WannaCry: A Ransomware Cryptoworm Unleashed in 2017

WannaCry, a ransomware cryptoworm, gained notoriety through its involvement in the WannaCry ransomware attack that occurred in 2017. This destructive cyber threat exploited a widely-known vulnerability known as "EternalBlue," originally developed by the National Security Agency (NSA). Notably, WannaCry is a 32-bit program coded in C++ designed specifically for the Windows operating system.

The WannaCry attack unfolds in three distinctive stages:

Stage 1: Deployment of a Secondary Executable

In the initial phase, the malware replaces the "tasksche.exe" file within the "C:\Windows" directory with a secondary executable.

Stage 2: Actions of the Secondary Executable

Upon execution, the second executable carries out critical actions, including the deployment of essential encryption resources like DLL and EXE files, cryptographic keys, and Bitcoin addresses, all integral components of the ransom process.

Stage 3: Encryption and File Manipulation

WannaCry initiates multiple threads within the victim's machine to systematically encrypt files, a pivotal component of its nefarious intent.

Common Symptoms of Infection Include:

- A change in the desktop background to a black background with red text.
- The encryption of files, identified by the "WNCRY" extension.
- The presence of a service labeled "mssecsvc.exe" with a display name of "Microsoft Security Center (2.0) Service."
- The existence of a registry key at "HKLM\SOFTWARE\Wow6432Node\WannaCrypt."

High-Level Technical Summary - WannaCry Ransomware

WannaCry ransomware operates in two distinct phases, each characterized by specific actions and behaviors:

Phase 1: Dropper and Kill Switch Check

In the initial phase, WannaCry begins by attempting to connect to a suspicious URL: hxxp[://]iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea[dot]com. This URL serves a dual purpose as a kill switch. If the domain responds to the request, WannaCry promptly terminates its execution. However, if the domain remains unresponsive, the malware continues its operation.

Following this determination, WannaCry proceeds to create a Windows service named "mssecsvc2.0" with a display name of "Microsoft Security Center (2.0) Service." The service is configured to execute the binary located at "<PATH_TO_WANNACRY>\wannacry.exe -m security." Concurrently, WannaCry embarks on its propagation phase, attempting to connect to a wide range of IPv4 addresses.

Phase 2: Payload Unpacking and Encryption

In the second phase, WannaCry unpacks the payload located at "C:\Windows\tasksche.exe /i" from the initial dropper. This phase focuses on establishing persistence mechanisms. To achieve this, WannaCry creates a folder within "C:\ProgramData" using a dynamically generated string as its name. The malware then proceeds to copy itself to the "C:\Windows" directory, naming the file "tasksche.exe."

Subsequently, a service is created, mirroring the name of the previously generated string for the folder, and it references the payload located at "C:\ProgramData<GENERATED_STRING>\tasksche.exe." Once this service is in place and the payload executes, the encryption process commences. This process encompasses changes to the desktop background, the delivery of decryption instructions, and the periodic appearance of a GUI application, reemerging every 60 seconds if closed.

1. **File Name: wannacry.exe**
    - SHA-1 Hash: E889544AFF85FFAF8B0D0DA705105DEE7C97FE26
2. **File Name: tasksche.exe**
    - SHA-1 Hash: 5FF465AFAABCBF0150D1A3AB2C2E74F3A4426467
    - SHA-256 Hash: ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa
3. **File Name: tasksche_res.zip**
    - SHA-1 Hash: 30F8820CF93A627C66195F0D77D6A409024C6E52
4. **File Name: taskdll.exe**
    - SHA-1 Hash: 47A9AD4125B6BD7C55E4E7DA251E23F089407B8F
5. **File Name: taskse.exe**
    - SHA-1 Hash: BE5D6279874DA315E3080B06083757AAD9B32C23

Additionally, here are the SHA-256 hashes for some notable files associated with WannaCry:

