



# MALWARE ANALYSIS

## WANNACRY

JAN 2023 | VZSECURE

---

## Content:

1. Execution Summary
  2. Getting Hashes
  3. Analyzing Hash
  4. Static Analysis
  5. Dynamic Analysis
  6. Overview of Basic Analysis
  7. Advance Static Analysis
  8. Advance Dynamic Analysis
  9. Yara Rules
  10. Exit
-

## 1. Execution Summary

When any user tries to download the file and tries to run in his local machine then in backend the virus executes and encrypt all the files present in the system. It also changes the wallpaper of the machine and pop-up small window for paying ransom to attacker. When we try to access the content of the file, it was completely encrypted format.

So, without wasting time let analyze the virus and show you what happening in the backend of the virus execution.



## 2. Getting Hashes

Getting the hash is important part of static analysis. With the help of the hash, we can get the hint that what the virus is suppose to do in real case scenario.

We will use floss to get the hash of the executable program. We will need either MD5 or SHA256 hash.

SHA256:

24d004a104d4d54034dbcffc2a4b19a11f39008a575aa614ea04703480b1022c

MD5: db349b97c37d22f5ea1d1841e3c89eb4

```
C:\Users\hacker\Desktop
λ sha256sum.exe Ransomware.wannacry.exe.malz
24d004a104d4d54034dbcffc2a4b19a11f39008a575aa614ea04703480b1022c *Ransomware.wannacry.exe.malz

C:\Users\hacker\Desktop
λ md5sum.exe Ransomware.wannacry.exe.malz
db349b97c37d22f5ea1d1841e3c89eb4 *Ransomware.wannacry.exe.malz
```

### 3. Analyzing Hash

After getting the hash we need to check that hash. So, we will take the help of virus total which will check if the hash matches any virus which was happens in past real case scenario. Then virus total has over 70 antivirus it will analyze the hash through antivirus and give us the result.

We can see that over 66 antiviruses have detected the hash for malicious.

66 / 71

66 security vendors and 5 sandboxes flagged this file as malicious

24d004a104d4d54034dbcf2a4b19a11f39008a575aa614ea04703480b1022c

lhdrgui.exe

3.55 MB Size

2022-12-26 10:11:31 UTC

2 days ago

EXE

peexe malware macro-create-ole runtime-modules detect-debug-environment exploit cve-2017-0147 long-sleeps direct-cpu-clock-access checks-user-input cve-2017-0144

Community Score

DETECTION DETAILS RELATIONS BEHAVIOR COMMUNITY

Security Vendors' Analysis

Acronis (Static ML)	ⓘ Suspicious	Ad-Aware	ⓘ Trojan.Ransom.WannaCryptor.H
AhnLab-V3	ⓘ Trojan/Win32.WannaCryptor.R200572	Alibaba	ⓘ Ransom.Win32/WannaCry.398
ALYac	ⓘ Trojan.Ransom.WannaCryptor	Antiy-AVL	ⓘ Trojan/Win32.HackTool.a
Arcabit	ⓘ Trojan.Ransom.WannaCryptor.H	Avast	ⓘ Sf:WNCryLdr-A [Trj]
AVG	ⓘ Sf:WNCryLdr-A [Trj]	Avira (no cloud)	ⓘ TR/Ransom.IZ
Baidu	ⓘ Win32.Worm.Rbot.a	BitDefender	ⓘ Trojan.Ransom.WannaCryptor.H
BitDefenderTheta	ⓘ Gen:NN.ZexaF.36150.Jl0@eePsbmpl	Bkav Pro	ⓘ W32.WannaCry.LTI.Trojan
ClamAV	ⓘ Win.Ransomware.Wanna-9769986-0	Comodo	ⓘ TrojWare.Win32.WannaCry.jet@714um4
Cybereason	ⓘ Malicious.7c37d2	Cylance	ⓘ Unsafe
Cynet	ⓘ Malicious (score: 100)	Cyren	ⓘ W32/Trojan.ZTSA-8671
DrWeb	ⓘ Trojan.Encoder.11432	Elastic	ⓘ Malicious (high Confidence)
Emsisoft	ⓘ Trojan.Ransom.WanaCrypt0r (A)	eScan	ⓘ Trojan.Ransom.WannaCryptor.H
FSFT-NOD32	ⓘ Win32/Fxinil CVF-2017-0147 A	Fortinet	ⓘ W32/RANSOM Altr

## 4. Static Analysis

Here the Static analysis begins, we will now analyze the malware without running the malware. So, we will use some kind of tool for analysis which will be discuss below.

