A Comprehensive Analysis of WannaCry Ransomware

Dr Priya P Sajan[1], Kartikey Vaishnav[2], Manoj S Patil[3], Priyanka S Kanade[4], Sagar R Kale[5], Bhagyashree Jadhav[6]

[1] *Dr Priya P Sajan Cyber Security and Forensics, C-DAC, Thiruvananthapuram, India*
[2]*Kartikey Vaishnav Cyber Security and Forensics, C-DAC, Thiruvananthapuram, India*
[3]*Manoj Shridhar Patil Cyber Security and Forensics, C-DAC, Thiruvananthapuram, India*
[4]*Priyanka S Kanade Cyber Security and Forensics, C-DAC, Thiruvananthapuram, India*
[5]*Sagar Raghunath Kale Cyber Security and Forensics, C-DAC, Thiruvananthapuram, India*
[6]*Bhagyashree 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 "EternalBlue" (CVE-2017-0144), WannaCry spread 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 crypto-ransomware, 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 self-propagating 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.

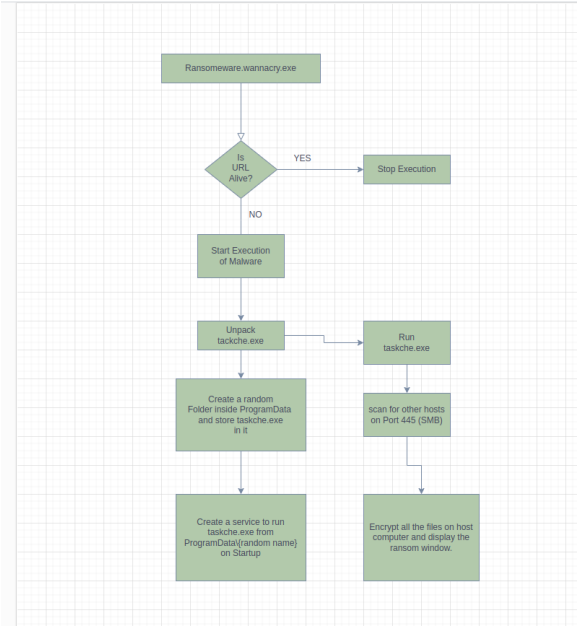


Fig.1.Malware Analysis Flowchart

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:** [24d004a104d4d54034dbcffc2a4b19a11f39008a575aa614e]

Extract readable text from the binary.

Tool used: Floss

```
FLOSS STATIC STRINGS (45337)

+------------------------------------+
| FLOSS STATIC STRINGS: ASCII (45038) |
+------------------------------------+

!This program cannot be run in DOS mode.
Rich
.text
`.rdata
@.data
.rsrc
```

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

```
60  WINDOWS
61  mssecsvc.exe
62  !This program cannot be run in DOS mode.
```

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.

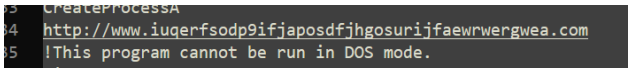
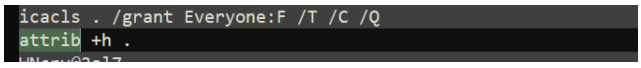


Fig.4 Modules used to open suspicious URL

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.

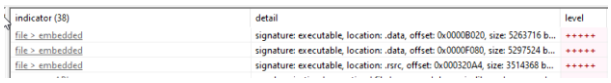


Fig.5 Some cryptography libraries

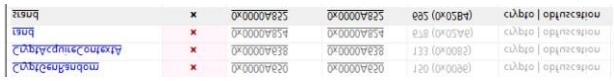


Fig.6 Windows Cryptography

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 .

