Analyzing WannaCry Ransomware Considering the Weapons and Exploits

Da-Yu KAO*, Shou-Ching HSIAO**, Raylin TSO***

*Department of Information Management, Central Police University, Taoyuan City 333, Taiwan

**Haishan Precinct, New Taipei City Police Department, New Taipei City 220, Taiwan

***Department of Computer Science, National Chengchi University, Taipei 116, Taiwan

dayukao@gmail.com, oliver84312@gmail.com, raylin@cs.nccu.edu.tw

Abstract— **As ransomware has increased in popularity, its creators are using our fears to their advantage. The rapid proliferation of ransomware attacks indicates the growing tendency of ransomware-as-a-service (RaaS) and the integration of hacking weapons. This paper presents the analysis of the infamous WannaCry ransomware, which is one of the most propagated and damaging malware in 2017. The anatomy of ransomware attacks is discussed to understand the multi-phased execution of WannaCry, including the deployment, installation, destruction, and command-and-control. The chain of WannaCry's execution comprises several hacking weapon components. WannaCry not only embeds the binary in the resource section for multi-phased execution, but also implements a strong encrypting algorithm and a key structure. A reverse engineering analysis of each component, along with the network analysis of WannaCry's exploits offers an insight into the inner design of WannaCry. The observations of this research contribute to recent security systems and future defense strategies.**

***Keywords*—Ransomware, Reverse Engineering Analysis, Network Analysis, Hacking Weapons, WannaCry Exploits**

I. INTRODUCTION

During previous decades, malware has evolved in terms of the sophisticated obfuscation of malicious software and the diversity of attack vectors [4]. Ransomware is one of the greatest and most rapidly growing threats to the digital world [11]. Ransomware typically operates by locking the desktop of a computer and by rendering it to be inaccessible to users or by encrypting, overwriting, or deleting the user's files [6]. Ransomware can cause global catastrophes using encryption to hold the victims' data for ransom. Further, ransomware attacks continue to target out-of-date systems as the recent WannaCry ransomware (also known as WCry, WannaCrypt, WannaCrpyt0r, or WannaCrpytor) has spread in a tragic scenario containing thousands of computers [5, 10]. The emergence of malware creation tools has facilitated the creation of new variations of the existing ransomware [1]. Ransomware can easily to modify its ability to propagate quickly [10]. The dark web is a repository of the hacking weapons. By installing the TOR (The Onion Router) browser, criminals can access the dark web to realize their intentions of conducting ransomware attacks, which requires only a few hundred dollars. Easy access to hacking weapons lowers the barrier to initialize a cyberattack. After a hacking weapon was newly developed and implemented during a hacking campaign or malware outbreak, it has become a component in the circular chain of hacking weapons. An observation of the hacking weapons in WannaCry has revealed that some of the modular code was obtained from public source or covert hacker channels while the other parts of code were observed to be designed by the creator. Additionally, the hacking weapons are reusable by nature.

A literature review is presented in Section 2. A reverse engineering analysis of WannaCry components is discussed in Section 3. A network analysis of WannaCry exploits is presented in Section 4, and our research observations about the execution of multi-stage WannaCry are described in Section 5. Our conclusions are given in Section 6.

II. LITERATURE REVIEWS

WannaCry contains various modular hacking weapons in its composition. (Fig. 1).

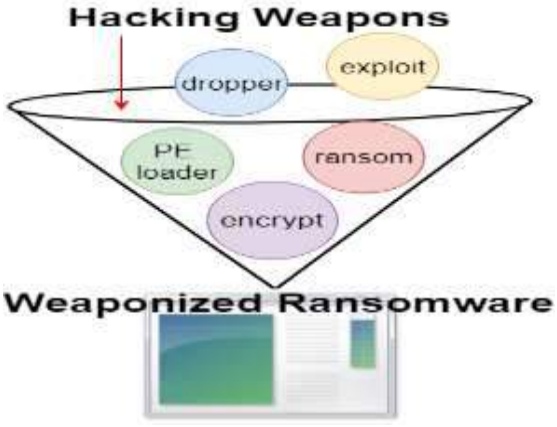

Fig. 1. Hacking weapons in weaponized ransomware

Manuscript received Dec 2, 2017. This work was a follow-up of the invited journal to the accepted & presented paper of the 20th Conference on Advanced Communication Technology (ICACT2018), and this research was partially sponsored by the Executive Yuan of the Republic of China under the Grants Forward-looking Infrastructure Development Program (Digital Infrastructure-Information Security Project-107) and the Ministry of Science and Technology of the Republic of China under the Grants MOST 106-2221-E-015-002.

Da-Yu Kao is with the Department of Information Management, Central Police University, Taoyuan City 333, Taiwan (Corresponding Author phone: +886-3-328-2321; fax: +886-3-328-5189; e-mail: dayukao@gmail.com).

Shou-Ching Hsiao is with Haishan Precinct, New Taipei City Police Department, New Taipei City 220, Taiwan (phone:+886-2-964-0329; e-mail: oliver84312@gmail.com).

Raylin Tso is with the Department of Computer Science, National Chengchi University, Taipei 116, Taiwan (phone: +886-2-2939-3091; +886-2-2937-8629; e-mail: raylin@cs.nccu.edu.tw).

A. *Weaponized Malware*

Weaponized malware deserves its name in two folds: the sophistication in composition and the intent for malicious purposes. As the more complexity and scale of malware attacks increase, malware developers tend to weaponize the malicious binary using different hacking weapons. WannaCry is a compound example of malware that not only contains dropper, resource loader, and ransomware binary for multi-execution flow, but is also weaponized with the Eternalblue exploit to ensure worm propagation capability. WannaCry is a type of worm-enabled ransomware and can be used as a weapon of digital destruction, which has cast a gloom over hospitals, banks, and enterprises all over the world, forcing individuals, enterprises, and public agencies alike to cease operation as people they attempt to cope with their infected computers.

B. *RaaS: Ransomware-as-a-service*

As cyber attackers increasingly use various Internet referrals to acquire ransomware modules, the convenience and service in RaaS have made it a new trend for people who intend to commit cybercrimes [1]. Since cybercriminals simply release these malicious codes on open source platforms, cybercrime has nothing to do with programming ability or hacking techniques anymore. Further, anyone can implement a ransomware attack easily, thereby drastically increasing the number of ransomware attacks. The unprecedented scale of RaaS has made cybercrimes more achievable and attainable, causing the wide spread of ransomware.

C. *Anatomy of a Ransomware Attack*

The objective of any ransomware attack is to extort the victims. Hackers who intend to conduct any type of attack tend to follow typical attack techniques and procedures. The life cycle of a general ransomware attack comprises the following stages [1] [7]:

1) ***Deployment.*** The first stage of a ransomware attack is to enter into targeted machine and execute its files. Several deployment methods, including phishing emails, malicious websites, vulnerable exploits are observed to vary from one to another.