6. **File Name: Ransomware.wannacry.exe**
    - SHA-256 Hash: 24d004a104d4d54034dbcffc2a4b19a11f39008a575aa614ea04703480b1022c
7. **File Name: @WanaDecryptor@[.]exe**
    - SHA-256 Hash: b9c5d4339809e0ad9a00d4d3dd26fdf44a32819a54abf846bb9b560d81391c25
8. **File Name: taskdl.exe**
    - SHA-256 Hash: 4a468603fdcb7a2eb5770705898cf9ef37aade532a7964642ecd705a74794b79
9. **File Name: taskhsvc.exe**
    - SHA-256 Hash: e48673680746fbe027e8982f62a83c298d6fb46ad9243de8e79b7e5a24dcd4eb

Wannacry.exe :

The initial executable that runs in the beginning

Tasksche.exe :

This executable is dropped by wannacry.exe file after execution. It is responsible for creating a
directory in the C:\ProgramData directory and copies itself into it in order to drop more files.

Tasksche_res.zip :

Resides in the resource section of tasksche.exe file which contains executable and files for encryption

taskdll.exe and taskse.exe :

These executable are responsible for encrypting files in the local system.

Static Analysis

Basic information about the executable :

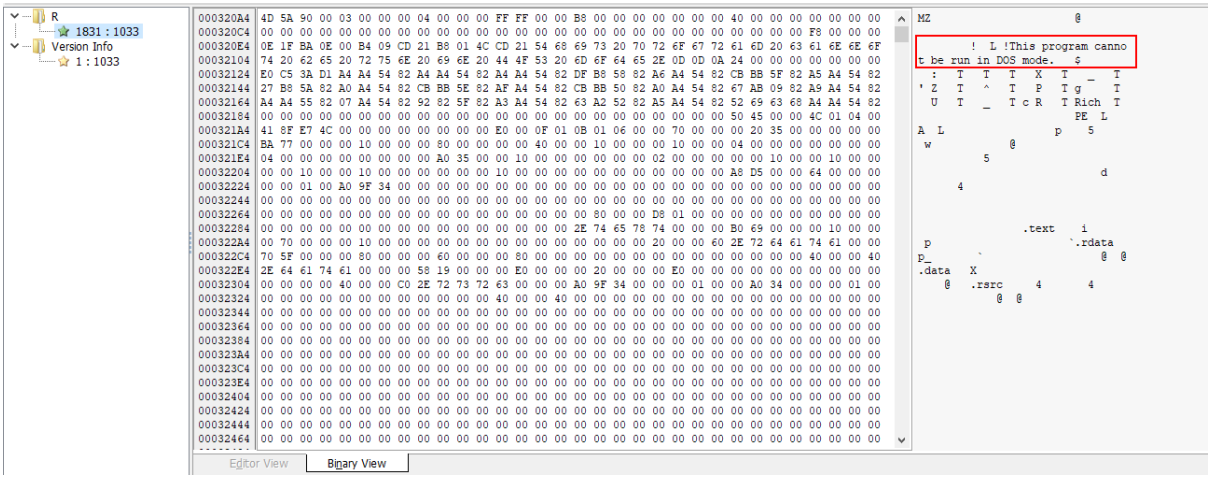
The original name can be found along with ***MD5*** and ***SHA1*** signatures of the binary in the 'version' tab in PE Studio:

Property	Value
File Name	C:\Users\Analyst\Desktop\wannacry.exe
File Type	Portable Executable 32
File Info	Microsoft Visual C++ 6.0
File Size	3.55 MB (3723264 bytes)
PE Size	3.55 MB (3723264 bytes)
Created	Thursday 24 February 2022, 11.13.31
Modified	Tuesday 19 March 2019, 11.32.14
Accessed	Monday 28 February 2022, 10.53.58
MD5	DB349B97C37D22F5EA1D1841E3C89EB4
SHA-1	E889544AFF85FFAF8B0D0DA705105DEE7C97FE26

