

A Comprehensive Analysis of WannaCry Ransomware

Dr Priya P Sajan¹, Kartikey Vaishnav², Manoj S Patil³, Priyanka S Kanade⁴, Sagar R Kale⁵, Bhagyashree Jadhav⁶

¹ Dr Priya P Sajan Cyber Security and Forensics, C-DAC, Thiruvananthapuram, India ²Kartikey Vaishnav Cyber Security and Forensics, C-DAC, Thiruvananthapuram, India ³Manoj Shridhar Patil Cyber Security and Forensics, C-DAC, Thiruvananthapuram, India ⁴Priyanka S Kanade Cyber Security and Forensics, C-DAC, Thiruvananthapuram, India ⁵Sagar Raghunath Kale Cyber Security and Forensics, C-DAC, Thiruvananthapuram, India ⁶Bhagyashree Jadhav Cyber Security and Forensics, C-DAC, Thiruvananthapuram, India

Abstract - WannaCry, a notorious ransomware strain that emerged in May 2017, quickly gained global attention due to its significant impact on both private and public sectors. This ransomware encrypted victims' files and demanded a Bitcoin ransom for their release, causing extensive disruption. Exploiting a vulnerability in Microsoft Windows' SMB protocol, known as (CVE-2017-0144), WannaCry spread "EternalBlue" rapidly across networks without user interaction. Notably, it severely affected the UK's National Health Service (NHS), disrupting healthcare services. The WannaCry attack highlighted critical cybersecurity vulnerabilities, emphasizing the need for timely software updates and robust security measures to defend against evolving cyber threat

Notably, the National Health Service (NHS) in the United Kingdom was among the high-profile victims, with the attack impacting numerous healthcare facilities and services. The WannaCry incident highlighted critical vulnerabilities in cybersecurity practices and the importance of timely software updates and robust security measures. It served as a stark reminder of the evolving nature of cyber threats and the need for ongoing vigilance in the protection of digital assets.

Key Words: WannaCry, ransomware, encryption, files, SMB, EternalBlue, healthcare impact, Bitcoin ransom, cyber threats.

1.INTRODUCTION

WannaCry is a form of malware classified as cryptoransomware, which encrypts users' files and demands a ransom for their decryption. The main objective of this malware is financial extortion. It employs fear tactics to pressure users into paying the ransom, exemplified by a countdown timer that threatens to permanently erase the decryption key if the payment is not made within three days.

Additionally, WannaCry is identified as a network worm because of its capability to self-replicate and spread across computer networks. Emerging on May 12, 2017, it is considered one of the most significant ransomware outbreaks in history, impacting over 200,000 computers in more than 150 countries.

The WannaCry attack comprises two primary elements: a worm module for propagating the infection and ransomware for encrypting files. It relies on Tor hidden services for command and control (C&C) functions, which are used to verify ransom payments and provide the decryption key. WannaCry is classified as selfpropagating malware due to its worm module, which is designed specifically to spread itself across both internal and external networks. This section will explore the vulnerability exploited by the malware and its propagation method.



2. METHODOLOGY

SMB Vulnerability

WannaCry takes advantage of a flaw in the Server Message Block (SMB) protocol within Windows systems. SMB is a transport protocol used for file sharing, printer sharing, and access to remote services in Windows, operating over TCP ports 139 and 445. The malware targets a vulnerability in SMB Version 1 (SMB v1) and TCP port 445 to spread. This weakness allows attackers to send specially crafted packets that execute arbitrary code on the victim's computer.

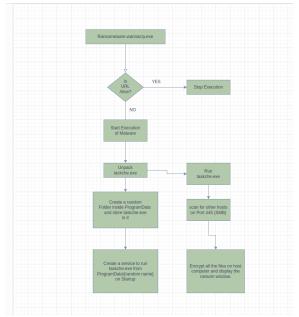


Fig.1.Malware Analysis Flowchart

Methods:

Now for the analysis of the wannacry ransomware we will mainly use Flare vm

installed on top a windows 10 virtual machine.

First we will do some analysis without detonation the malware on the vm we will just analyse the malicious executable.

3. IMPLEMENTAION

Now for the analysis of the WannaCry's ransomware we will mainly use Flare vm

installed on top a windows 10 virtual machine.

First we will do some analysis without detonation the malware on the vm we will just analyse the malicious executable.

3.1 STATIC ANALYSIS

Sample Overview

- Filename: Ransomware.wannacry.exe (may vary)
- **File Type:** Executable (.exe)
- File Size: Approximately 500 KB
- MD5 Hash: [db349b97c37d22f5ea1d18413c89eb4]
 SHA256 Hash:
 - [24d004a104d4d54034dbcffc2a4b19a11f39008a 575aa614e]

Extract readable text from the binary.

Tool used: Floss

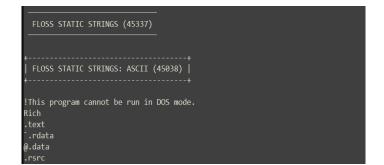


Fig.2 Floss Strings Floss

the strings This program cannot be run in DOS mode we can conclude that it's a portable binary. We also saw the repetition of the above string, so this might be a packed portable binary. Some suspicious exe we might want to note down for further analysis

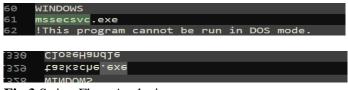


Fig.3 String Floss Analysis



3.2 Suspicious URL

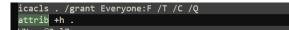
Now this an unregistered domain. However, a security researcher discovered this behavior and registered the domain, which effectively acted as a "kill switch" for the ransomware. Once the domain was registered, any instance of Wanna Cry that could reach the domain would stop executing, significantly reducing the spread and impact of the ransomware.

22	creati	eprocess	4						
34	http:	//www.iud	qerfsodp	<u>91</u>	fjapo	osd	Fjhgo	osurijfaewrwergwea.com	
35	!This	program	cannot	be	run	in	DOS	mode.	

Fig.4 Modules used to open suspicious URL

InternetCloseHandle	
InternetOpenUrlA	
InternetOpenA	
WININET.dll	

Now these modules are not itself malicious and are used in many legitimate software but also very common in malware trying to connect to a domain.