2) ***Installation.*** After accessing the system initially, the ransomware will install and attempt to take complete control of the infected host. After successful control, the ransomware may either add its autorun registry key, create itself as a service, or dll load-order hijacking to achieve persistence.

3) ***Destruction.*** The ransomware blocks users' access to documents or systems by locking and encrypting files on the compromised device. Usually, ransomware will use public key algorithm along with private key algorithm to form complex encryption structure.

4) ***Command-and-Control.*** The actions of ransomware attack depend on the form of command-and-control systems. The metamorphic ransomware families and variants may differentiate the command-and-control channels, which may sometimes be as simple as web-based communications using HTTP protocol to as complex as the complicated TOR service connections.

III. Reverse Engineering Analysis Of WannaCry Components

The main reason for applying reverse engineering to the WannaCry ransomware is to reveal the actual functionality of the binary, which is a module of code, and why it comes as such designation. The "IDA Pro" is a useful reversing tool for disassembling the WannaCry binaries and offers a deep insight about the manner in which the WannaCry was developed and about the details of its execution flow. Different components, such as the launcher, dropper, resource loader, main ransom body, and encryption dll, implement the functionality in each phase. The chain of reverse engineering analysis is explored by extracting the main components during execution. The details of WannaCry components are listed in Table I.

A. *Deployment Phase: Export PlayGame*

A WannaCry ransomware attack exploits the MS17-010 vulnerability to inject the initial binary "launcher.dll" through the Eternalblue exploit and Doublepulsar backdoor. WannaCry exploits the SMB driver "srv2.sys" in the kernel module to access the compromised devices and inject the malicious payload [5]. Further, "launcher.dll" is injected into the lsass.exe system process and serves as the loader for mssecsvc.exe (Fig. 2).

Fig. 2. Launching mssecsvc.exe within lsass.exe process

The "launcher.dll" is executed only in memory and leaves no file artifacts on disk. This paper examined the lsass process memory from memory dumps using the "RWX (Read, Write, and Execute)" permission attributes. After the dumped memory was loaded into IDA Pro, the exported entry exhibited that this DLL can be accommodated within PlayGame, which is tasked to start up the ransomware execution. The PlayGame function mainly calls two sub-functions, "ExtractResource" and "CreateProcessMSSECSVC" (Fig. 3).

```
; Exported entry   1. PlayGame


public PlayGame
PlayGame proc near
push    offset aMssecsvc_exe ; "mssecsvc.exe"
push    offset aWindows ; "WINDOWS"
push    offset Format   ; "C:\\%s\\%s"
push    offset Dest     ; Dest
call    ds:sprintf
add     esp, 10h
call    ExtractResource
call    CreateProcessMSSECSVC
xor     eax, eax
retn
PlayGame endp
```

Fig. 3. Export PlayGame

TABLE I

Main Components of WannaCry

Phase	Execution Component	WannaCry		
		File (Internal) Name	***File Description***	***SHA256***
Deployment	Export PlayGame	launcher.dll	Inject through Doublepulsar backdoor	9411c59a83c8c32a925d53a902bef168ebe5b403a88ab4d8dfe807fd7435dd9e
Installation	Dropper and Infection	mssecsvc.exe (lhdfrgui.exe)	Microsoft® Disk Defragmenter	24d004a104d4d54034dbcffc2a4b19a11f39008a575aa614ea04703480b1022c
	Resource Loader	tasksche.exe (diskpart.exe)	DiskPart	ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa
Destruction	Encryption DLL	kbdlv (3.13)	Latvia Keyboard Layout	1be0b96d502c268cb40da97a16952d89674a9329cb60bac81a96e01cf7356830
Command-and-control	Trace Infection and Payments	@WanaDecryptor@.exe (LODCTR.EXE)	Load PerfMon Counters	b9c5d4339809e0ad9a00d4d3dd26fdf44a32819a54abf846bb9b560d81391c25

The function "ExtractResource" aims for extracting "W/101" resource to create the mssecsvc.exe file at "C:\WINDOWS\mssecsvc.exe" and to launch it. The Windows API "CreateFileA" and "WriteFile" are used to write the loaded resource to "mssecsvc.exe," which is followed by a series of resource extraction routines, including "FindResourceA", "LoadResource", "LockResource", and "SizeofResource". Finally, the mssecsvc process is launched through calling "CreateProcessA" in the "CreateProcessMSSECSVC" sub-function.

B. *Installation Phase: Dropper, Infection, and Resource Loader*

The infection begins when the ransomware payload is delivered to the victim's machine. Once the payload successfully injects the launcher.dll into the lsass.exe system process, the dll launches mssecsvc.exe, which analyzes the system to determine whether it is located on a real computer or in a virtual sandbox [11]. Before any operation, two Windows API "InternetOpenA" and "InternetOpenUrlA" are used to query a hard-coded domain name, which in this sample, checks "www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com" (kill-switch URL). A successful connection will cause the mssecsvc.exe to terminate. Otherwise, it will proceed with the dropping of "tasksche.exe" and the infection (Fig. 4). The kill-switch in the execution flow provides an opportunity to slow down the malware [10].

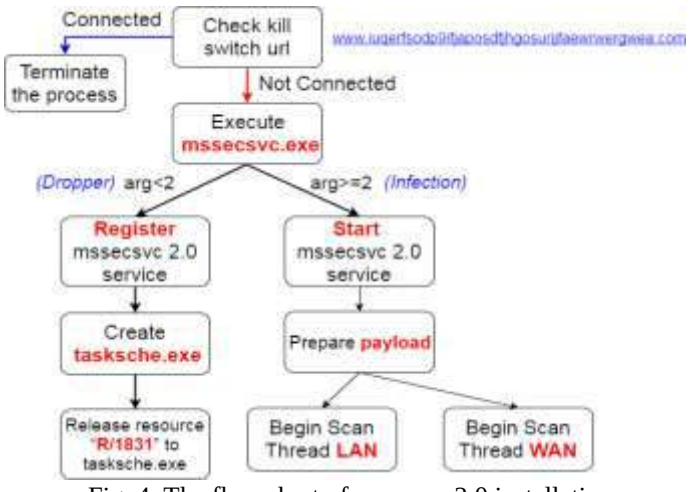

Fig. 4. The flow chart of mssecsvc2.0 installation

In the installation phase, "mssecsvc.exe" and "tasksche.exe" are two focused binaries. The "mssecsvc.exe" comprises two main execution functions, dropper and infection, and has a different entry point of execution depending on the command parameters (Algorithm1). The "tasksche.exe" is responsible for resource loading and the encryption environment setting.