Now let analyze the malware with the help of strings, tries to find if we are able to get useful string.

```
__TREEPATH_REPLACE__
Microsoft Base Cryptographic Provider v1.0
Microsoft Security Center (2.0) Service
C:\%s\qeriuwjhrf
http://www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com
!This program cannot be run in DOS mode.
,4$8'9-6:.6$1#?*XhHpSeA~NrZlE
4$8,9-6'.6$:#?*1hHpXeA~SrZlN
$8,4-6'96$:.?*1#HpXhA~SeZlNrSbE
8,4$6'9-$:.6*1#?pXhH~SeAlNrZbE
inflate 1.1.3 Copyright 1995-1998 Mark Adler
```

We have got some useful strings which is shown above image. We came to know that there is some file creation on [C://?/?/qeriuwjhrf], There is using of sting format which means it will get path from string format and create the file.

There is one unknown domain which we can see, the malware should use domain to fetch something from internet.

Let's take the help of one tool which is "PEVIEW" which is present in flareVM. After importing the program in tool we get lots of information. We will get same information as we get in floss and getting hash. But we also see what API call is present within the binary.

The screenshot shows the PEVIEW tool interface. On the left, a tree view lists various binary sections like 'dos-header', 'rich-header', 'file-header', 'optional-header', 'directories', 'sections', 'libraries', 'imports', 'exports', 'exceptions', 'tls-callback', 'relocations', '.NET', 'resources', 'strings', 'debug', 'manifest', 'version', and 'overlay'. The 'version' section is selected, showing details for 'lhdfgrui.exe'.

property	value
md5	DB349B97C37D22F5EA1D1841E3C89EB4
sha1	E889544AFF85FFAF8B0D0DA705105DEE7C97FE26
sha256	24D004A104D4D54034DBCFFC2A4B19A11F39008A575AA614EA04703480B1022C
first-bytes-hex	4D 5A 90 00 03 00 00 04 00 00 FF FF 00 00 B8 00 00 00 00 00 40 00 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 BB 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 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 screenshot shows the 'Indicators (73)' section in PEVIEW. It lists various indicators categorized by type (file, library, resource, string, etc.) and their suspiciousness level (1 to 3). The 'strings' section is expanded, showing a list of strings and their sizes.

indicator (73)	detail	level
file > extensions (Ransomware   Wiper) > count	159	1
imports > flag	28	1
strings > flag	61	1
library > flag	IP Helper API	1
library > flag	Internet Extensions for Win32 Library	1
resource > size > suspicious	R:1831_3514368 bytes	1
library > flag	Windows Socket Library	1
URL > pattern	http://www.iugersodp9ifjaposdfjhgosurijfaewnwergwea.com	1
file > embedded	signature: executable_location: .data_offset: 0x0000B020_size: 5263716	1
file > embedded	signature: executable_location: .data_offset: 0x0000F080_size: 5297524	1
file > embedded	signature: executable_location: .rsrc_offset: 0x000320A4_size: 3514368	1
imports > anonymous	13	2
string > size > suspicious	1403 bytes	2
string > size > suspicious	1430 bytes	2
string > size > suspicious	1430 bytes	2
string > size > suspicious	1554 bytes	2
string > size > suspicious	2693 bytes	2
string > size > suspicious	2693 bytes	2
string > size > suspicious	2988 bytes	2
resources > file-ratio	94.41%	2
overlay > signature > name	executable	2
file > checksum > invalid	0x00000000	3
rich-header > offset	0x00000080	3
entry-point > location	0x00009A16	3
file > image-base	0x00400000	3
rich-header > checksum	0xC33D5D11	3
strings > count	114377	3
dos-stub > size	184 bytes	3



## 5. Dynamic Analysis

In this section we will cover about the dynamic analysis on binary and see what is actually happening behind the execution of the binary. We will run the malware in created Sandbox environment so that it will not pivot over the local network and analyze the malware working.