Icacls is a Windows utility (Displays or modifies discretionary access control lists (DACLs) on specified files) Here it is used to grant permission to everyone in current working directory and this directory is also hidden. Now these two strings raises a lot off suspicion as a normal user won't even know that this hidden directory even exist .

In further analysis we will discover that this directory in

C:\ProgramData\{hidden directory with random name} is used as a staging area for the malware execution

3.3 Tool used PEstudio

Here we can see that there are three packed binaries present inside the file.

1	indicator (38)	detail	level
	file > embedded	signature: executable, location: .data, offset: 0x0000B020, size: 5263716 b	*****
	file > embedded	signature: executable, location: .data, offset: 0x0000F080, size: 5297524 b	•••••
	file > embedded	signature: executable, location: .rsrc, offset: 0x000320A4, size: 3514368 b	*****

Fig.5 Some cryptography libraries

srand	×	0x0000A852	0x0000A852	692 (0x02B4)	crypto obfuscation
rand	×	0x0000A824	0x0000A824	678 (0x02A6)	crypto obfuscation
CryptAcquireContextA	×	0x0000A638	0x0000A638	133 (0×0085)	crypto obfuscation
<u>CryptGenRandom</u>	×	0×0000A650	0x0000A650	150 (0x0096)	crypto obfuscation

Fig.6 Windows Cryptography

The **CryptGenRandom** function is part of the Windows Cryptography API and is used to generate cryptographically secure random numbers. It fills a buffer with random bytes, which can be used for various cryptographic operations such as key generation, nonce creation, or other purposes where randomness is required.

The **CryptAcquireContext** function is part of the Windows Cryptography API, which is used to acquire a handle to a particular key container within a cryptographic service provider (CSP). This handle is then used in subsequent calls to other cryptographic functions.

3.4 Tool used disassembler cutter

Now we will use disassembler to convert machine code (binary code) back into assembly language, which is a more human-readable form of the code.

Fig7. Disassembler Cutter

Now the first thing to notice is that the string referenced to the URL is loaded in ESI (Extended Source Index). It's the same URL found in String analysis. In virus total.

[8x00408140]		
		r #Hangy, char #Henyp);
		Pistack - 0x64
		9 stack - 0x50 8 stack - 0x17
		e stack = kki17 # stack = (kk13
var int32		
var int32		
; var int32		
var int32.		
var int32		
0x00408140	sub	esp. 8x58
0000405143		
0x00405145	push	esi edi
8x88408145 8x8846814a		
2003405146	nov lea	<pre>esi, str.http:www.lugerfscdp9ifjaposdfjhgosurijfaewnwergwea.com ; 0x4313d0 edi, [var_50h]</pre>
8x88406153 8x88408155		
0x00405155		novsd dword es [edi], dword ptr [esi]
	novsb	byte es [edi], byte ptr [esi]
2x03405155		
8x8846815c		
8x03408160		
8x8848315c		
		dword [InternetOpenA] ; 0x40a134
		esi, eax
8x88488190		
		dword EinternetOpenUrIAT : 8x48a138
		edi, eax
		esi, dword [InternetCloseHandle] ; 0x40a13c
		edi edi

Fig.8 Security Venders Analysis

The first API call is **[InternetOpenA]** is part of the Windows API for handling HTTP/HTTPS requests. This function is used to initialize an application's use of the WinINet API, which provides functions for internet access.

Fig.9 InternetOpenA

0x00408175	push	eax
0x00408177	mov	byte [var_1h], al
0x0040817b	call	dword [InternetOpenA]; 0x40a134
0x00408181	push	
0x00408183	push	0x84000000
0x00408188	push	
0x0040818a	lea	ecx, [var_{
0x0040818e	mov	esi, eax
0x00408190	push	
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

In malware analysis, seeing **InternetOpenA** can indicate that the malware is attempting to establish internet connectivity, possibly for:

- Downloading additional payloads.
- Communicating with a command and control (C2) server.
- Sending Exfiltrated data.

0x00408176	push	eax
0x00408177	mov	byte [var_1h], al
0x0040817b	call	dword [InternetOpenA] ; 0x40a134
0x00408181	push	
0x00408183	push	0x84000000
0x00408188	push	
0x0040818a	lea	ecx, [var_64h]
0x0040818e	mov	esi, eax
0x00408190	push	
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	ine	0x4081bc

Fig.10 InternetOpenUrlA

InternetOpenA and its usage in both legitimate and malicious contexts is essential for effective network security and malware analysis.

The content of ESI is pushed onto the stack which will be used as a parameter for the API call. Then after that [InternetOpenUrlA] API is called with the URL as a parameter.

The InternetOpenUrlA function is part of the Windows API and is used to open a URL and obtain a handle to the internet resource. This function is typically used after initializing an internet session with InternetOpenA.

Now if the above API is able to connect to the URL then 0 is loaded in EAX (Extended Accumulator register) else 1 is loaded. The value of EAX is moved to EDI (Extended Destination Index).

0x0040819a	mov	edi,	eax
0x0040819c	nush	esi	

TEST edi , edi is ran which means bitwise boolean AND operator is used.

0700400130	nov	esi, awora Li	1
0x004081a3	test	edi, edi	

Next a jne (jump if not equal) instructions ran

0x004081a5	jne	0x4081bc	

0x004081a	5 jr	e	0x4081bc			
		Ţ			_	
0x004081a7] 0x004081a7		esi		[0x004081bc] 0x004081bc		esi
0x004081a9				0x004081be		edi
0x004081ab	call	esi		0x004081bf	call	esi
0x004081ad			00408090 ; fcn.00408090	0x004081c1	pop	edi
0x004081b2	рор	edi		0x004081c2		eax, eax
0x004081b3		eax,	eax	0x004081c4	рор	esi
0x004081b5	pop	esi		0x004081c5	add	esp, 0x50
0x004081b6	add	esp,	0x50	0x004081c8		
0x004081b9		0x10				

IF the API is able to make a connection with the (hxxp://iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.co m/) then



Fig.11 Kill Switch UPL Function

0x00408105 0x0040810d 0x0040810e 0x00408116 0x0040811e 0x00408126 0x0040812c 0x0040812d	lea mov push mov mov call pop add ret	eax, [lpServiceStartTable] dword [lpServiceStartTable], str.mssecsvc2.0 ; 0x4312fc eax ; unknom.t 41pServiceStartTable dword [Var_24h], 0x408000 ; data.00408000 dword [Var_26h], 0 dword [Var_1ch], 0 dword [Var_1ch], 0 dword [VartServiceCtrDispatcherA] ; 0x408000 ; BOOL StartServiceCtrlDis edi esp, 0x10
--	---	--

Then this part of the code runs which basically clear the stack and returns, which means malware does not run further and the victim is safe. Now this is the kill switch URL function if malware is able to connect to the url it acts as a kill switch. A security researcher known as "MalwareTech" discovered this behavior and registered the domain, inadvertently activating the kill switch. This action significantly slowed the spread of the ransomware and prevented further infections. The domain registration essentially created a sinkhole, preventing the malware from executing its payload on affected systems.