Algorithm 1:

if (*argc* < 2)
then
InstallMssecsvc2.0Service();
ExtractResourceToTasksche(); *(Dropper Phase)*
else
Call StartServiceCtrlDispatcherA() to start mssecsvc2.0 service; *(Infection Phase)*

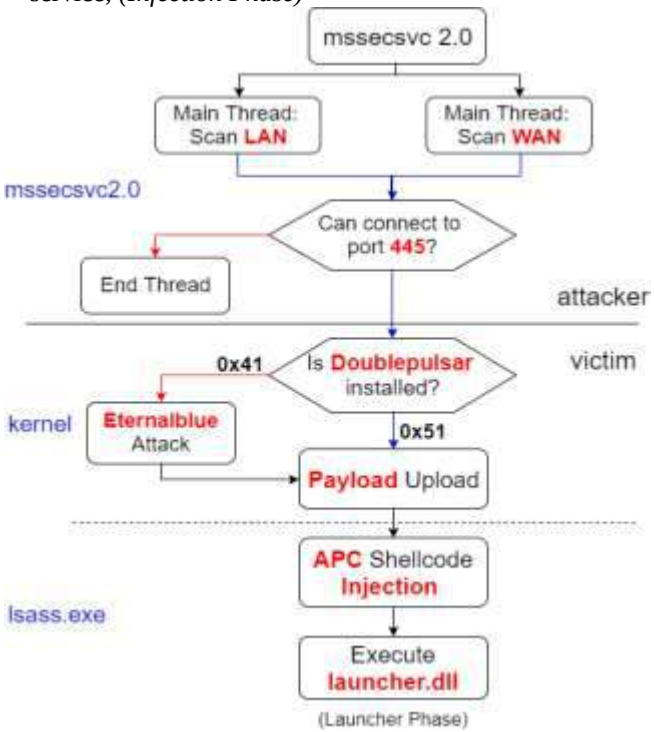

Fig. 5. The flow chart of infection

1) ***Dropper.*** At the beginning of execution, the "mssecsvc.exe" is run without any parameter. Two sub-functions are called to install the "mssecsvc2.0" service and to drop the next-stage binary "tasksche.exe". WannaCry is highly modular in a multi-stage campaign. This resource extraction routine is exactly a modular

example. After the complete extraction of the resource binary, the contents of the binary are written into "tasksche.exe".

2) ***Infection.*** If mssecsvc runs with a parameter "-m security", the execution falls to the infection function. The mssecsvc service is created to abuse the exploit of MS17-010 and the Doublepulsar backdoor for infection [8]. Fig. 5 summarizes WannaCry's infection flow, including the initial stage of the mssecsvc2.0 service that is running on the attackers' machine and the kernel Doublepulsr backdoor's implantation on the victims' machine. On the attacker machine, the mssecsvc2.0 service probes SMB protocol and port 445 [5]. If successfully connected, the attacker will be able to transmit the crafted packets with specific opcode to verify whether the Doublepulsar backdoor was set up to upload the payload. If the target machine has not had Doublepulsar backdoor installed (0x41 in response), the exploit code will proceed to initialize the Eternalblue attack. As soon as the setup of Doublepulsar is confirmed, the payload will be uploaded directly. The payload contains the kernel shellcode, userland shellcode, and launcher.dll with mssecsvc binary embedded in the resource section. The anatomy of the infection can also be revealed by analyzing the network packet using Wireshark.

3) ***Resource Loader.*** The main ransomware "tasksche.exe" is thrown by "mssecsvc.exe" into the dropper phase. It extracts the compressed XIA resource from its resource section, which contains several specific WannaCry files. While analyzing the tasksche reversing code, we organized the ransomware execution flow as depicted in Fig. 6. First, it generates a randomized unique ID for naming the folder that was prepared to contain the extracted resource. Second, the tasksche process verifies if a parameter exits prior to the execution of any operation. The command parameter "i" represents the installation mode of "tasksche.exe". After the installation, tasksche.exe is run without any parameter; further, a chain of functions is called to prepare for the encryption phase. It creates an autorun registry key as persistence mechanism, releases the resource zip file, and unzips it into the installation folder. Additionally, WannaCry uses the rand() function to randomly select one of the three hardcoded bitcoin addresses and writes it to c.wnry. Then, WannaCry adds the hidden attributes to the installation folder, grants complete access to all users, and decrypts the t.wnry binary to generate the encryption dll.

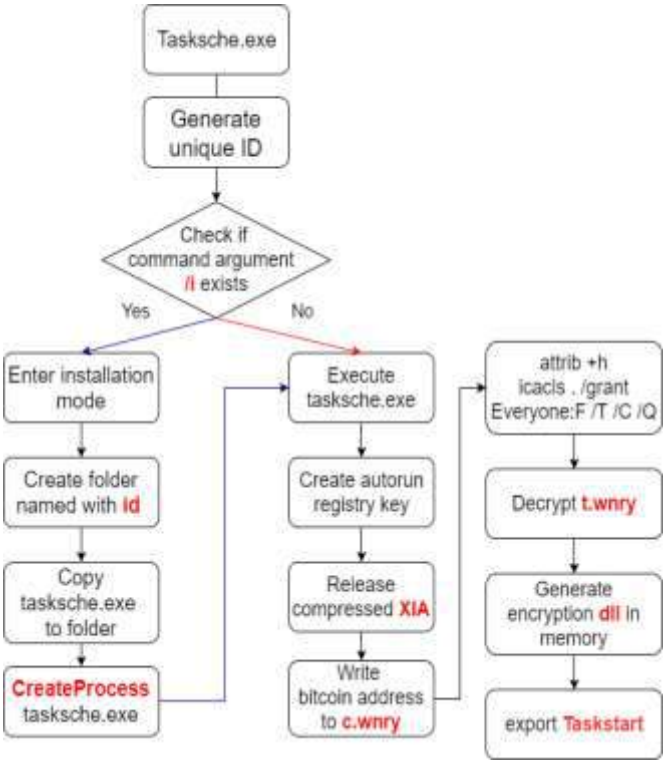

Fig. 6. The flow chart of main ransomware

The resource zip file "XIA" embedded in the resource section is extracted as the same resource loading process in the launcher and dropper phase. As soon as the resource loading completed, the code will unzip the resource file using the password "WNcry@2ol7". The XIA resource contains several WannaCry files, which are presented in Table II.