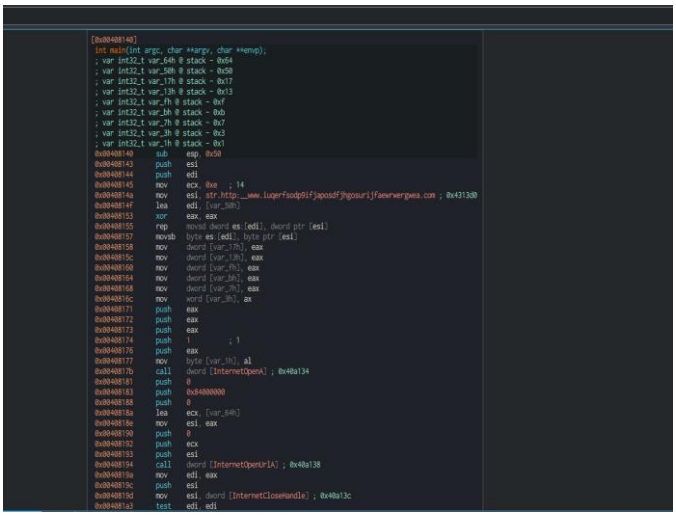
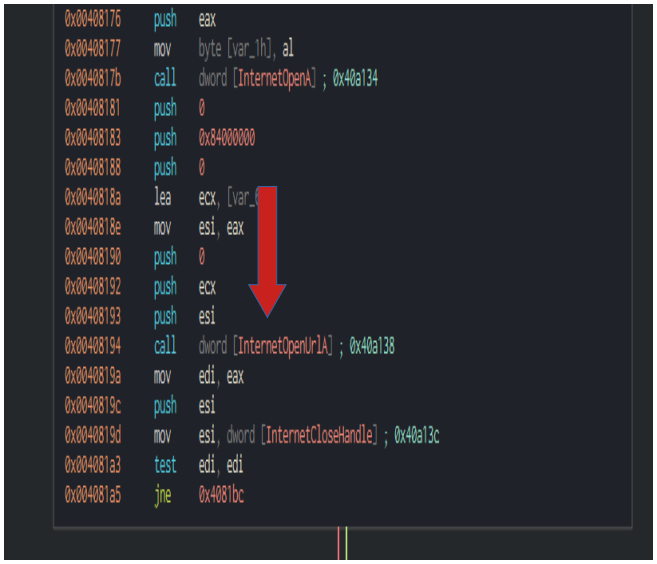


Fig.8 Security Venders Analysis

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



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.

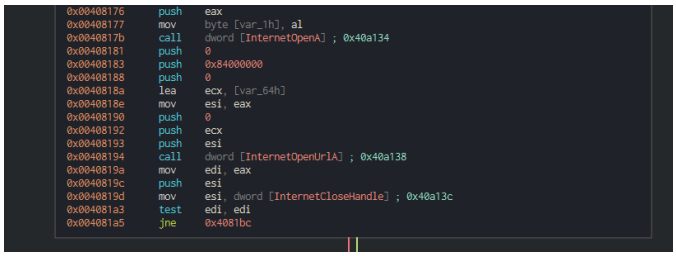


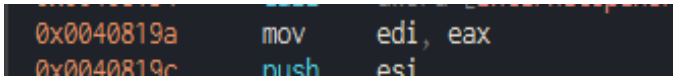
Fig.10 InternetOpenUrlA

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

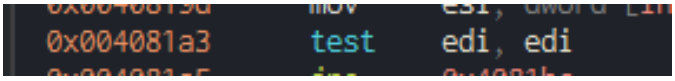
The content of ESI is pushed onto the stack which will be used as a parameter for the API call. Then after that [InternetOpenUrlA] API is called with the URL as a parameter.

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).



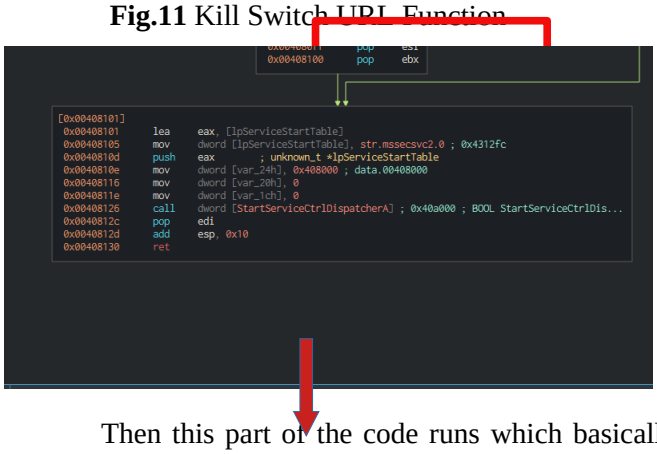
TEST edi , edi is ran which means bitwise boolean AND operator is used.



Next a jne (jump if not equal) instructions ran

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

Fig.11 Kill Switch URL Function



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.

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.

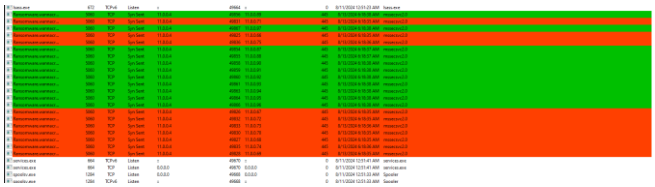


Fig.13 Static Service Dispatcher A

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.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com". On successful connection it does not infect the system.