0x004081a	5 jr		0x4081bc				
		Ļ			_		
[0x004081a7] 0x004081a7 0x004081a9 0x004081ab	call push call	esi Ø esi		[0x004081bc] 0x004081bc 0x004081be 0x004081bf	call push call	esi edi esi	
0x004081ad 0x004081b2 0x004081b3 0x004081b5	call pop xor pop	edi	.00408090 ; fcn.00408090 , eax	0x004081c1 0x004081c2 0x004081c4 0x004081c5	pop xor pop add	edi eax, esi esp,	
0x004081b5 0x004081b6 0x004081b9	add ret		, 0x50 3	0x004081c8	ret	0x10	0220

Fig.12 Encryption Code Unpacked

If the other part of code runs it calls a **function** and clears the stack. This code is the rest of the encryption payload it open up and unpack the rest of the Unpacked Portable executable in a hidden directory. It then install itself as a service and becomes persistent. Every time a victim restart windows machine the service also starts up and encrypt any new files added to the machine.

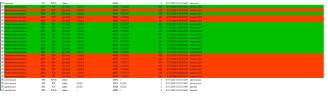


Fig.13 Static Service Dispatcher A

4. DYNAMIC ANALYSIS

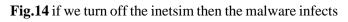
Now we will detonate the exe on the virtual machine

4.1 Tool used: inetsim

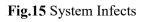
First we will detonate the malware with inetsim on When the malware is executed with inetsim set as our DNS resolver, the malware does not execute. It tries to connect to

"hxxp://www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergw ea.com". On successful connection it does not infect the system.