We will take help “Wireshark” and “Process-Monitor” tool which is pre-installed in FlareVM.

The image shows a Wireshark packet capture interface. The top pane displays a list of network packets. Packet 12 is highlighted, showing a GET request from 10.0.0.1 to 10.0.0.1 on port 80. The packet details pane on the left shows the structure of the packet, including Ethernet II, Internet Protocol Version 4, and Hypertext Transfer Protocol. The packet bytes pane on the right shows the raw data of the packet, which is a GET request to a fake InetSim HTTP server.

No.	Time	Source	Destination	Protocol	Length	Info
9	5.117924323	10.0.0.129	10.0.0.128	TCP	66	27268 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
10	5.117942446	10.0.0.128	10.0.0.129	TCP	66	80 → 27268 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128
11	5.118146441	10.0.0.129	10.0.0.128	TCP	66	27268 → 80 [ACK] Seq=1 Ack=1 Win=262144 Len=0
12	5.119172531	10.0.0.129	10.0.0.128	HTTP	124	GET / HTTP/1.1
13	5.119182703	10.0.0.128	10.0.0.129	TCP	54	80 → 27268 [ACK] Seq=1 Ack=101 Win=64256 Len=0
14	5.133238264	10.0.0.128	10.0.0.129	TCP	284	80 → 27268 [PSH, ACK] Seq=1 Ack=101 Win=64256 Len=150 [TCP segment of a reassembled PDU]
15	5.133459425	10.0.0.129	10.0.0.128	TCP	66	27268 → 80 [ACK] Seq=101 Ack=151 Win=261888 Len=0
16	5.133472386	10.0.0.128	10.0.0.129	HTTP	312	HTTP/1.1 200 OK (text/html)
17	5.133618273	10.0.0.129	10.0.0.128	TCP	66	27268 → 80 [ACK] Seq=101 Ack=409 Win=261632 Len=0
18	5.133692837	10.0.0.129	10.0.0.128	TCP	66	27268 → 80 [FIN, ACK] Seq=101 Ack=409 Win=261632 Len=0
19	5.133752404	10.0.0.129	10.0.0.128	TCP	66	27268 → 80 [RST, ACK] Seq=102 Ack=409 Win=0 Len=0

Wireshark · Follow TCP Stream (tcp.stream eq 1) · ens33

GET / HTTP/1.1  
Host: www.luderrfso0p91fjapodfjhg0sur1jfaawwergwea.com  
Cache-Control: no-cache

HTTP/1.1 200 OK  
Content-Length: 258  
Date: Sat, 31 Dec 2022 00:52:43 GMT  
Server: InetSim HTTP Server  
Connection: Close  
Content-Type: text/html

```
<html>
<head>
<title>InetSim default HTML page</title>
</head>
<body>
<p></p>
<p align="center">This is the default HTML page for InetSim HTTP server fake mode.</p>
<p align="center">This file is an HTML document.</p>
</body>
</html>
```

Host: www.luderrfso0p91fjapodfjhg0sur1jfaawwergwea.com\r\n

We can see that virus is reaching to the present domain but the virus is not harmed our system while it reaches to internet. Let's check one more time without internet.



When execute the program without the internet connection it starts working in background and encrypt the system. First, I saw that in “TCPVIEW” it gives random remote IP address which tries to communicate, it means that this is not a real IP.