Property	Value
CompanyName	Microsoft Corporation
FileDescription	Microsoft® Disk Defragmenter
FileVersion	6.1.7601.17514 (win7sp1_rtm.101119-1850)
InternalName	lhdfrgui.exe
LegalCopyright	© Microsoft Corporation. All rights reserved.
OriginalFilename	lhdfrgui.exe
ProductName	Microsoft® Windows® Operating System

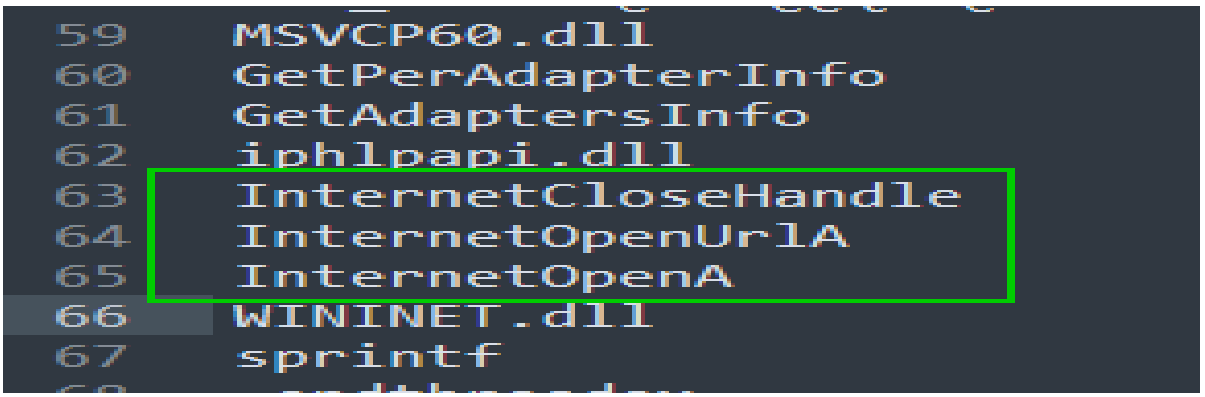
You can see the string **"This program cannot run in dos mode"** in the below image. This string shows in the DOS Stub Header of the PE file. We can double confirm it is a **.exe** file.

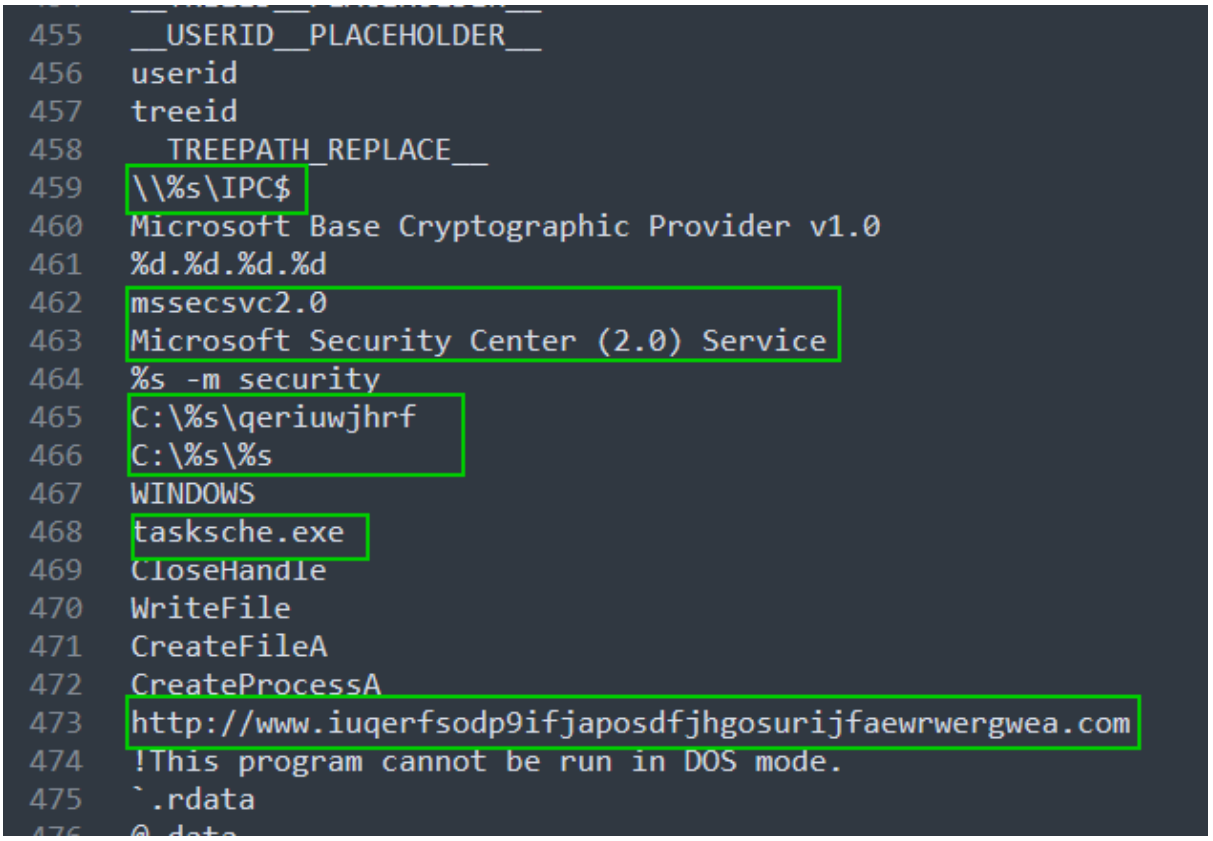
And you can see some encoded strings in the given image below, so we confirm it is packed.