	1 0.00000000	Source 11.0.0.4	Destination 11.0.0.3	TCP	Length Info 66 49693 - 80 [SYN] Seg=0 Win=65535 Len=0 MSS=1460 WS=256 SACK PERN=1
	2 0.008815876		11.0.0.4	TCP	66 80 - 49693 [SYN, ACK] Seg=0 Ack=1 Min=64240 Len=0 MSS=1468 SACK PERM=1 MS=128
	3 0.000269208		11.0.0.3	TOP	60 49593 - 80 (ACK) Seg=1 Ack=1 Win=262144 Len=0
	4 0.000360939		11.0.0.3	HTTP	00 43033 - 00 [ALK] 300-1 ALK-1 WIN-202144 LEN-0
	5.0.000366057	11.0.0.4	11.0.0.3	TCP	154 GET 7 HTTP/1-1 54 80 - 49693 [ACK] Seg=1 Ack=101 Win=64256 Len=0
		11.0.0.3	11.0.0.4	TCP	54.80 - 49093 [ACK] Seq=1 ACK=101 Win=04250 Len=0 204.80 - 49093 [PSH, ACK] Seq=1 Ack=101 Win=64256 Len=150 [TCP segment of a reassembled PI
	6 0.013248401		11.0.0.4	TOP	
	7 0.013575934				60 49693 - 80 [ACK] Seq=101 Ack=151 Win=261888 Len=0
	8 0.013585518		11.0.0.4	HTTP	312 HTTP/1.1 200 OK (text/html)
	9 0.013764926		11.0.0.3	TCP	60 49593 - 80 [FIN, ACK] Seq=101 Ack=151 Win=261888 Len=0
	10 0.013764967		11.0.0.3	TCP	60 49593 - 80 [ACK] Seq=102 Ack=409 Win=261632 Len=0
	11 0.014173272		11.0.0.3	TCP	60 49593 - 89 [RST, ACK] Seq=102 Ack=409 Win=0 Len=0
	12 5.088301365		PcsCompu_6a:da:0		42 Mho has 11.0.0.47 Tell 11.0.0.3 60 11 8 8 4 1s at 08:00:27:6a:6a:80
	13 5.089877362	PcsCompu_6a:da:00	PcsCompu_52:5f:6	ic ARP	60 11.0.0.4 is at 08:00:27:6a:da:00
the nto ran	ernet II, Src: P ernet Protocol V nsmission Contro cricki Transfer ET / HTTP/1.1\r/	csCompu_6a:da:00 (08 ersion 4, Src: 11.0.) 1 Protocol, Src Port Protocol A	:00:27:6a:da:00}, E 0.4, Dst: 11.0.0.3 : 49693, Dst Port:	Dst: PosCompú. 80, Seq: 1, A	on interface englisä, id 0 52:87:66 (00:00:27:52:87:66) cck: 1, Len: 100
Ethi Inte Tran H C	ernet II, Src: P ernet Protocol V numission Contro ortext Transfer ET / HTTP/1.1\r\ ost: WWW.imperfs ache-Control: no r\n	csCompu_6a:da:00 (08 ersion 4, Src: 11.0: 1 Protocol, Src Port Protocol a odp91fjaposdfjhgosur	:00:27:6a:da:00), E 0.4, Dst: 11:0.0.3 : 49693, Dst Port: 1)facwrwergwea.com	Dst: PosCompú 80, Seq: 1, A \r\n	32:3f764 (00:00:27:32:3f76c) 641: 1, Len: 100
Ethi Inte Tran H C	ernet II, Src: P ernet Protocol V numission Contro ortext Transfer ET / HTTP/1.1\r\ ost: WWW.imperfs ache-Control: no r\n	csCompu_daida:00 (08 ersion 4, Src: 11.0.1 Protocol, Src Port Protocol m odp01fjaposdfjhgosur -cache\r\n : http://www.imperfs	:00:27:6a:da:00), E 0.4, Dst: 11:0.0.3 : 49693, Dst Port: 1)facwrwergwea.com	Dst: PosCompú 80, Seq: 1, A \r\n	32:3f764 (00:00:27:32:3f76c) 641: 1, Len: 100
	ernet II, Src: P ernet Protocol V nemission Contro ortext Transfer EI / HTTP/1.1/r/ ost: www.iugerfs ache-Control: no r\n Full request URI	csCopp_dsids:00 (00 ersion 4, Src: 11.0.1 Protocol, Src Port Protocol a odp01fjaposdfjhgosur -cache\r\n : http://www.iugerfs]	:00:27:6a:da:00), E 0.4, Dst: 11:0.0.3 : 49693, Dst Port: 1)facwrwergwea.com	Dst: PosCompú 80, Seq: 1, A \r\n	32:3f764 (00:00:27:32:3f76c) 641: 1, Len: 100
Ethe Inte Tran	ernet II, Src: P ernet Protocol V nomission Control Ortext Transfer ET / HTTP/3.3rv ost: www.imperfs ache-Control: no r\n Full request URI HTTP request URI	csCopp_dsids:00 (00 ersion 4, Src: 11.0.1 Protocol, Src Port Protocol a odp01fjaposdfjhgosur -cache\r\n : http://www.iugerfs]	:00:27:6a:da:00), E 0.4, Dst: 11:0.0.3 : 49693, Dst Port: 1)facwrwergwea.com	Dst: PosCompú 80, Seq: 1, A \r\n	32:3f764 (00:00:27:32:3f76c) 641: 1, Len: 100
	ernet II, Src: P ernet Protocol V memission Contro ertext Transfer EI/ HTTP/14.3xr ost: MAM.lugerfs ache-Control: no Full request UR HTTP request UR HTTP request UR HTTP request I/3	csCorpu_deida:00 (08 ersion 4, 8rc: 11.0. 1 Protocol, Src: Port rotocol, Src: Port adp01fjaposdfjhgosur- -cache\r\n : http://www.iugerfs] w: 8]	100:27:6a:da:00), E 0.4, Dst: 11.0.0.3 : 49093, Dst Port: 1jfaewrwergwea.com edp9ifjaposdfjhgosd	Dst: PosCompú_ 80, Seq: 1, A \r\n urijfacwrwerge	ssister (enenizisister.eo) Aki 1, ten: 500 es.cov]
Ethi Inte Trai H C L	ernet II, Src: P ernet Protecol V ernet Protecol V ertext Transfer ET / HTTP/12.3/Y/ ost: Max.Jugerfs Full request UR2 HTTP request 1/2 Response in fram 08 00 27 52 5f	estorpu_daida:00 (dB ersion 4, Src: 11.0.) I Protocol, Src: Port Protocol an odp01fjaposdfjhgosur- -cache\r\n : http://www.lugerfs] e: 0] 6c 08 00 27 6a da 0	<pre>:00:27:6a:da:00), E 0.4, Dst: 11.0.0.3 : 49653, Dst Port: ijfaewrwergwea.com edp0ifjaposdfjhgosi 0 08 00 45 00</pre>	Dst: PosCompú 80, Seq: 1, A Ar\n urijfamwrwergw 'R 1 'ii	2013fra((0800)27)255fra) Aki J, Len 190 ##.com/
Ethi Inte Tran C C C	ernet II, Src: P ernet Protecol V memission Contro ortext Transfer EIY HTTP/14.SNY ost: MAM.lugerfs ache-Control: no Full request UR HTTP requ	<pre>cscorpu_dsida:00 (08 ersion 4, src: 11.0. 1 Protocol, Src Port /retocol doplifjaposdfjhgosur achtekr/m : http://www.isoperfs] e: 0] 6c 08 00 27 6s da 0 08 00 6 28 98 00 27</pre>	<pre>:00:27:4a:da:00, t 0.4, Dst: 11.0.0.3 : 49053, Dst Port: i)facwrwergwea.com edp9ifjaposdfjhgosi 0 00:45:00 0 00:45:00</pre>	Dst: PosCompú 80, Seq: 1, A Ar\n urijfamwnwerge 'R_1- 'j1 -0 8	Szarfae (menizzisztef) Ak: 1, Len: 190 es.cor] E.