Ransomware.wannacr...	9144	TCP	Syn Sent	169.254.104.33	27736	169.254.83.2	445	12/31/2022 2:44:00 PM	mssecsv2.0
Ransomware.wannacr...	9144	TCP	Syn Sent	108.0.0.129	27737	167.217.251.150	445	12/31/2022 2:44:00 PM	mssecsv2.0
Ransomware.wannacr...	9144	TCP	Syn Sent	108.0.0.129	27738	191.17.149.151	445	12/31/2022 2:44:00 PM	mssecsv2.0
Ransomware.wannacr...	9144	TCP	Syn Sent	108.0.0.129	27739	167.79.124.190	445	12/31/2022 2:44:00 PM	mssecsv2.0
Ransomware.wannacr...	9144	TCP	Syn Sent	169.254.104.33	27740	169.254.84.2	445	12/31/2022 2:44:00 PM	mssecsv2.0
Ransomware.wannacr...	9144	TCP	Syn Sent	169.254.104.33	27741	169.254.85.2	445	12/31/2022 2:44:00 PM	mssecsv2.0
Ransomware.wannacr...	9144	TCP	Syn Sent	108.0.0.129	27742	81.33.35.65	445	12/31/2022 2:44:00 PM	mssecsv2.0
Ransomware.wannacr...	9144	TCP	Syn Sent	108.0.0.129	27743	144.177.85.18	445	12/31/2022 2:44:01 PM	mssecsv2.0
Ransomware.wannacr...	9144	TCP	Syn Sent	169.254.104.33	27744	169.254.86.2	445	12/31/2022 2:44:01 PM	mssecsv2.0
Ransomware.wannacr...	9144	TCP	Syn Sent	108.0.0.129	27745	190.131.94.119	445	12/31/2022 2:44:01 PM	mssecsv2.0
Ransomware.wannacr...	9144	TCP	Syn Sent	108.0.0.129	27746	14.100.79.22	445	12/31/2022 2:44:01 PM	mssecsv2.0
Ransomware.wannacr...	9144	TCP	Syn Sent	169.254.104.33	27747	169.254.87.2	445	12/31/2022 2:44:01 PM	mssecsv2.0
Ransomware.wannacr...	9144	TCP	Syn Sent	169.254.104.33	27748	169.254.88.2	445	12/31/2022 2:44:01 PM	mssecsv2.0
Ransomware.wannacr...	9144	TCP	Syn Sent	108.0.0.129	27749	16.177.216.124	445	12/31/2022 2:44:01 PM	mssecsv2.0
Ransomware.wannacr...	9144	TCP	Syn Sent	108.0.0.129	27750	89.197.63.198	445	12/31/2022 2:44:01 PM	mssecsv2.0
Ransomware.wannacr...	9144	TCP	Syn Sent	108.0.0.129	27751	123.181.238.32	445	12/31/2022 2:44:01 PM	mssecsv2.0
Ransomware.wannacr...	9144	TCP	Syn Sent	169.254.104.33	27752	169.254.89.2	445	12/31/2022 2:44:01 PM	mssecsv2.0
Ransomware.wannacr...	9144	TCP	Syn Sent	108.0.0.129	27753	107.100.91.170	445	12/31/2022 2:44:01 PM	mssecsv2.0
Ransomware.wannacr...	9144	TCP	Syn Sent	169.254.104.33	27754	169.254.90.2	445	12/31/2022 2:44:01 PM	mssecsv2.0
Ransomware.wannacr...	9144	TCP	Syn Sent	108.0.0.129	27755	80.52.54.180	445	12/31/2022 2:44:01 PM	mssecsv2.0
Ransomware.wannacr...	9144	TCP	Syn Sent	108.0.0.129	27756	216.187.149.152	445	12/31/2022 2:44:01 PM	mssecsv2.0
Ransomware.wannacr...	9144	TCP	Syn Sent	108.0.0.129	27757	65.48.84.21	445	12/31/2022 2:44:01 PM	mssecsv2.0

In “Procmon” we are able to see that there is lot of files and registry operation are carried on in background on execution.