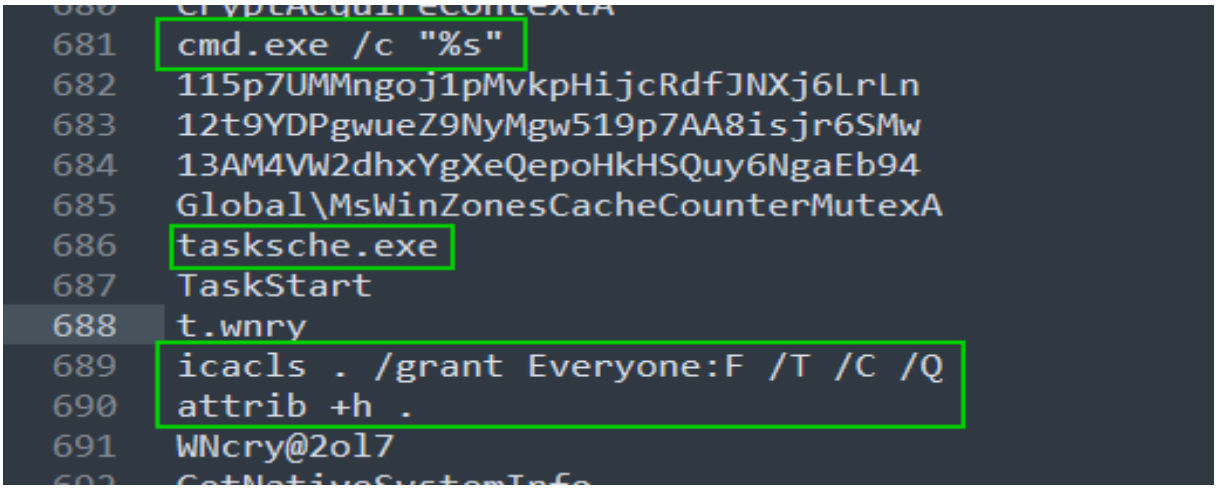


It shows the binary view of wannacry ransomware and also you can see the **ASCII** column, it shows **"This program cannot be run in DOS mode"**, and **"PE"** strings, so we confirm it is an executable file.