Ethi International H C \ []	ernet II, Src: P ernet Protecol V ernet Protecol V ertext Transfer II / HTTP/14, SVY ost: Maw.imperfs rim Full request UR2 Response in fram 08 00 27 52 5f 08 06 cab cd 40 08 03 cd 400	escorpu, da: da:00 (08 ersion 4, src: 11.8. 1 Protocol, Src Port <u>Srctocol</u> a dopi1ijaposdfjhgosur -cache\r\n :: http://www.luqerfs m: 0] 60 88 00 27 5a da 8 00 88 00 27 5a da 8 00 88 00 38 58 00 6 ce 2	100:27:4a:da:00), E 4., Dst: 13.0.0.3 : 49653, Dst Port: 1)faewrwergwea.com adp9ifjaposdfjhgosi 0 08 00 45 00 0 04 05 00 10 05 28 18	Dst: PosCompú_ 80, Seq: 1, A Arin urijfæwnwerge 'R_1- 'ji -@- 8 Pf-g	Salafie (Heleiziziafie) Aki J, Leu: 190
Ethi Inte Trai H C L [] [] [] [] [] [] [] [] [] [] [] [] []	ernet II, Src: P ernet Protocol V nemission Contro III MAN, Lugerfs ache-Control: no rin Ache-Control: no rin Full request URD HTTP request URD HTTP request J/J 88 00 27 52 5f 08 8c ab d 40 08 03 c2 1d 00 04 00 b8 12 00	csCorpu, ds: ds:00 (08 ersion 4, src: 11.8.1 1 Protocol, Src: Port Protocol a adpl:fjaposdfjhgosur - achekr/n : http://www.iugerfs] #: 0] 6: 08 00 27 6a da 0 08 00 0 27 6a da 0 08 00 0 0 dc e 0 5: 05 152 00 dc e 0 0 07 45 0 52 0 27 2	<pre>:00:27:4a:da:00, t 4, Dst: 11.0.0.3 : 49053, Dst Port: i)facwrwergwea.com edp9ifjaposdfjhgosi 0 00 45 00 :00 04 50 :00 04 :00 04 50 :00 04 :00 04 :00 :00 :00 :0</pre>	Dst: PcsCompú_ 80, Seq: 1, A \r\n urijfaewnwergu 'R_1- 'ji 8 8 - P	Salafia (Heleiziziafia) Aki 1, Leni 180 ma.tok) 6.
Ethi Intr Trai C \ [[] 00 100 100 100 100 100 100 100 100	ernet II, Src: P ernet Protocol V msmission Contro List Transfor EI / HITP/L.Str Bill request US Response in fran 08 80 27 52 5f 08 85 cab cd 40 08 83 cd 40 08 12 00 04 80 05 12 00	escorpu, da: da:00 (08 ersion 4, src: 11.8. 1 Protocol, Src Port <u>Srctocol</u> a dopi1ijaposdfjhgosur -cache\r\n :: http://www.luqerfs m: 0] 60 88 00 27 5a da 8 00 88 00 27 5a da 8 00 88 00 38 58 00 6 ce 2	100:27:4a:da:00), t 64, Dst: 13.0.0.3 149653, Dst Port: 1)faewrwergwea.com edp9ifjaposdfibgosi 8 08 00 45 00 8 08 00 45 00 10 db 25 00 10 db 73 08 10 db	Dst: PosCompú 80, Seq: 1, A Nr\n urljfatwrwerge 'R_1- 'ji 8 8 8 7 2 7 / 11 2 8 7 / 11	Statfie (Heinizzisisfie) Ak: 1, Len: 190 es.cor] E
Ethi Intr Trai C \ E E E E E	ernet II, Src: P ernet Protocol V memission Centro Mittale Areasion all / HTTP/4.3Vr OSI: Mar.iugerTS Phil Phil HTTP request URI HTTP request URI HTTP request URI HTTP request URI HTTP request URI HTTP and the second B 80 02 7 52 51 08 8c ab cd 40 08 80 c2 16 00 07 53 20 33 00 10 78 71 46 72	csCorpu, da: da:00 (08 ersion 4, src: 11.8.1 1 Protocol, Src: Port Protocol 8 adglfjaposdfjhgosur -cathekrin 1 http://www.isoperfs 3 e:8] 60 80 60 27 6a da 80 80 60 27 6a da 50 85 b2 06 dc e5 2 50 85 b2 06 85	100:27:04:04:00; K 04, D6:13:0.0.3 : 49693, D5: Port: 1jfaewrwergwea.com edp9ifjaposdfjhgos 0 06 45 00 0 06 45 00 0 06 45 00 0 07 7 77 27 0 06 65 01 10 06 70 17 10 06 50 10 10 00 100 10 10 00 100 100 10 00 10000000000	Dst: PosCompú 80, Seq: 1, A Vr\n urijfatwawerge 'R_1- 'j R_1- 'j R_1- 'j R_1- 'j R_1- 'j R_1- 'g R_1- 'g R_1- 'g R_1- 'g R_1- 'g	Saarfae (minizrisaff.ee) Aki 1, ken: 500 ee.coo/] E
Ethi Intr Trai C \ [[] 00 100 100 100 100 100 100 100 100	ernet II, Src: P ernet Protocol V nsmission Contro III / HTP/1.3VY. Soft and Protocol V Fill Control: no rh Full request U23 HTP request U23 Response in fram 08 80 27 52 57 08 86 ab cd 40 08 80 21 40 08 80 21 40 09 80 22 14 80 04 80 55 12 40 09 80 22 14 80 04 80 55 12 40 00 40 55 12 40 05 7 32 46 60 07 73 46 60	csCorpeu, ds: ds: 00 (8) ersion 4, 8rc 11.8. 1 Protocol, Src Port Protocol a ddp2fjaposdfjhgosur -cachety'n : http://www.imgerfs : ttp://www.imgerfs : bttp://www.imgerfs : bttp://wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww	100:27:04:04:00; K 0.4, Dot: 13.0.0.3 : 49693, Dst Port: 1)ffacurwergues.com edp9ifjaposdfjbgosi 00 04 05 00 00 04 05 00 3 06 71 58 10 00 04 55 00 3 06 71 58 10 0 06 66 01 76 0 06 05 05 0 06 05 05 0 06 05 05 0 06 05 05 0 06 05 0 06 05 0 0 0 0	Dat: PosCompú, 80, Seq: 1, A Vr\n urljfaterrerge 'R_1- 'ji Postafig R_1	Salarine (deležizianine) kt: 1 ₂ . ten: 190 es. cov] c
Ethi Intra Tran C \ [[[000 100 100 100 100 100 100 100 1	ernet II, Sicci P ernet Protocol V nsmission Contro ericat Transfor ET / NTTP/1.1VT No SC: MALINET Fache-Control: no rin Full request U2 Response in fram 08 00 27 52 57 08 06 ab cd 40 08 00 27 52 57 08 06 ab cd 20 08 00 27 52 57 08 06 ab cd 20 08 00 27 52 57 08 06 ab cd 20 08 00 27 52 57 07 50 50 50 07 50 50 07 50 50 50 07 50 50 07 50 50 07 50 50 05 50 05 05 05 05 05 05 05 05 05 05 05 05 0	csCorpu, da: da:00 (08 ersion 4, src: 11.8.1 1 Protocol, Src: Port Protocol 8 adglfjaposdfjhgosur -cathekrin 1 http://www.isoperfs 3 e:8] 60 80 60 27 6a da 80 80 60 27 6a da 50 85 b2 06 dc e5 2 50 85 b2 06 85	190127:481.04.00), L 64, DSC: 11.0.0.3 149653, DSL Port: 197664, DSC: 11.0.0.3 197664, DSL Port: 197664, DSL Port: 197664, DSL Port: 19767, DSL Port: 19777, 2077, 2073 19666, 20176, 101 19777, 2077, 2073 19666, 20176, 101 19772, 2077, 2073 19766, 201766, 20176, 20176, 20176, 20176, 20176, 20176,	Dst: PosCompú 80, Seq: 1, A Vr\n urijfatwawerga 'R_1- 'j R_1- 'j R_1- 'j R_1- 'j R_1- 'j R_1- 'g R_1- 'g R_1- 'g	Salafie (HENRIZISISTIN) Aki J, Len 190 Ha.com]