Table II

Files in XIA Resource

File Name	XIA Resource	
	File Description	***MD5***
msg\m_*.wnry	ransom notes in different languages	
b.wnry	display instructions for decryption	c17170262312f3be 7027bc2ca825bf0c
c.wnry	target address and TOR information	c17170262312f3be 7027bc2ca825bf0c
r.wnry	ransom note	c17170262312f3be 7027bc2ca825bf0c
s.wnry	TOR software executable	ad4c9de7c8c40813 f200ba1c2fa33083
t.wnry	encrypted ransomware DLL	ad4c9de7c8c40813 f200ba1c2fa33083
u.wnry	"@WanaDecryptor@.exe" decrypter file	7bf2b57f2a205768 755c07f238fb32cc
f.wnry	decrypt for demo	c17170262312f3be 7027bc2ca825bf0c
taskdl.exe	Enumerating and deleting temp files	4fef5e34143e646d bf9907c4374276f5
taskse.exe	Enumerate active RDP sessions and run a process on connected remote machines	8495400f199ac778 53c53b5a3f278f3e
@WanaDecryptor@.exe	Present user interface, C&C communication, and volume shadow deletion.	7bf2b57f2a205768 755c07f238fb32cc
00000000.eky	generated private key	6317124f38c33cce 36291ec3bc835db4
00000000.pky	generated public key	6f4e6640a2bc54a0 778130f7a25cb1b1
00000000.res	TOR/C2 information	168d54591c029609 959eb4256cbcea26

C. Destruction Phase: Encryption DLL

All the files on the victims' machines begin to be infected, encrypted, or locked by WannaCry. In the resource loader phase, tasksche.exe will throw a zip file from the resource section, which includes the t.wnry file. After a series of pre-processing tasks, tasksche.exe will decrypt the t.wnry into a dll exported TaskStart as the beginning of the encryption. The encryption flow and the key system are the two main themes that are closely related (Fig. 7). The encryption operation is heavily dependent on the management of the key system. WannaCry uses various pairs of keys to successfully form the encryption flow, including the RSA (Ron Rivest, Adi Shamir and Leonard Adleman) and AES (Advanced Encryption Standard) algorithms.

The key system begins from the RSA root public key, whose corresponding root private key is in the hands of WannaCry author. It is difficult for others to source the root key and to solve the encryption knot. A pair of RSA-2048 public and private keys is generated by the DLL, which respectively saves as 00000000.pky and 00000000.eky files. Prior to the RSA-2048 private key being saved to the 00000000.eky file, the private key is encrypted by the root public key in advance. For each targeted file, the encryption routine is initiated by the random creation of AES-128 encryption keys for different files. A unique AES-128 encryption key is also encrypted by the public key read from the 00000000.pky. During the generation of the encrypted file, the unique encrypted AES-128 key is embedded into the encrypted file's header followed by the 8-byte magic value "WANACRY!" and the 4-byte length of AES key as shown in Fig. 8.

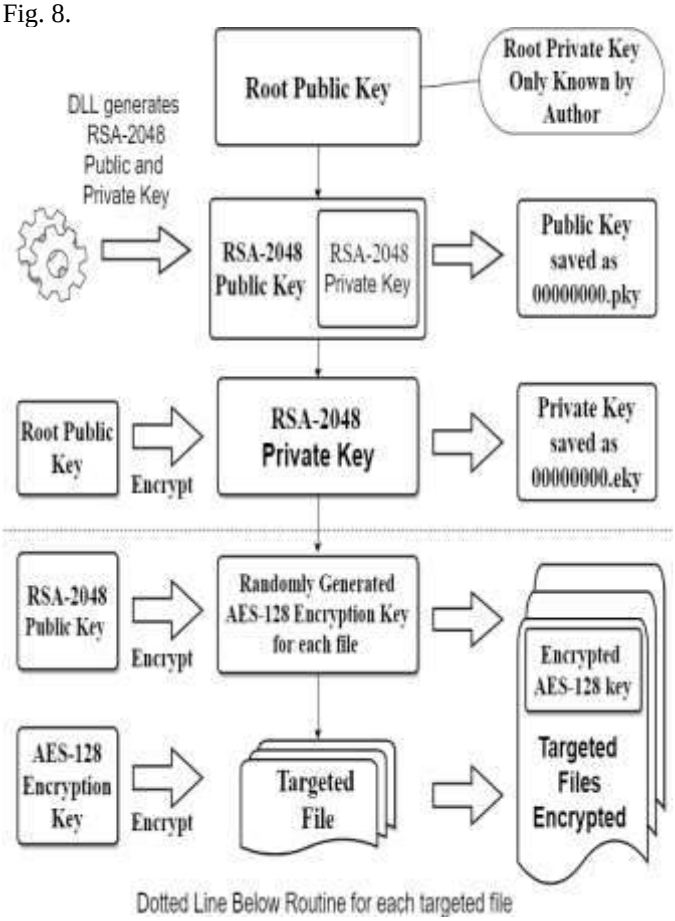

Fig. 7. The encrypting flow and key structure

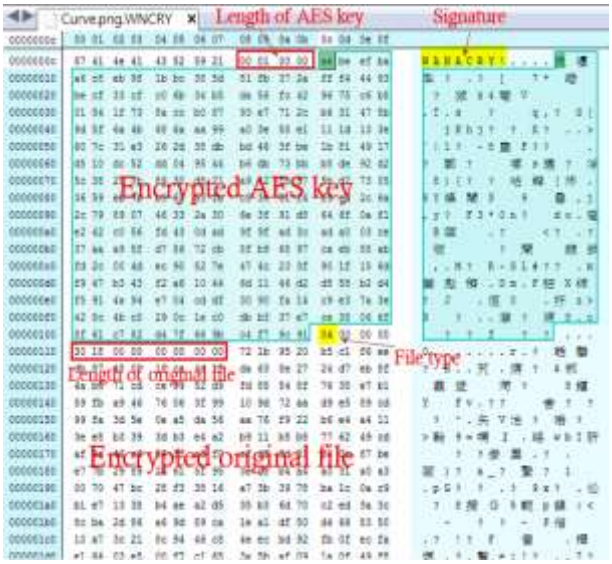

Fig. 8. The file structure of encrypted files

1) ***Delete or Erase Original Files.*** For the original files, the WannaCry will either delete or erase them depending on the file location in the system. WannaCry uses the wiping technique to prevent the user from restoring the files that are saved in the following folders: user desktop, user document, all user desktop, and all user document. The CryptGenRandom() function is called to generate random numbers, which are used to overwrite the original file contents. WannaCry will directly delete the files or move to a hidden recycle folder. The file deleting function taskdl is responsible for the deleting routine.

2) ***Demonstrate Decryption.*** When the machine is compromised with WannaCry, a random number of encrypted files in the folder C:\ ProgramData\<randomized unique ID>\f.wnry can be decrypted for free as a demonstration. The user can retrieve up to 10 original files in the decrypting demonstration. There is no guarantee that all the encrypted files can be decrypted through the ransom payment.

D. Command-and-Control Phase: Tracing of infection and management of payments

All actions require some form of command-and-control processes to determine the succeeding actions to be undertaken [7]. The "@WanaDecryptor@.exe" is the binary for the user interface, C&C communication, and volume shadow deletion. This binary can be run using one of three parameters: "fi", "co", or "vs". The malware installs the necessary library dependencies to execute the TOR service. If the "@WanaDecryptor@.exe" runs with the parameter "fi," it attempts to connect to the onion server (C&C) and send the user name, host name and some other information about the infected system. If the parameter "co" is delivered, it launches "taskhsvc" as a sub-process to communicate with the onion server. The response from the C&C server should include a unique bitcoin address,which will update the string in c.wnry. The onion server domains are listed in the "c.wnry" as follows:

1) ***gx7ekbenv2riucmf.onion***

2) ***57g7spgrzlojinas.onion***

3) ***xxlvbrloxvriy2c5.onion***

4) ***76jdd2ir2embyv47.onion***

5) ***cwwnhwhlz52maqm7.onion***

IV. Network Analysis on WannaCry Exploits

To dissect WannaCry's exploits, network analysis was conductd by examining the network packets that were observed between the propagated machine and an infected machine. To perform such analysis, a VMware Workstation was adopted to build two host-only machines and to configure them in the same LAN. The captured packets were analyzed through Wireshark.