Some interesting strings inside the binary using floss :







Some interesting import functions API are :

DOCUMENTATIONS :

1.) https://docs.microsoft.com/en-us/windows/win32/api/wininet/nf-wininet-internetopenurla
2.) https://docs.microsoft.com/en-us/windows/win32/api/wininet/nf-wininet-internetopenurla
3.) https://docs.microsoft.com/en-us/windows/win32/api/wininet/nf-wininet-internetclosehandle
4.) https://docs.microsoft.com/en-us/windows/win32/api/processthreadsapi/nf-processthreadsapi-terminatethread
5.) https://docs.microsoft.com/en-us/windows/win32/api/libloaderapi/nf-libloaderapi-loadresource
6.) https://docs.microsoft.com/en-us/windows/win32/api/winbase/nf-winbase-findresourcea
7.) https://docs.microsoft.com/en-us/windows/win32/api/libloaderapi/nf-libloaderapi-getprocaddress
8.) https://docs.microsoft.com/en-us/windows/win32/api/libloaderapi/nf-libloaderapi-getmodulehandlea
9.) https://learn.microsoft.com/en-us/windows/win32/api/processthreadsapi/nf-processthreadsapi-exitprocess
10.) https://docs.microsoft.com/en-us/windows/win32/api/winsvc/nf-winsvc-startservicectrldispatcherahttps://docs.microsoft.com/en-us/windows/win32/api/winsvc/nf-winsvc-startservicectrldispatchera
11.) https://learn.microsoft.com/en-us/windows/win32/api/winsvc/nf-winsvc-registerservicectrlhandlera
12.) https://learn.microsoft.com/en-us/windows/win32/api/winsvc/nf-winsvc-changeserviceconfig2a
13.) https://learn.microsoft.com/en-us/windows/win32/api/winsvc/nf-winsvc-setservicestatus
14.) https://docs.microsoft.com/en-us/windows/win32/api/winsvc/nf-winsvc-openscmanagera
15.) https://docs.microsoft.com/en-us/windows/win32/api/winsvc/nf-winsvc-createservicea
16.) https://learn.microsoft.com/en-us/windows/win32/api/winsvc/nf-winsvc-closeservicehandle

Call via	Name	Ordinal	Original Thunk	Thunk	Forwarder	Hint
A134	InternetOpenA	-	A7DC	A7DC	-	92
A138	InternetOpenUrlA	-	A7C8	A7C8	-	93
A13C	InternetCloseHandle	-	A7B2	A7B2	-	69

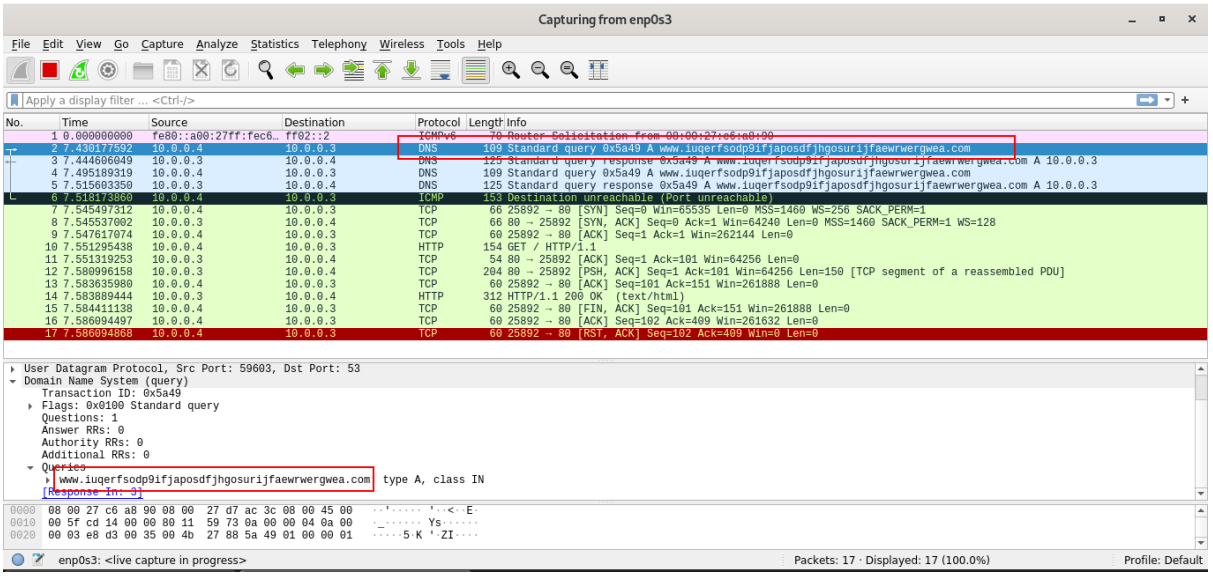
Call via	Name	Ordinal	Original Thunk	Thunk	Forwarder	Hint
A000	StartServiceCtrlDispatcherA	-	A6F6	A6F6	-	24A
A004	RegisterServiceCtrlHandlerA	-	A6D8	A6D8	-	20C
A008	ChangeServiceConfig2A	-	A6C0	A6C0	-	34
A00C	SetServiceStatus	-	A6AC	A6AC	-	244
A010	OpenSCManagerA	-	A69A	A69A	-	1AD
A014	CreateServiceA	-	A688	A688	-	64
A018	CloseServiceHandle	-	A672	A672	-	3E