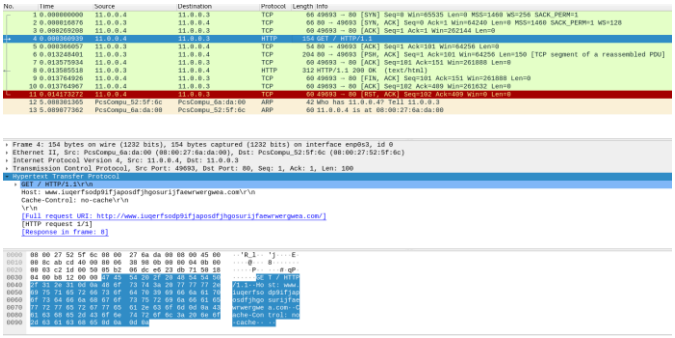


Fig.14 if we turn off the inetsim then the malware infects

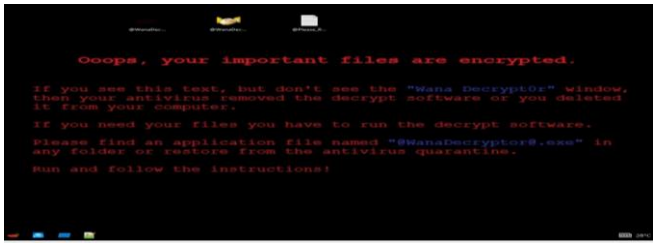


Fig.15 System Infects

4.2 Tool Tcp view

Fig.16 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.

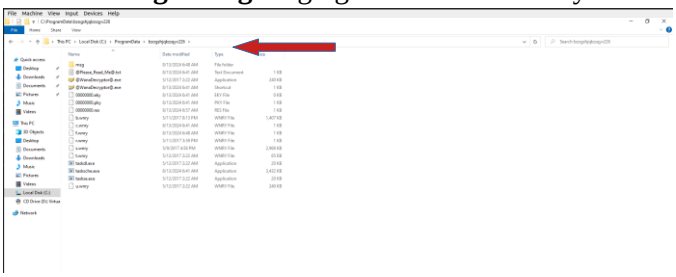
4.4 Tool used procmon

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

Fig.18 Fig Staging Area of Wanna Cry



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.

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.

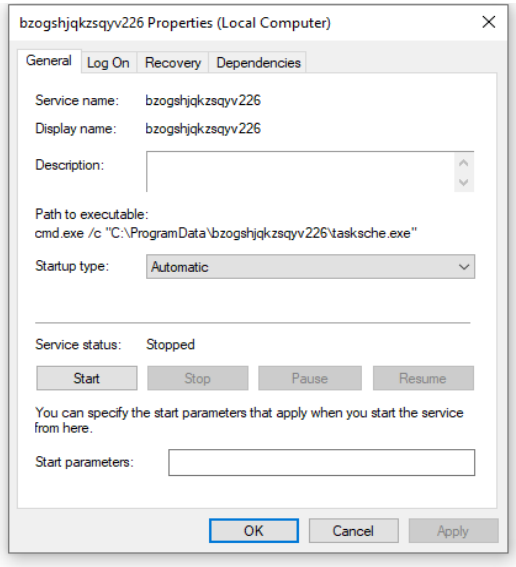


Fig.20 The startup type of this service is automatic

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.

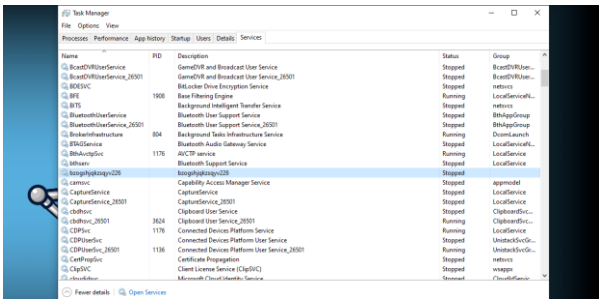
- Identify Entry Points: Use these tools to identify the entry point of the executable, which will be useful when setting initial breakpoints.

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:



- 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 Megira[1], A R Pangesti[1] and F W Wibowo[1] 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 Indonesia**Citation** S Megira *et al* 2018 *J. Phys.: Conf. Ser.* **1140** 012042**DOI** 10.1088/1742- 6596/1140/1/012042

Conference Paper:

1.Malware Analysis October 2014

DOI:10.13140/2.1.4750.6889 , Conference: Ethical Hacking , At: Nirma University https://www.researchgate.net/publication/267777154_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:10.18517/ijaseit.8.4-2.6827 License CC BY 4.0 https://www.researchgate.net/publication/328760930_A_Survey_on_Malware_Analysis_Techniques_Static_Dynamic_Hybrid_and_Memory_Analysis

Online Resources

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

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

3. https://github.com/HuskyHacks/PMAT-labs