Time ...	Process Name	PID	Operation	Path
2:43:2...	Ransomware.w...	3588	Process Start	
2:43:2...	Ransomware.w...	3588	Thread Create	
2:43:2...	Ransomware.w...	3588	Load Image	C:\Users\hacker\Desktop\Ransomware.wannacr.exe
2:43:2...	Ransomware.w...	3588	Load Image	C:\Windows\System32\ntdll.dll
2:43:2...	Ransomware.w...	3588	Load Image	C:\Windows\SysWow64\ntdll.dll
2:43:2...	Ransomware.w...	3588	Create File	C:\Windows\Prefetch\WANNACRY.EXE-5664CACA.pf
2:43:2...	Ransomware.w...	3588	RegOpenKey	HKLM\System\CurrentControlSet\Control\Session Manager
2:43:2...	Ransomware.w...	3588	RegOpenKey	HKLM\System\CurrentControlSet\Control\Session Manager
2:43:2...	Ransomware.w...	3588	RegQueryValue	HKLM\System\CurrentControlSet\Control\Session Manager\RaiseExceptionOnPossibleDeadlock
2:43:2...	Ransomware.w...	3588	RegCloseKey	HKLM\System\CurrentControlSet\Control\Session Manager
2:43:2...	Ransomware.w...	3588	RegOpenKey	HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\Segment Heap
2:43:2...	Ransomware.w...	3588	RegOpenKey	HKLM\System\CurrentControlSet\Control\Session Manager\Segment Heap
2:43:2...	Ransomware.w...	3588	RegOpenKey	HKLM\SYSTEM\CurrentControlSet\Control\Session Manager
2:43:2...	Ransomware.w...	3588	RegOpenKey	HKLM\System\CurrentControlSet\Control\Session Manager
2:43:2...	Ransomware.w...	3588	RegQueryValue	HKLM\System\CurrentControlSet\Control\Session Manager\ResourcePolicies
2:43:2...	Ransomware.w...	3588	RegCloseKey	HKLM\System\CurrentControlSet\Control\Session Manager
2:43:2...	Ransomware.w...	3588	Create File	C:\Windows
2:43:2...	Ransomware.w...	3588	Load Image	C:\Windows\System32\wow64.dll
2:43:2...	Ransomware.w...	3588	Load Image	C:\Windows\System32\wow64win.dll
2:43:2...	Ransomware.w...	3588	QueryOpen	C:\Windows\System32\wow64log.dll
2:43:2...	Ransomware.w...	3588	Create File	C:\Windows
2:43:2...	Ransomware.w...	3588	QueryNameInfo...	C:\Windows
2:43:2...	Ransomware.w...	3588	Close File	C:\Windows
2:43:2...	Ransomware.w...	3588	RegOpenKey	HKLM\Software\Microsoft\Wow64\86
2:43:2...	Ransomware.w...	3588	RegQueryValue	HKLM\SOFTWARE\Microsoft\Wow64\86\Ransomware.wannacr.exe
2:43:2...	Ransomware.w...	3588	RegQueryValue	HKLM\SOFTWARE\Microsoft\Wow64\86\86\Default
2:43:2...	Ransomware.w...	3588	RegCloseKey	HKLM\SOFTWARE\Microsoft\Wow64\86
2:43:2...	Ransomware.w...	3588	Load Image	C:\Windows\System32\wow64cpu.dll
2:43:2...	Ransomware.w...	3588	RegOpenKey	HKLM\System\CurrentControlSet\Control\Session Manager
2:43:2...	Ransomware.w...	3588	RegOpenKey	HKLM\System\CurrentControlSet\Control\Session Manager
2:43:2...	Ransomware.w...	3588	RegSetInfoKey	HKLM\System\CurrentControlSet\Control\Session Manager
2:43:2...	Ransomware.w...	3588	RegQueryValue	HKLM\System\CurrentControlSet\Control\Session Manager\RaiseExceptionOnPossibleDeadlock
2:43:2...	Ransomware.w...	3588	RegCloseKey	HKLM\System\CurrentControlSet\Control\Session Manager

## 6. Overview of Basic Analysis

### 1. Internet connection Enable:

- a. When we execute the program while internet connection is enabled it tries to reach the domain.
- b. The malware didn't work, our all files were safe.