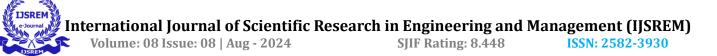
	eres)		



4.2 Tool Tcp view



Here we see that wanna cry is trying to target Port 445, which is primarily used for SMB its a network file sharing protocol that allows applications to read and write to files and request services from server programs. Wanna Cry exploited a vulnerability in the Server Message Block (SMB) protocol. This vulnerability, known as Eternal Blue.



Eternal Blue allowed the ransomware to spread quickly across networks by taking advantage of unpatched Windows systems.

4.3 Worm-Like Behavior

Once Wanna Cry infected a system, it used the SMB vulnerability to scan and infect other vulnerable systems on the same network. This worm-like behavior enabled it to spread rapidly from one infected machine to others within the same network or across connected networks.

CVE-2017-0144: This is the specific CVE identifier for the vulnerability.

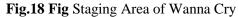
Conhost.exe (1996)	Console Window Host	C:\Windows\System32\Conhost.exe
taskdl.exe (2280)	SQL Client Configuration Utility EXE	C:\ProgramData\bzogshjqkzsqyv226\taskdl.exe
ggg cmd.exe (540)	Windows Command Processor	C:\Windows\SysWOW64\cmd.exe
Conhost.exe (2676)	Console Window Host	C:\Windows\System32\Conhost.exe
Sector (2408)	Microsoft @ Console Based Script Host	C:\Windows\SysWOW64\cscript.exe
istaskdl.exe (5252)	SQL Client Configuration Utility EXE	C:\ProgramData\bzogshjqkzsqyv226\taskdl.exe
taskdl.exe (5400)	SQL Client Configuration Utility EXE	C:\ProgramData\bzogshjqkzsqyv226\taskdl.exe
taskdl.exe (5564)	SQL Client Configuration Utility EXE	C:\ProgramData\bzogshjqkzsqyv226\taskdl.exe
taskdl.exe (5688)	SQL Client Configuration Utility EXE	C:\ProgramData\bzogshjqkzsqyv226\taskdl.exe
sychost.exe (1568)	Host Process for Windows Services	C:\Windows\system32\sychost.exe

4.4 Tool used procmon

Fig.17 Procmon tool used

we see a file named tasksche.exe is created which was also found in static analysis. By applying filters in procmon for parent id we will see all child processes spawned by our malware. Here we notice that a directory is created so let's check its content.

This is the staging area of wanna cry



	Local Dok (C) + ProgramDate	+ http://phote-125 +	-	-		v 0 P Seechimphyloge03	
	n	Date modified	100	-			
Quick access		\$13/201648 AM	The fielder				
Dektop /	Officers Read Math Art	Arthonics was deal	Text Decompany	1.63			
🕹 Devenisada 🕜 🤯	@Wwwalkerspice@.eve	\$/12/30/73/22 AM	Application	343 88			
S Decuments 🖌 😏	Companyation and	STRUCTURE AN	Shortout	182			
EFitam / 🗋	00000000.wky	\$13/3004-641 AM	EKY Film	0.62			
3 Mair	00000000. gay	0/13/2024/041 AM	PICTER	1.63			
	0000000C mm	6/13/2004/6/57 AM	RES File	1.03			
	b.wmy	5/11/20/7 @13 PM	WMEPTIK	1,407.63			
	CARRY	6/13/2024/641 AM	WMERTER	1.63			
	f.wmey	8/13/2024/648 AM	WM07758	1.63			
	sweey.	5/11/2017 3-59 PM	WM87File	1.68			
S Decuments	Linety	5/9/2017 4/58 PM	WMR7 File	2,668 838			
	Lwwy	5/12/00/73/22 AM	WMR7 File	65.83			
	tadidiore	5/12/2017 2:22 AM	Application	20 83			
	ted of team	6/13/2024/641 AM	Application	3,432.83			
	ted ou and	5/12/2017 2:22 AM	Application	20.83			
Local Dek (C)	awny	5/12/2017 2:22 AM	WMOTHE	240 83			
CD Drive DO Tirbar							
🦉 CD Drive (Di) Todae							
Network							

b.wnry is an image file used for displaying instructions for the decryption of user files. It starts with 42 4Dstrings, which indicates that this file is a bitmap image

c.wnry contains a list of Tor addresses with .onionextension and a link to a zipped installation file of the Tor browser from Tor Project

r.wnry is a text file in English with additional de-cryption instructions to be used by the decryptioncomponent

s.wnry file is a ZIP archive (HEX signature 50 4B 0304) which contains the Tor software executable. This executable has been obtained with the assistance of the Win Hex tool [12] by saving raw binary data with.zip extension.

t.wnry — Encrypted DLL containing file-encryption functionality.

u.wnry — Main module of the WCry ransomware "decryptor".

taskse.exe — Program that displays decryptor window to RDP sessions.

msg — Directory containing Rich Text Format (RTF) ransom demands in multiple languages.

taskdl.exe — WNCRYT temporary file cleanup program.

 N11
 Resonance assessment av
 N10 Chroadrik
 C/M benefisies to asses
 RCC020
 Deter Access To are discuss. Discuss To an annual international internatinterexecutive internatinterexecutive international inter

Fig.19 taskdl.exe is trying to end SQL Client Configuration utility executable

Here we can see that taskdl.exe is trying to end SQL Client Configuration utility executable so sql data can also be encrypted.

Similarly, WCry terminates several services so that their data stores can be encrypted:

taskkill.exe /f /im mysqld.exe

taskkill.exe /f /im sqlwriter.exe

taskkill.exe /f /im sqlserver.exe

taskkill.exe /f /im MSExchange

taskkill.exe /f /im Microsoft.Exchange.

A service with the same name is the directory is also created its a persistence mechanism so if the victim adds new files to the system it will also be encrypted.



	kzsqyv22	6 Propertie	s (Local Computer)	>
General	Log On	Recovery	Dependencies	
Service name: bzogshigkzsqyv226 Display name: bzogshigkzsqyv226 Description: Path to executable: cmd exe /c "C:\ProgramData\bzogshigkzsqyv226\tasksche.exe" Startup type: Automatic Service status: Stopped Start Stop Pause Resume You can specify the start parameters that apply when you start the service				
Beneral Log On Recovery Dependencies Service name: bzogshjqkzsqyv226 Display name: bzogshjqkzsqyv226 Description: ~ Path to executable: ~ cmd.exe /c "C:\ProgramData\bzogshjqkzsqyv226\tasksche.exe" ~ Startup type: Automatic ~ Service status: Stopped				
Descrip	tion:			< >
			bzogshjqkzsqyv226\tasksche.exe''	
Startup	type:	Automatic		\sim
			Pause Res	sume
S	itart n specify t	Stop		
You car from her	itart n specify t	Stop		

Fig.20 The startup type of this service is automatic

The startup type of this service is automatic. You can disable this service from windows services

5. Tool used x32 debugger:

Key Steps in Malware Analysis Using x32dbg

5.1. Setting Up a Safe Environment:

Virtual Machine: Always analyse malware in a virtual machine (VM) to prevent

accidental infection of your primary system. Tools like VMware or VirtualBox are commonly used.