Call via	Name	Ordinal	Original Thunk	Thunk	Forwarder	Hint
A054	TerminateThread	-	A4E4	A4E4	-	35F
A058	LoadResource	-	A5A6	A5A6	-	257
A05C	FindResourceA	-	A5B6	A5B6	-	E3
A060	GetProcAddress	-	A5C6	A5C6	-	1A0
A064	GetModuleHandleW	-	A5D8	A5D8	-	182
A068	ExitProcess	-	A5EC	A5EC	-	B9
A06C	GetModuleFileNameA	-	A5FA	A5FA	-	17D

Basic Dynamic Analysis :

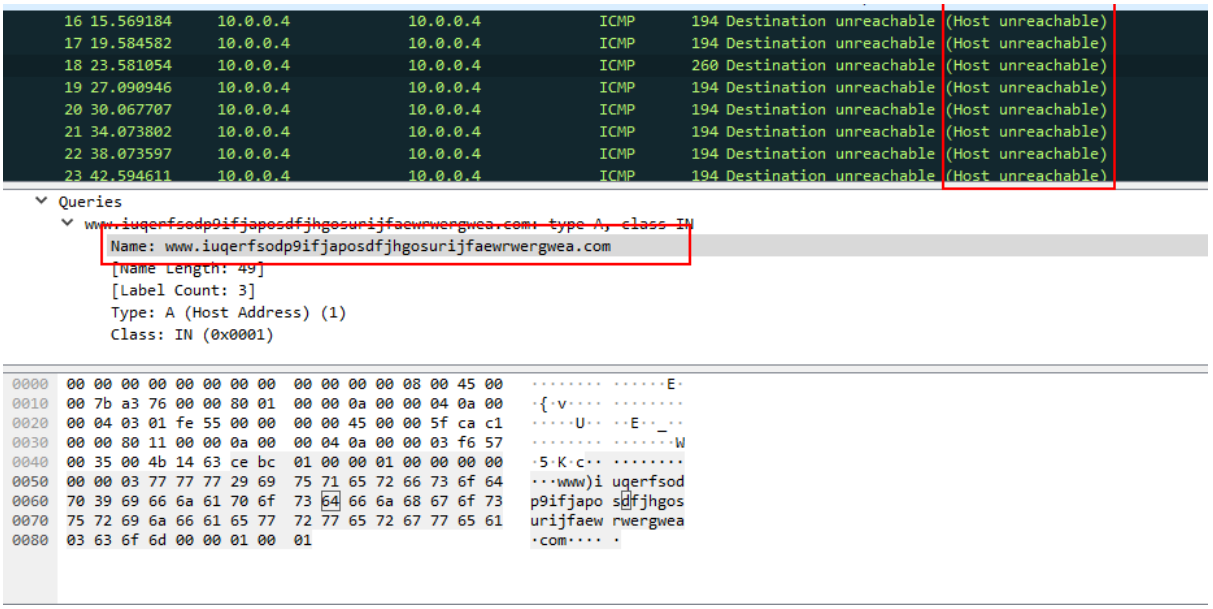
Analyzing the network using inetsim :