### 2. Internet connection Disable:

- a. When we execute the program, we able to see random IP which is trying to communicate with remote server.
- b. It creates file and registry operation.
- c. It encrypts all the content of the file and small pop-up is reflected in screen.

---

## 7. Advance Static Analysis

In this section we will analyze the code compares to basic static analysis but this time will analyze the assembly language of the binary. We need to know about machine level instruction to analyze the malware.

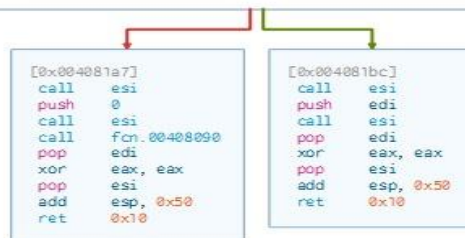
We will take help of cutter tool and know what it supposed to do.

In Image we can see that the API call is make when we execute the program. The domain address is stored in the ESI and afterward the ESI instruction is passed in the “InternetOpenUrlA” API call.

```

push esi
push edi
mov ecx, 0xe ; 14
esi, str:http://www.iugerfsodp9ifjaposdfjhgosurijfaewrwengwea.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
call dword [InternetOpenUrlA] ; 0x40a138
mov edi, eax
push esi
mov esi, dword [InternetCloseHandle] ; 0x40a13c
test edi, edi
jne 0x4081bc

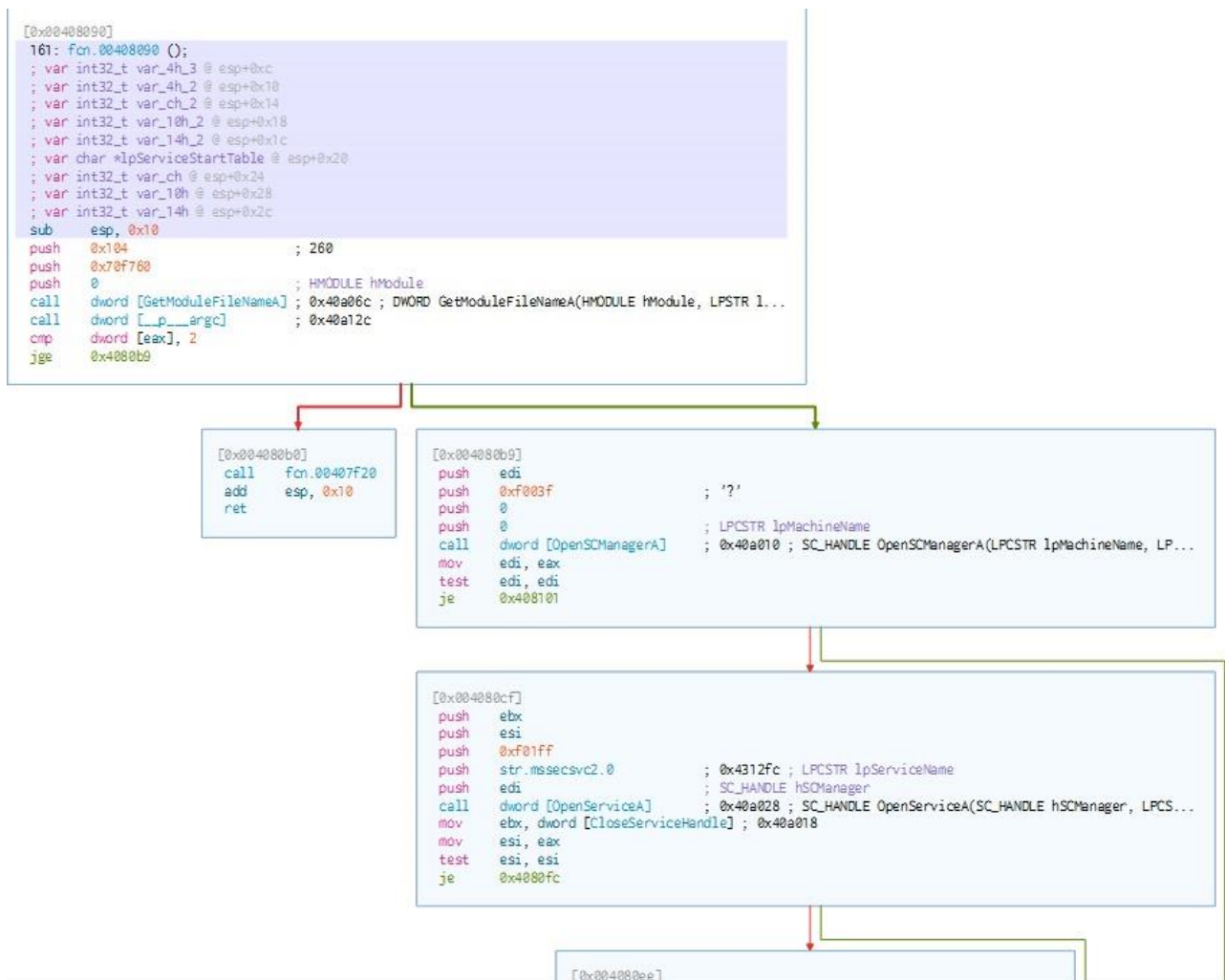
```