Once a machine with an open NetBIOS port was observed, WannaCry will gain a TCP socket for port 445, connect to SMB socket, and obtain an SMB tree ID for later use. Another characteristic is WannaCry's transmission of three NetBIOS session setup packets to it. One has the proper IP address (192.168.135.131 in our experiment) of the machine being exploited. Others contain two IP addresses (192.168.56.20 and 172.16.99.5) hard-coded in the malware body. The phenomenon and characteristic of the hard-coded IP addresses were probed for the target system's exploit status [11].

A. MS17-010 SMB RCE Detection

The detection method of information disclosure was used to determine if the MS17-010 has been patched [11]. WannaCry connected to the IPC$ tree and attempted a transaction on FID 0. If the status returned is "STATUS_INSUFF_SERVER_RESOURCES", it indicated that the machine did not have the MS17-010 patch (Fig. 9).

```
SMB          142 Negotiate Protocol Request
SMB          143 Negotiate Protocol Response
SMB          157 Session Setup AndX Request, User: .\
SMB          146 Session Setup AndX Response
SMB          149 Tree Connect AndX Request, Path: \\192.168.135.131\IPC$
SMB          104 Tree Connect AndX Response
SMB Pipe     132 PeekNamedPipe Request, FID: 0x0000
SMB           93 Trans Response, Error: STATUS_INSUFF_SERVER_RESOURCES
```

Fig. 9. The detection packets

B. SMB Doublepulsar Probe

The intent of the SESSION SETUP Trans2 Request was to verify if the system had already been compromised with the Doublepulsar backdoor (Fig. 10).

```
SMB     191 Negotiate Protocol Request
SMB     187 Negotiate Protocol Response
SMB     194 Session Setup AndX Request, User: anonymous
SMB     193 Session Setup AndX Response
SMB     150 Tree Connect AndX Request, Path: \\192.168.56.20\IPC$
SMB     114 Tree Connect AndX Response
SMB     136 Trans2 Request, SESSION_SETUP
SMB      93 Trans2 Response, SESSION_SETUP, Error: STATUS_NOT_IMPLEMENTED
```

Fig. 10. The probing packets by Wireshark

If the field "Multiplex ID" is equal to 65(0x41), it indicates the current system is normal systems (Fig. 11). Otherwise, "Multiplex ID" that is equal to 81(0x51) indicates that the system has already been infected with Doublepulsar backdoor.

```
SMB (Server Message Block Protocol)
 SMB Header
     Server Component: SMB
     [Response to: 589]
     [Time from request: 0.000442000 seconds]
     SMB Command: Trans2 (0x32)
     NT Status: STATUS_NOT_IMPLEMENTED (0xc0000002)
   > Flags: 0x98, Request/Response, Canonicalized P
   > Flags2: 0xc007, Unicode Strings, Error Code Ty
     Process ID High: 0
     Signature: 0000000000000000
     Reserved: 0000
   > Tree ID: 2048 (\\192.168.56.20\IPC$)
     Process ID: 65279
     User ID: 2048
     Multiplex ID: 65
```

Fig. 11. The SMB Header of Trans2 Response

C. Triggering the Vulnerability

If the detection result shows that the target contains MS17-010 vulnerability and it is not yet infected with the Doublepulsar backdoor, it will proceed to install a Doublepulsar backdoor through the Eternalblue exploit (Fig. 12).

```
SMB     191 Negotiate Protocol Request
SMB     161 Negotiate Protocol Response
SMB     194 Session Setup AndX Request, User: anonymous
SMB     243 Session Setup AndX Response
SMB     146 Tree Connect AndX Request, Path: \\172.16.99.5\IPC$
SMB     114 Tree Connect AndX Response
SMB    1138 NT Trans Request, <unknown>
SMB      93 NT Trans Response, <unknown (0)>
```

Fig. 12. The vulnerability triggering packets by Wireshark

An initial NT Trans request comprised a sequence of NOPs, which sought for the vulnerabilities in the compromised devices. The attacker could leverage a specialized-crafted packet to exploit targets' SMB protocol (Fig. 13). The large NT Trans request caused multiple Secondary Trans Requests and served as indicators for attackers to trigger the vulnerabilities.

```
0000  00 0c 29 c2 35 42 00 0c  29 3c 08 cc 08 00 45 00   ..).5B.. )<....E.
0010  04 64 07 fd 40 00 80 06  00 00 c0 a8 87 a8 c0 a8   .d..@... ........
0020  87 a3 c1 68 01 bd 98 de  1b 7d 22 f3 84 9f 50 18   ...h.... .}"...P.
0030  00 ff 94 f3 00 00 00 00  04 38 ff 53 4d 42 a0 00   ........ .8.SMB..
0040  00 00 00 18 07 c0 00 00  00 00 00 00 00 00 00 00   ........ ........
0050  00 00 00 08 ff fe 00 08  40 00 14 01 00 00 1e 00   ........ @.......
0060  00 00 d0 03 01 00 1e 00  00 00 00 00 00 00 1e 00   ........ ........
0070  00 00 4b 00 00 00 d0 03  00 00 68 00 00 00 01 00   ..K..... ..h.....
0080  00 00 00 ec 03 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0090  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00a0  00 00 00 00 01 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00b0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00d0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00e0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0100  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0110  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0120  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0130  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0140  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0150  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0160  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
```

Fig. 13. The large sequence of NOPs

D. Doublepulsar Instrcution

After the completion of Eternalblue attack, the execution control was transferred to the Doubepulsar backdoor. A series of SMB packets were transmitted between the WannaCry propagating machine and the targeted victim, and the Doublepulsar instructions were hidden in specific fields (Fig. 14 and Table III).

Fig. 14. Doublepulsar instruction process

TABLE III
Doublepulsar Instruction Details

Doublepulsar Instruction	Details of Instruction		
	opcode	Hidden Field	Description
Ping Request	0x23	Timeout	Check if Doublepulsar backdoor exits.
Ping Response: Infected	0x81	Multiplex ID (MID)	Respond from Doublepulsar backdoor
Exec Request	0xC8	Timeout	Upload payload and inject.
Exec Response: Completed	0x82	Multiplex ID (MID)	Complete

1) ***Ping Request.*** After the initial negotiation and session setup, WannaCry will send a ping request to the target by sending multiple ping packets to the compromised system. The purpose of ping request was to check if the hook of Doublepulsar was installed successfully. The "ping" instruction was hidden in the "Timeout" field, which was originally the amount of time that the client had to wait for the server to respond to an outstanding request (Fig. 15). According to the Microsoft Open Specifications, the default value of Timeout field was set to 45 seconds. In the WannaCry network packet, the Timeout field was set to 4 hours 20 minutes 10.881 seconds (0x00ee3401). This abnormal Timeout value did not actually refer to the time out set but it implied the Doublepulsar instruction opcode. The algorithm of calculating this opcode is adding each byte and removing the overflow as result. If the Doublepulsar backdoor has successfully installed on the infected system, it will send back a crafted packet with "Multiplex ID" field purposely set..