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.

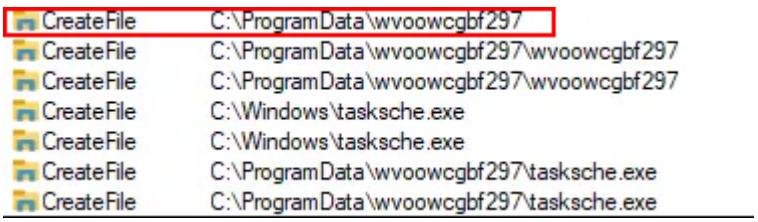


Then analyzing without inetsim :

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



Analysing the file creation using procmon service :



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

The service name will be same as the random filename : xncldfcnvvj305 tasksche.exe. This service just invokes the tasksche.exe command on startup.

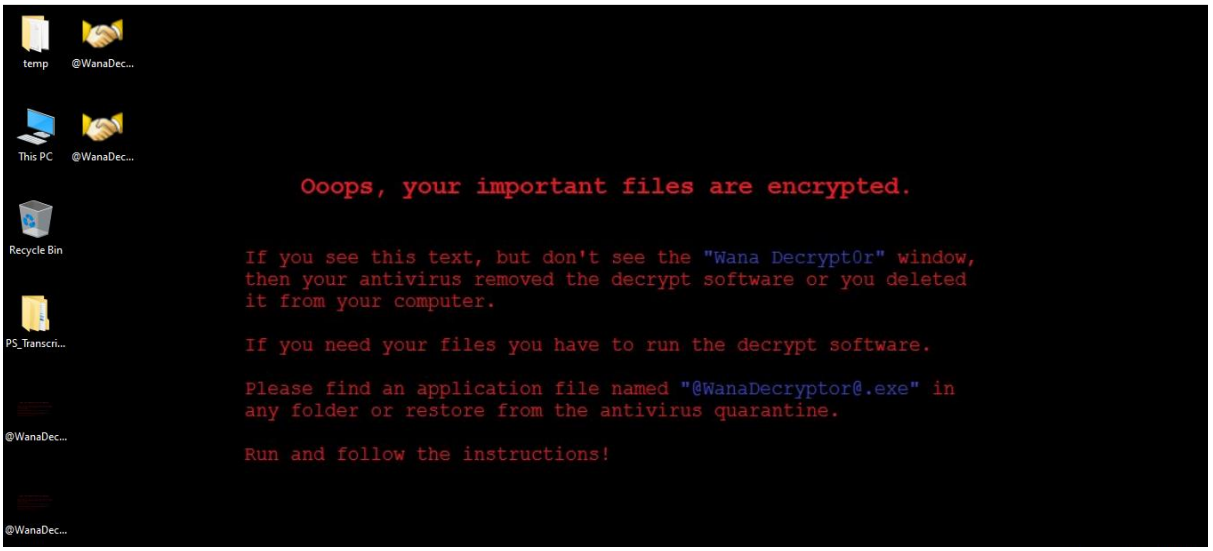
*

After this step the ransomwares new files is added and the old files will be encrypted using .WNCRY → WannaCry

*

After the infection it changes the icons of the victims files and desktop background or wallpaper will be changed and the ransom payment popup

The ransom message to threaten will be in the wallpaper or background of the desktop



While analysing the taskche.exe using cutter :