We can see that as the binary connects to internet it will immediately close the connection via “InternetCloseHandle” API call and does not harm the machine.

We can see “jne (jump if not equal to zero)” instruction which perform some condition. It checks the binary is able to communicate over the internet. If yes then it will return true (1) and exit binary or else it will return False (0) and continue.

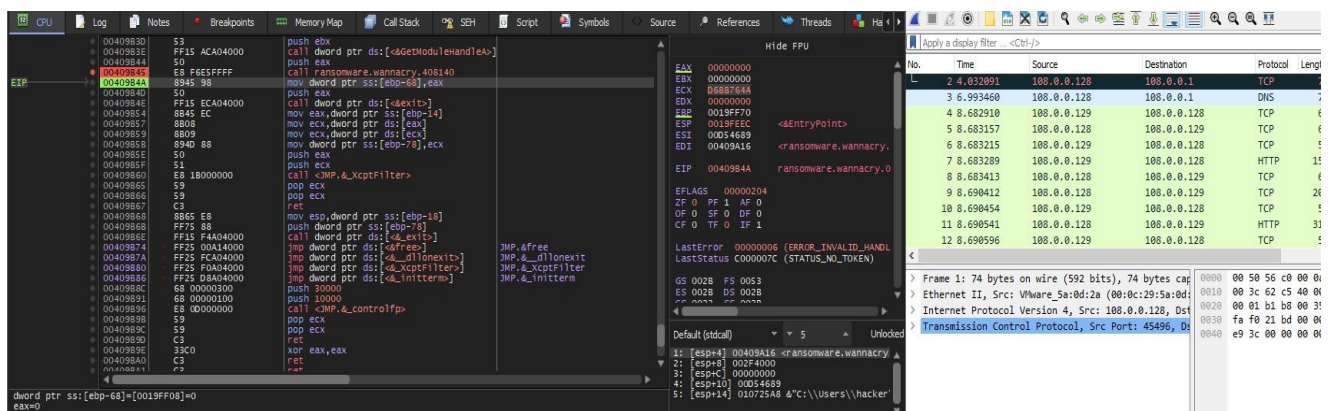
What if there is not internet connection? So, that time while condition checking it will return False as it not reaches to domain. There is function call within the condition, which take to address (0x00408090). Here program executes and encrypt all files present in the system.



## 8. Advance Dynamic Analysis

In this section the analysis will be same as Static but this time we will run the binary in debugger and control the binary. As we understand the static analysis, we are able to control the flow of binary.