Snapshots: Take snapshots of your VM before starting the analysis so you can easily revert to a clean state.

5.2. Initial Examination:

• Static Analysis: Before launching x32dbg, use static analysis tools like PEiD,

PEview, or CFF Explorer to gather basic information about the malware, such as packers, imports, and section headers.

• Identify Entry Points: Use these tools to identify the entry point of the executable, which will be useful when setting initial breakpoints.

5.3. Loading Malware into x32dbg:

Load the malware sample into x32dbg. Ensure that the process starts paused so

you can set breakpoints before any malicious code executes.

5.4. Setting Breakpoints:

f Task Manager					
File Options View					
Processes Performance App	history	Ratup Users Details Services			
Name	PID	Description	Status	Group	^
BcastDVRUserService		GameDVR and Broadcast User Service	Stopped	BcastDVRUser	
BoastDVRUserService 26501		GameDVR and Broadcast User Service 26501	Stopped	BcastDVRUser	
G. BOESVC		BitLocker Drive Encryption Service	Stopped	netsvcs	
C. BFE	1908	Base Filtering Engine	Running	LocalServiceN	
C. BITS		Background Intelligent Transfer Service	Stopped	netsycs	
C BluetoothUserService		Bluetooth User Support Service	Stopped	8thAppGroup	
BluetoothUserService_26501		Bluetooth User Support Service, 26501	Stopped	8thAppGroup	
C. Brokerinfrastructure	804	Background Tasks Infrastructure Service	Running	DcomLaunch	
C BTAGSenvice		Bluetooth Audio Gateway Service	Stopped	LocalServiceN	
BthAvctpSvc	1176	A/CTP service	Running	LocalService	
C. bthserv		Bluetooth Support Service	Stopped	LocalService	
a bzogshjąkzsąyv226		bzogshijąkzsąyv226	Stopped		
Camsvc		Capability Access Manager Service	Stopped	appmodel	
CaptureService		CaptureService	Stopped	LocalService	
CaptureService_26501		CaptureService_26501	Stopped	LocalService	
G cbdhsvc		Clipboard User Service	Stopped	ClipboardSvc	
Codhsvc_26501	3624	Clipboard User Service, 26501	Running	ClipboardSvc	
CDPSvc	1176	Connected Devices Platform Service	Running	LocalService	
CDPUsedvc		Connected Devices Platform User Service	Stopped	UnistackSvcGr	
CDPUserSvc_26501	1136	Connected Devices Platform User Service_26501	Running	UnistackSvcGr	
CertPropSvc		Certificate Propagation	Stopped	netsvcs	
CipSVC		Client License Service (ClipSVC)	Stopped	wsappx	
n electricher		Microsoft Cloud Islandiko Sansica	Stronmark	ClevelleServic	v

- Entry Point: Set a breakpoint at the entry point of the malware to start analysing its behaviour from the very beginning.
- Imports: Set breakpoints on important API calls that malware often uses, such as Create Process, WriteProcessMemory, RegSetValueEx, and networking functions like send or connect.

5.5. Dynamic Analysis:

- Step Through Code: Use single-step execution (F7) to walk through the code line by line. This helps in understanding the flow of execution and observing any suspicious behaviour.
- Function Calls: Pay close attention to function calls, especially those that interact with the operating system or manipulate files, memory, or the registry.
- Observing Registers: Monitor CPU registers closely as they often contain important information like addresses, return values, or parameters passed to functions.

5.6. Memory Inspection:

- Memory Dump: Inspect the memory dump to look for decrypted strings, injected code, or other indicators of malicious activity.
- Heap and Stack: Analyse the heap and stack for any anomalies or patterns that might indicate malicious behaviour, like unusual data being pushed onto the stack.

5.7. Code Patching and Manipulation:

Bypass Anti-Debugging: Many malware samples use anti-debugging techniques.

x32dbg allows you to patch out these checks so that you can continue your analysis without the malware detecting the debugger.

Modify Execution Flow: You can patch the code to alter the execution flow, such as skipping over harmful code or forcing specific outcomes to see how the malware reacts.

5.8. Behavior Analysis:

API Monitoring: Observe how the malware interacts with system APIs, focusing

on actions like file manipulation, registry changes, or network communications.

Network Traffic: If the malware connects to a remote server, analyse the traffic

it generates. You can use tools like Wireshark alongside x32dbg for this purpose.9. Identifying Persistence Mechanisms: Malware often tries to establish persistence by modifying startup entries or

dropping files in system directories. Track these activities by setting breakpoints

on relevant functions like RegCreateKeyEx or WriteFile.

5.9. Logging and Reporting:

Trace Logs: Use x32dbg's logging features to keep a record of important

function calls, register changes, and memory modifications.

Document Findings: Create detailed reports of your analysis, including

screenshots, code snippets, and explanations of how the malware operates.

6.Results:

6.1WannaCryRansomware: AGlobalCyberattack

WannaCry was a devastating ransomware attack that caused widespread disruption worldwide in May 2017. It exploited a vulnerability in Microsoft's SMB protocol known as EternalBlue, which had been leaked by the hacking group Shadow Brokers.

6.2 How WannaCry Worked

- 1. **Infection:** The ransomware spread rapidly through networks, infecting vulnerable computers and encrypting their files.
- 2. Encryption: Once infected, WannaCry would encrypt files using a strong encryption algorithm, making them inaccessible to the user.
- 3. **Ransom Note:** A ransom note was displayed on the infected computer, demanding a ransom payment in Bitcoin to decrypt the files.

