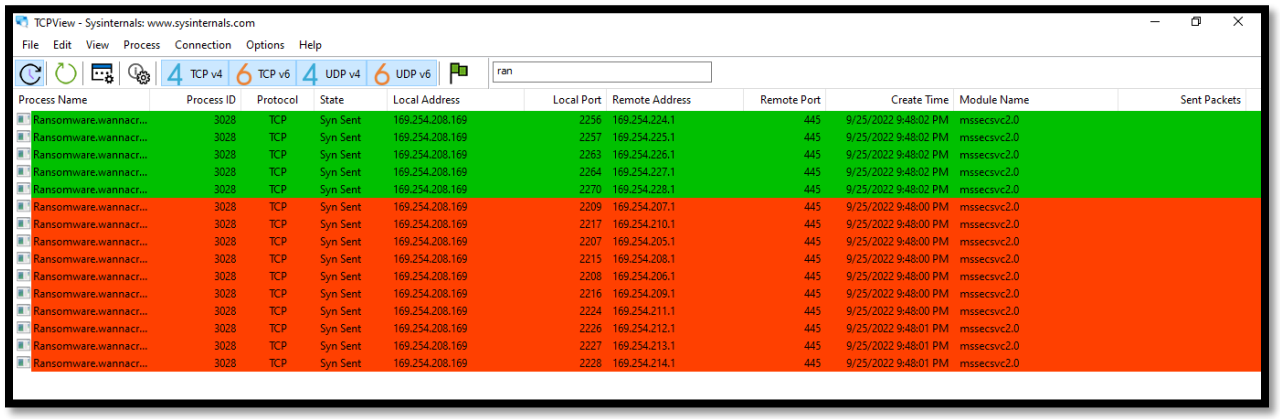


Fig 26: Locating other machines and exploiting them using 445 port (SMB)

Host-based Indicators

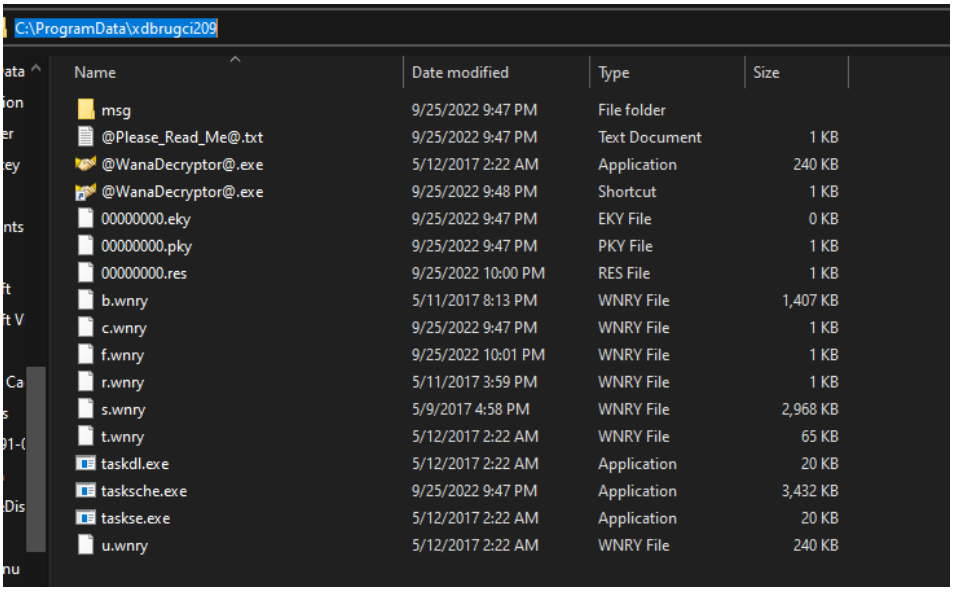


Fig 27: random folder present inside C:\ProgramData which contains tasksche.exe. This exe is executed on startup.

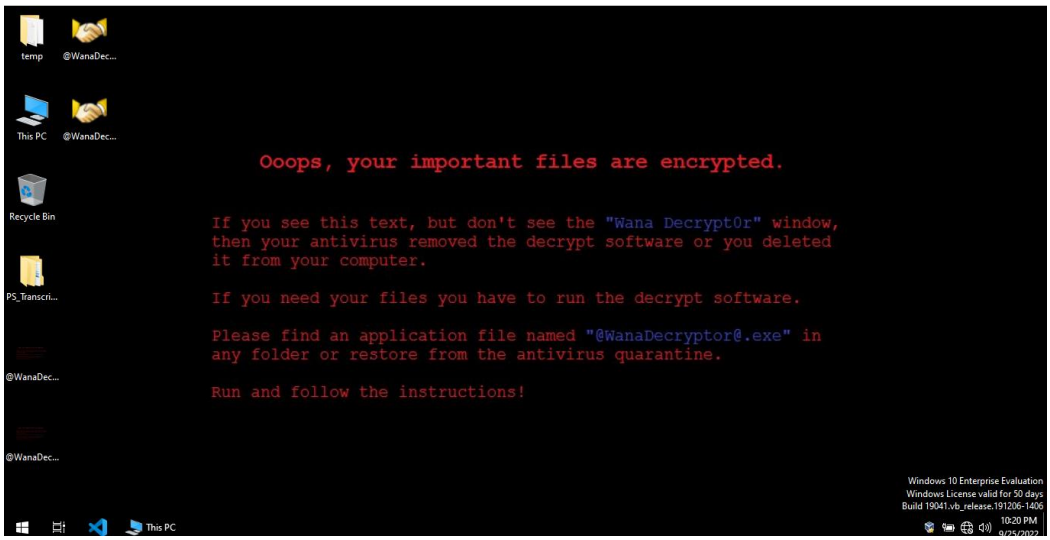


Fig 28: @WanaDecryptor@.bpm, @WanaDecryptor@.exe present on User's desktop

Rules & Signatures

YARA Rule

```
rule Ransomware_WannaCry {

  meta:
    last_updated = "2022-09-26"
    author = "rishank-shah"
    description = "Yara rule for WannaCry Ransomware"

  strings:
    $string1 = "attrib +h ." fullword ascii
    $string2 = "icacls . /grant Everyone:F /T /C /Q"  fullword ascii
    $string3 = "C:\\%s\\qeriuwjhrf" fullword ascii
    $string4 = "WNcry@2ol7" fullword ascii
    $string5 = "wnry" ascii
    $url = "www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com" ascii
    $payload = "tasksche.exe" ascii
    $PE_magic_byte = "MZ"

  condition:
    $PE_magic_byte at 0 and
    ($url or 1 of ($string*) or $payload)
}
```