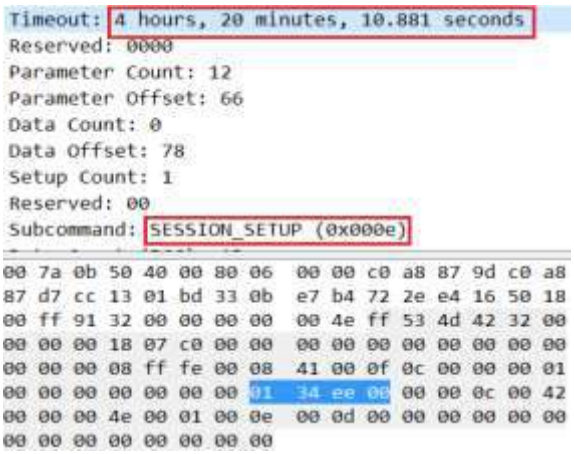

Fig. 15. "Ping" command in hidden Timeout field

2) ***Ping Response: Infected.*** While the Doublepulsar backdoor responded to the "ping" command with the field "Multiplex ID (MID)" set to 0x81, it implied the presence of itself. This packet had another implication using the "Signature" field (Fig. 16), which was set to value 0x011f7a1332. For little-endian, the first byte was set to 0x01, which indicated the machine was developed on an x64 platform. The WannaCry will prepare the payloads according to this probing result. For the remaining four bytes (0x1f7a1332), the encrypted XOR key was used to encode the payload during the uploading stage. The XOR key decrypting routine was conducted before WannaCry started using the XOR key to encode payload. The decrypting algorithm is demonstrated through IDA Pro reversing (Algorithm2).

Algorithm 2: (a1 = encrypted XOR key)
Decrypted XOR key = 2 * a1 ^ ((((a1 >> 16) | a1 & 0xFF0000) >> 8) | (((a1 << 16) | a1 & 0xFF00) << 8))

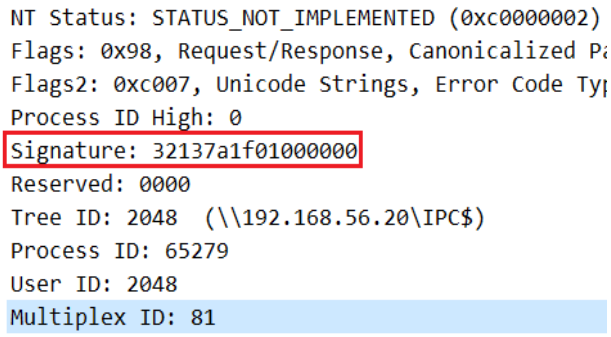

Fig. 16. Hidden Response in MID field with Signature field set to contain XOR key

3) ***Exec Request.*** After the confirming the presence of the backdoor, WannaCry will resume sending the "exec" Doublepulsar command to the target and ordered the backdoor on the target to start the injection of the ransomware into the lsass process. As indicated in the "ping" command, the packet set the "Timeout" field to an abnormal value. In the "exec" command, the "Timeout" field was again set to the value 0x001a8925 (Fig. 17).

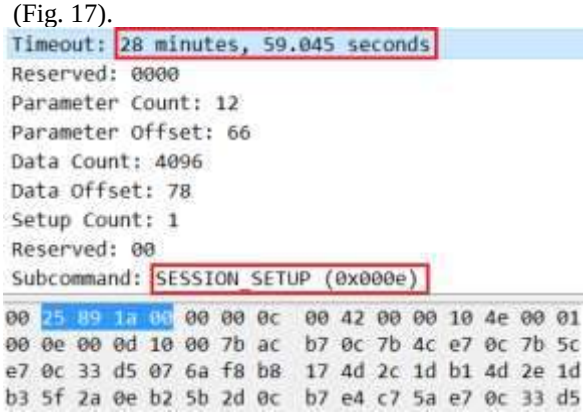

Fig. 17. "Exec" command hidden in Timeout field

4) ***Exec Response: Completed.*** As the shellcode completed, the Doublepulsar backdoor will send a packet with the field MID set to 0x82, to signal the completion of the task (Fig. 18).

```
NT Status: STATUS_NOT_IMPLEMENTED (0xc0000002)
Flags: 0x98, Request/Response, Canonicalized Pa
Flags2: 0xc007, Unicode Strings, Error Code Typ
Process ID High: 0
Signature: 0000000000000000
Reserved: 0000
Tree ID: 2048  (\\192.168.56.20\IPC$)
Process ID: 65279
User ID: 2048
Multiplex ID: 82
```

Fig. 18. Task completed

V. Research Finding

A. Multi-Phased WannaCry Execution

The increasing modularization of ransomware has encouraged security researchers to explore the inner design of each binary module. The typical resource extracting module used by WannaCry is the key to enable multi-phased execution. The exploit module provides WannaCry with an opportunity to rapidly propagate across the Internet. The WannaCry ransomware follows an execution flow when it gains access to a system and starts the propagation and encryption of files. Moreover, it inflicts damage by executing a series of tasks [10]. A cyclical life cycle exists throughout the entirety of the WannaCry code (Fig. 9). In this research, the anatomy of ransomware attack has been grouped into four phases in this research finding [1, 7]: *deployment, installation, destruction, and command-and-control.* In each phase, the component behavior is determined by the process parameter. The processes with their parameters are summarized in Table IV.