For these we will take help of any debugger and control the flow of program



I have setup the breakpoint where the internet communication is to be done. When we move one step forward of that instruction, we are able to get HTTP response in Wireshark. Let take a look at the breakpoint call.



```

00408176 50          push eax
00408177 884424 68    mov byte ptr ss:[esp+68],al
00408178 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]
0040818E 8BF0        mov esi,eax
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
0040819C 56          push esi
0040819D 8B35 3CA14000 mov esi,dword ptr ds:[<&InternetCloseHandle>]
004081A3 85FF        test edi,edi
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 85C4 50     test eax,edx
004081B8 C2 1000     ret 1000
004081BC FFD6        call esi
004081BE 57          push edi
004081BF FFD6        call esi

```

Hide FPU

EAX 00CC000C  
 EBX 00000000  
 ECX C756088D  
 EDX 00000000  
 EBP 0019FF70  
 ESP 0019FE7C  
 ESI 72D503D0 <wininet.InternetCloseHandle>  
 EDI 00CC000C

EIP 004081A5 ransomware.wannacry.004081A5

EFLAGS 00000206  
 ZF 0 PF 1 AF 0  
 OF 0 SF 0 DF 0  
 CF 0 TF 0 IF 1

LastError 00000000 (ERROR\_SUCCESS)  
 LastStatus C000007C (STATUS\_NO\_TOKEN)

GS 002B FS 0053  
 FS 007B DS 007B

Default (stdcall)

1: [esp+4] 00409A16 <ransomware.wannacry.EntryPoint>  
 2: [esp+8] 004C4689  
 3: [esp+C] 70747468  
 4: [esp+10] 772F2F3A  
 5: [esp+14] 692E7777

Jump is taken  
 ransomware.wannacry.004081BC

When it hit the second breakpoint in above image it will tries to communicate over internet and goes further jump instruction. The jump instruction checks the condition we can see in right section of image the “ZF (Zero Flag)” its value will be 1 if the condition value is 0 or else it will be 0. As in image it is 0 so “JNE” will goes to arrow pointed toward another call.

What If we try to control the jump instruction? As we have full control of binary which is running in debugger. Let’s try it.



```

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]
0040818E  8BF0      mov esi,eax
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
0040819C  56        push esi
0040819D  8B35 3CA14000 mov esi,dword ptr ds:[<&InternetCloseHandle>]
004081A3  85FE      test edi,edi
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
004081BE  57        push edi
004081BF  FFD6      call esi
004081C1  5F        pop edi
004081C2  33C0      xor eax,eax
004081C4  5E        pop esi
  
```

Registers:

EAX	00000001
EBX	00000000
ECX	C756088D
EDX	00000000
EBP	0019FF70
ESP	0019FE7C
ESI	72D503D0
EDI	00CC000C
EIP	004081AB

Flags: 00000246  
 ZF 1 PF 1 AF 0  
 OF 0 SF 0 DF 0  
 CF 0 TF 0 IF 1

LastError: 00000000 (ERROR\_SUCCESS)  
 LastStatus: C000007C (STATUS\_NO\_TOKEN)

GS 002B FS 0053  
 FS 002B DS 002B

Default (stdcall)  
 1: [esp] 00000000  
 2: [esp+4] 00409A16 <ransomware.wannacry...>  
 3: [esp+8] 004C4689

In above image I have set the Zero Flag to 1 in which the condition will have value 0 and it will execute further. The binary will execute the function call and start causing harm to machine and we are able to see pop-up window of WannaCry Malware.

Hope so you get deep understanding for working of the malware.

## 9. Yara Rules

Yara rules classify and identify malware samples by creating descriptions of malware families based on textual or binary patterns. We can use Yara rules to define text or binary patterns that will match a file or component of a file to quickly find malicious files.

```
rule WannaCry_Malware {  
  meta:  
    latest_update = "01-01-2023"  
    author = "VZsecure"  
    description = "Analyze the wannacry malware, save your system from such attacks."  
  strings:  
    $Domain_call = "http://www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.comYOUARETHEMANNOWDOG" ascii  
    $Command_exec = "cmd.exe" ascii  
    $magic_byte = "MZ"  
    $File_Create = "tasksche.exe" ascii  
  condition:  
    $magic_byte at 0 and  
    $Domain_call or  
    $Command_exec or  
    $File_Create  
}
```

*Thank You!*