6.3 Impact of the Attack

- **Global Disruption:** WannaCry affected hospitals, businesses, and government agencies around the world, causing significant disruptions to operations and services.
- **Financial Losses:** Many organizations suffered financial losses due to downtime, data loss, and ransom payments.
- **Public Safety Concerns:** In some cases, the attack compromised critical infrastructure, such as healthcare systems, posing risks to public safety.

6.4 The Eternal Blue Vulnerability and Global Response

The WannaCry ransomware attack exploited a vulnerability in Microsoft's Server Message Block (SMB) protocol known as EternalBlue. This vulnerability allowed attackers to execute arbitrary code on vulnerable systems, making them susceptible to infection.

6.5 The Eternal Blue Vulnerability:

- **SMB Protocol:** A network protocol used for sharing files and printers between computers.
- **Exploit:** The EternalBlue exploit allowed attackers to gain unauthorized access to vulnerable systems and execute malicious code.
- Origin: The exploit was stolen from the National Security Agency (NSA) and leaked by the hacking group Shadow Brokers.



7. Discussion:

WannaCry, a ransomware worm that exploited the EternalBlue vulnerability in Microsoft's SMB protocol, caused significant disruption worldwide in May 2017. This attack highlighted the criticality of cybersecurity and the potential consequences of vulnerabilities left unpatched.

One key discussion point is the role of nation-states in cyberattacks. The EternalBlue exploit was stolen from the NSA and leaked by the Shadow Brokers group. This raises questions about the responsibility of governments in securing their cyber capabilities and the potential consequences of such leaks.

Another important aspect is the rapid spread of WannaCry due to its worm-like nature. This underscores the need for robust network segmentation and proactive monitoring to prevent lateral movement of malware within organizations. Organizations must also invest in endpoint security solutions to detect and block malicious activity.

The attack also exposed the vulnerabilities of legacy systems, as many of the affected computers were running older versions of Windows. This highlights the importance of maintaining up-to-date software and implementing a regular patching schedule.

Furthermore, WannaCry serves as a reminder of the financial risks associated with ransomware attacks. Businesses and individuals who are not prepared with adequate backups or insurance may face significant losses.

In conclusion, the WannaCry ransomware attack was a watershed moment in the cybersecurity landscape. It exposed critical vulnerabilities, highlighted the role of nation-states in cyber warfare, and underscored the need for organizations to prioritize cybersecurity best practices. The lessons learned from this attack can help organizations strengthen their defenses against future ransomware threats.

8. CONCLUSION

Wanna Cry is an opportunistic ransomware family whose propagation methods allow it to spread quickly. CTU researchers recommend that clients implement the following best practices to mitigate the threat: Apply the Microsoft security updates for MS17-010, including the updates for the Windows XP and Windows Server 2003 legacy operating systems.

- Disable SMBv1 on systems where it is not necessary (e.g., hosts that do not need to communicate with Windows XP and Windows 2000 systems). Carefully evaluate the need for allowing SMBv1-capable systems on interconnected networks compared to the associated risks.
- Segment networks to isolate hosts that cannot be patched, and block SMBv1 from traversing those networks.
- Scan networks for the presence of the Double Pulsar backdoor using plugins for tools such as Nmap.
- Use network auditing tools to scan networks for hosts that are vulnerable to the vulnerabilities described in MS17-010.
- Filter emails containing potentially dangerous file types such as executables , scripts, or macro-enabled documents.
- Implement a backup strategy that includes storing data using offline backup media. Backups to locally connected, network-attached, or cloud-based storage are often insufficient because ransomware frequently accesses and encrypts files stored on these systems.

ACKNOWLEDGEMENT

We would like to extend my sincere and heartfelt thanks towards Project Guide Dr. Priya S and Mr. Jayaram P helped us in making this project. It gives us immense pleasure and satisfaction to put forward this report and to express our sincere gratitude to many staff who have helped us directly or indirectly in this project. First of all, we would like to thank our Project guide, Dr. Priya Sajan, for her support, encouragement, and guidance at every stage of our project.

We would like to thank our course co-ordinator, Mr. Jayaram, for his support and encouragement throughout the course. We would also like to thank all teaching staff along with our parents and friends who has helped us in every little way towards the working of the project.

REFERENCES

Books:

1.Skoudis, E., & Liston, T. (2006). *Counter Hack Reloaded: A Step-by-Step Guide to Computer Attacks and Effective Defenses* (2nd ed.). Prentice Hall.

2.Sikorski, M., & Honig, A. (2012). *Practical Malware Analysis: The Hands-On Guide toDissecting Malicious Software*. No Starch Press.

3.Stuttard, D., & Pinto, M. (2011). *The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws.* Wiley.

Journal Articles:

1.S Megira¹, A R Pangesti¹ and F W Wibowo¹ Published under licence by IOP PublishingLtd Journal of Physics: Conference Series, Volume 1140, International Conference on Electrical, Electronic, Info+rmatics and Vocational Education (ICE-ELINVO 2018)13 September 2018, Yogyakarta Special Province, Republic of IndonesiaCitation S Megira et al 2018 J. Phys.: Conf. Ser. 1140 012042DOI 10.1088/1742-6596/1140/1/012042

Conference Paper:

1.Malware Analysis October 2014

DOI:<u>10.13140/2.1.4750.6889</u>, Conference: Ethical Hacking, At: Nirma University <u>https://www.researchgate.net/publication/267777</u> 154_Malware_Analysis

2.A Survey on Malware Analysis Techniques: Static, Dynamic, Hybrid and Memory Analysis, September 2018, International Journal on Advanced Science Engineering andInformation Technology 8(4-2):1662 8(4-2):1662

DOI:<u>10.18517/ijaseit.8.4-2.6827</u> License <u>CC BY 4.0</u> <u>https://www.researchgate.net/publication/328760930 A</u> <u>Survey_on_Malware_Analysis_</u> <u>Techniques_Static_Dynamic_Hybrid_and_Memory_An</u> alysis

Online Resources

1.TCM Security Academy. (n.d.). "Practical Malware Analysis & Triage." Retrieved from <u>https://academy.tcm-</u> <u>sec.com/p/practical-malware-analysis-</u> triage

2.Husky Hacks. (n.d.). "Notes on Malware Analysis and Cybersecurity." Retrieved from https://notes.huskyhacks.dev/

3. <u>https://github.com/HuskyHacks/PMAT-labs</u>