1) ***Deployment Phase.*** The launcher.dll is remotely injected into the lsass process through the infamous Eternalblue exploit and Doublepulsar backdoor. Launcher exports the PlayGame function, which uses resource-manipulation API functions, such as FindResource, LoadResource, LockResource, and SizeofResource to initialize the embedded mssecsvc binary in the launcher.dll resource section. And then, the mssecsvc process is launched through the path of "C:\Windows\mssecsvc.exe".
2) ***Installation Phase.*** This phase comprises two components, "mssecsvc.exe" and "tasksche.exe". The mssecsvc.exe starts up the mssecsvc2.0 service for propagation and drops the "tasksche.exe" using the same resource-manipulation API functions and routines in the deployment phase. The tasksche.exe is responsible for resource loading, environment setting, and the decryption of t.wnry. The "mssecsvc.exe" starts propagating while the parameter "m security" is identified. The propagation process has been categorized into the following four stages: *MS17-010 SMB RCE detection, SMB Doublepulsar probe, triggering the vulnerability, and Doublepulsar instruction*.
3) ***Destruction Phase.*** In the destruction phase, tasksche decrypts t.wnry from its resource section and loads the encryption dll in memory to execute the tasks. The encryption dll exports TaskStart to initiate the encryption. The management of the key system creates a complex encryption knot. For each encrypted target, an AES-128 encryption key is generated, which is also encrypted by the public key read from the 00000000.pky

Table IV

Execution Phase with Main Processes and their Features

Execution Phase	Processes	Features	
		parameter	***operation***
Deployment	launcher dll in lsass	*N/A*	Export PlayGame which loads resource into the mssecsvc binary and launch it.
Installation	mssecsvc	*N/A*	Install mssecsvc2.0 service for propagation and load resources into the tasksche binary before launching it.
		m security	Scan for devices both locally and on the Internet, exposes port 445, exploits the MS17-010 vulnerability, and installs the Doublepulsar backdoor.
	tasksche	*i*	Imply the installation mode of tasksche. It first creates a working directory C:\Windows\ProgamData\<randomized_id>\tasksche.exe to store its released binaries. After installation, the tasksche will then be executed without parameters.
		N/A	Create the registry key: HKLM\Software\Wow6432Node\Microsoft\Windows\CurrentVersion\Run\ <randomized_id>, release XIA resource, get TOR configuration from c.wnry, and run command "attrib +h" and "icacls . /grant Everyone:F /T /C /Q".
Destruction	encryption dll in tasksche	*N/A*	Decrypt t.wnry into encryption dll and export TaskStart to run.
Command-and-Control	@WanaDecryptor@	*N/A*	Present the ransomware user interface.
		fi	Attempt to connect to the onion server (C&C) in the dark web and send the user name, host name, and some information about the infected system. The response may include an updated bitcoin address in c.wnry.
		co	Launch "taskhsvc" as sub-process to do the communication with onion server (C&C) and send some information about encrypting the users' files from 00000000.res, including end time of encryption, the amount, the size of encryption
		vs	Delete volume shadow copies utilizing the Windows built-in vssadmin utility. It will launch the following command as sub-process "vssadmin.exe delete shadows /all /quiet" to implement the shadow deleting utililty.

4) **Command-and-Control Phase.** @WanaDecryptor@.exe is responsible for command-and-control. WannaCry tracks the payment and transmits the encryption information back to the onion servers in the command-and-control phase. The main execution flow is shown in Fig. 19.

B. WannaCry exploit signatures

During the initial exploitation, WannaCry will do the SMB tree connection, which contains the packet contents with the SMB header of "SMBr" (0x534D4272), "SMBs" (0x534D4273), "SMBu" (0x534D4275), and "SMB2" (0x534D4232). In addition, two hardcoded IP addresses are used to do the null connection for information disclosure. Therefore, the unique patterns of packets, and the hardcoded IP addresses can be used to generate the Yara rule (Rule: WannaCry_exploits).

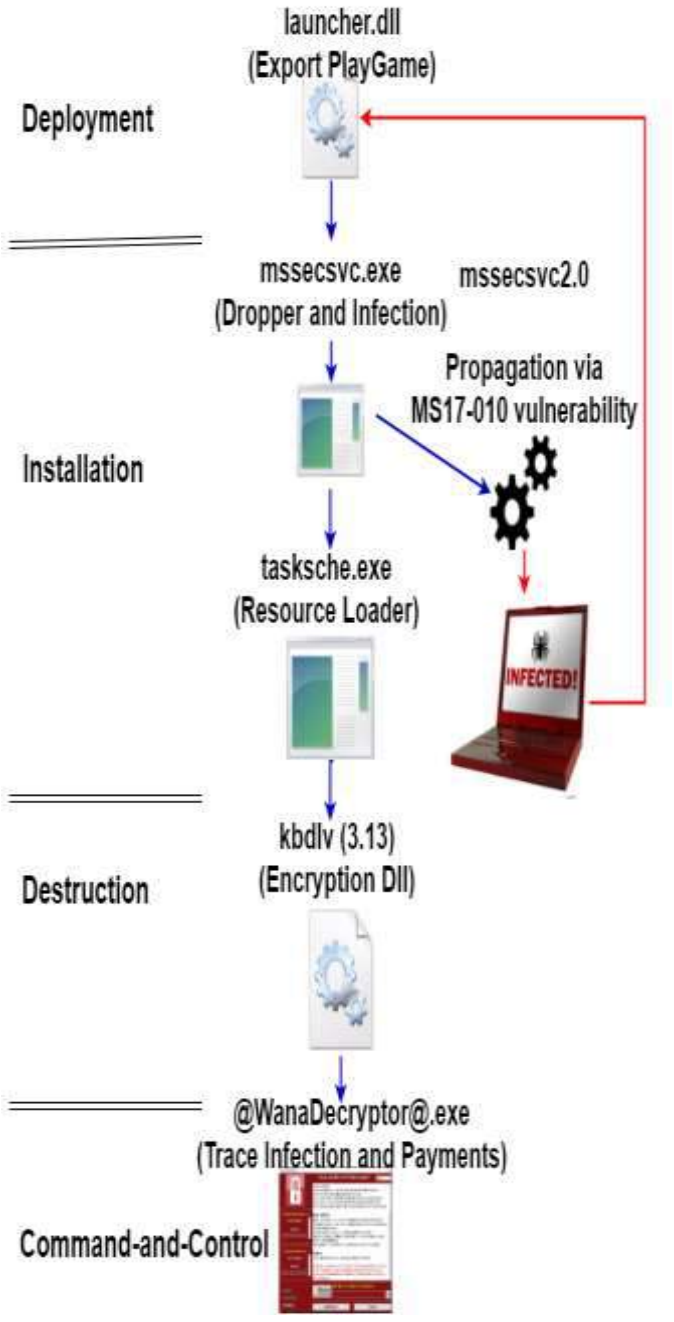

Fig. 19. Main execution flow of WannaCry ransomware

```
Rule: WannaCry_exploits{
    meta:
      description = "Detect WannaCry propagation"
    strings:
      $op1 = {53 4D 42 72 00 00 00 00 18 53 C0 00 00 00 00 00
    00 00 00 00 00 00 00 00 00 00 FF FE 00 00 40 00  00 62 00
    02 50 43 20 4E 45 54 57 4F 52 4B 20 50  52 4F 47 52 41}
      $op2 = {53 4D 42 73 00 00 00 00 18 07 C0 00 00 00 00 00
    00 00 00 00  00 00 00 00 00 00 FF FE 00 00 40 00 0D FF
    00 88  00 04 11 0A 00 00 00 00 00 00 00 01 00 00 00 00}
      $op3 = {53 4D 42 75 00 00 00 00 18 07 C0 00 00 00 00 00
    00 00 00 00 00 00 00 00 00 00 FF FE 00 08 40 00 04 FF 00
    5C 00 08 00 01 00 31 00 00 5C 00 5C 00 31 00 39 00 32}
      $op4 = {53 4D 42 32 00 00 00 00 18 07 C0 00 00 00 00 00
    00 00 00 00 00 00 00 00 00 08 FF FE 00 08 41 00 0F 0C 00
    00 00 01 00 00 00 00 00 00 00 01 34 EE 00 00 00 0C 00}

      $s1 = "\\\\192.168.56.20\\IPC$" fullword wide
      $s2 = "\\\\172.16.99.5\\IPC$" fullword wide

    condition:
        uint16(0) == 0x5a4d and all of ($s*) and 2 of ($op*)
    and pe.imports("ws2_32.dll", "connect") and
    pe.imports("ws2_32.dll", "send") and
    pe.imports("ws2_32.dll", "recv") and
    pe.imports("ws2_32.dll", "socket")
}
```

VI. CONCLUSIONS

The WannaCry outbreak is a significant security incident that spurs everyone to seriously consider the fundamentals of patching computers to current status. As malware developers tend to apply modular hacking weapons to new variants of malware, the detection technique adopted by security defenders becomes more granular based on the composed hacking weapon binaries. A thorough malware analysis was conducted to (i) identify the malicious binary, (ii) examine the exploits, (iii) collect malicious patterns, (iv) understand the indicators of compromised situation, and (v) report the observations to ensure the formulation of future defense strategies.

This paper conducted the reverse engineering analysis on WannaCry's components, and a network analysis on the WannaCry exploits. The modular hacking weapon in each component and its execution flow were dissected and analyzed. In addition, the techniques used by WannaCry exploits were unveiled by examining the packet details. The research findings, including representative hacking weapon modules and network signatures, can be documented to develop future defense strategies. The trend of integrating the artificial intelligence into unknown malware detection has become a popular issue. It may become a possible extension of our research for future ransomware detection based on the features of these hacking weapons.

ACKNOWLEDGMENT

The authors would like to thank Enago for the English language review.

REFERENCES

[1] Awad, R. A. and Sayre, K. D., "Automatic Clustering of Malware Variants," *2016 IEEE Conference on Intelligence and Security Informatics (ISI 2016)*, pp. 298–303, 2016.

[2] Ceron, J. M., Margi, C. B., and Granville, L. Z., "MARS: An SDN-based Malware Analysis Solution," *2016 IEEE Symposium on Computers and Communication (ISCC)*, pp. 525–530, June 2016.

[3] Fujino, A., Murakami, J., and Mori, T., "Discovering Similar Malware Samples Using API Call Topics," *2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC)*, pp. 140–147, Jan 2015.

[4] Hansen, S. S., Larsen, T. M. T., Stevanovic, M., and Pedersen, J. M., "An Approach for Detection and Family Classification of Malware Based on Behavioral Analysis," *2016 International Conference on Computing, Networking and Communications (ICNC)*, pp. 1–5, Feb 2016.

[5] Islam, A., Oppenheim, N., and Thomas, W., "SMB Exploited: WannaCry Use of Eternalblue." [Online]. Available: https://www.fireeye.com/blog/threat-research/2017/05/smb-exploited-WannaCry-use-of-Eternalblue.html

[6] Kharaz, A., Arshad, S., Mulliner, C., Robertson, W., and Kirda, E., "UNVEIL: A Large-Scale, Automated Approach to Detecting Ransomware," *25th USENIX Security Symposium (USENIX Security 16)*, pp. 757–772, USENIX Association, 2016.

[7] Liska, A. and Gallo, T., *Ransomware: Defending Against Digital Extortion*, 1st Edition, O'Reilly Media Inc., pp. 1-22, 2016.

[8] Microsoft, "Microsoft Security Bulletin MS17-010 - Critical: Security Update for Microsoft Windows SMB Server (4013389)." [Online]. Available: https://technet.microsoft.com/en-us/library/security/ms17-010.aspx

[9] Mosli, R., Li, R., Yuan, B., and Pan, Y., "Automated Malware Detection Using Artifacts in Forensic Memory Images," *in 2016 IEEE Symposium on Technologies for Homeland Security (HST)*, pp. 1–6, May 2016.

[10] Rousseau, A., "WCry/WanaCry Ransomware Technical Analysis." [Online]. Available: https://www.endgame.com/blog/wcrywanacry-ransomware-technical-analysis

[11] Rudman, L. and Irwin, B., "Dridex: Analysis of the Traffic and Automatic Generation of IOCs," *2016 Information Security for South Africa (ISSA)*, pp. 77–84, Aug 2016.

Da-Yu Kao received the B.S. and M.S. degree in information management from Central Police University, Taiwan, in 1993 and 2001, the Ph.D degrees in Crime Prevention and Correction from Central Police University, Taiwan, in 2009, respectively. From 1993 to 1996, he was with Taipei City Police Department, Taiwan, where he was an information technology police officer involved in the development of policing information systems. From 1996 to 2007, he was with Criminal Investigation Bureau, National Police Administration, Taiwan, where he was a detective and forensic police officer in cybercrime investigation and digital forensics. From 2007 to 2013, he was with Maritime Patrol Directorate General, Coast Guard Administration, Taiwan, where he was an information technology section chief in the department of information and communication. Since 2013, he has been with Central Police University, Taiwan, where he is currently an associate professor in the Department of Information Management. His research interests include cybercrime investigation, digital forensics, digital evidence, information management, criminal profiling, and cyber criminology.

Shou-Ching Hsiao received the B.S. degree in information management from Central Police University, Taiwan, in 2016. Since 2016, she has been with Haishan Precinct, New Taipei City Police Department, Taiwan, where she is currently an information lieutenant and is responsible for information system management, information security, malware analysis, and real-time video for security control. In 2018, she is also working toward the M.S. degree in the Department of Computer Science, National Chengchi University, Taiwan. Her current research interests include malware analysis, cybercrime investigation, digital forensics, digital evidence, information management, criminal profiling, and cyber criminology.

Raylin Tso received the B.S. degree in Industrial Engineering from National Tsing Hua University, Hsinchu, Taiwan and M.S. degree in Management Science - Division of Management and Public Policy, University of Tsukuba, Tsukuba, Ibaraki, Japan, in 2002, M.S. degree in Risk Engineering, Graduate School of Systems and Information Engineering, University of Tsukuba, Tsukuba, Ibaraki, Japan, in 2004, the Ph.D degrees in Risk Engineering, Graduate School of Systems, University of Tsukuba, Tsukuba, Ibaraki, Japan, in 2006, respectively. From 2006 to 2008, he was with Graduate School of Systems and Information Engineering, University of Tsukuba, Japan, where he was an academic researcher. Since 2008, he has been with National Chengchi University, Taiwan, where he is currently an associate professor and chairman in the Department of Computer Science. His research interests include cryptography, information security, post-quantum, blockchain and privacy enhancement.