Sri Lanka Institute of Information Technology

Malware Analysis Report

Information Warfare – IE4032

Assignment

Submitted by Thomas R.L - IT19095936

Malware Analysis Report: WannaCry Malware

IT19095936 R.L.Thomas

***Abstract*—** known as crypto-ransomware, WannaCry is a self-propagating form of malware that spread across the internet and infected more than 200,000 computers. Malicious software that is specifically created to cause harm to a computer system is known as malware. A malicious computer program known as a crypto-ransomware encrypts user files in order to demand payment. Additionally, a computer worm is included in WannaCry. Electronic money known as Bitcoin allows for anonymous transactions. WannaCry also communicates with its command and control servers via the Tor network. The communications between a malware operator and the malware itself are referred to as command and control. In this paper, we look into the WannaCry ransomware component. WannaCry, analysis, ransomware, encryption, files, network, SMB, vulnerability, motivation are all index terms.

1. Introduction

WannaCry Malware

WannaCry is a type of malware known as a crypto ransomware, which is characterized as software that encrypts users' files and demands payment in order to decrypt them. This type of malware is designed to extort money [1]. A three-day countdown and a threat that the user's decryption key would be deleted if he didn't pay on time are two examples of how crypto-ransomware employs shock and fear to persuade users to pay the demanded ransom [1]. Because it can spread itself via computer networks, WannaCry is also referred to as a network worm. On May 12, 2017, WannaCry was first observed in the wild. The WannaCry ransomware outbreak is regarded as the largest in human history. In more than 150 countries, it had infected over 200,000 computers. A worm module for self-propagation and ransomware for file encryption make up WannaCry [3]. WannaCry's C&C (command and control) communications utilize Tor hidden services. In WannaCry, the C&C is primarily used to determine whether the victim has paid the ransom and to deliver the decryption key [2].

Worm Module

Because of the worm module, WannaCry is referred to as a self-propagating piece of malware. This module is only used to spread itself to all computers connected to the internal and external networks through propagation. We will talk about the malware's exploit as well as the propagation process in this section.

SMB Vulnerability

The Server Message Block (SMB) protocol's flaw in Windows' implementation is exploited by the WannaCry malware. SMB is a transport protocol that Windows uses for sharing files, sharing printers, and accessing remote services. The SMB protocol uses TCP ports 139 and 445 for operation. The SMB v1 vulnerability and TCP port 445 are both used by the malware to spread. This vulnerability enables remote attackers to run arbitrary code on the victim's computer using malformed packets.

2. Preparing Sandbox

A sandbox enables the safe detection and analysis of online threats. The system cannot be accessed by a suspicious file, and all data is kept secure. Malware processes can be tracked, their patterns studied, and their behavior examined.

Creating VM for Malware Analysis

Installing & Configuring Virtual Machine

To prevent infecting the host operating system, running malware should take place in a properly isolated environment. Although having a separate computer is preferable, you can set up one or more virtual machines with various OS versions. There are many virtual machines (VMs) available, including VirtualBox, KVM, Oracle VM VirtualBox, Microsoft Hyper-V, Parallels, and Xen. Modern malware is intelligent; it can tell whether a virtual machine is being used to run it or not. Artifact removal is crucial for this reason. Verify code, disable detection, and other things.

Assign a realistic amount of resources

Making a system appear as authentic as possible is our aim in order to get any malicious program to run. Make sure to allocate a reasonable amount of resources, such as 100 GB or more of disk space, at least 4 cores, and more RAM than 4 GB. To pass for a legitimate system, that is a necessary prerequisite. Still, bear in mind that malware examines the setup of the system. The name of a virtual machine must be present somewhere for a malicious object to recognize it and stop functioning. A malicious object will be analyzed if you install Windows and leave it unattended.

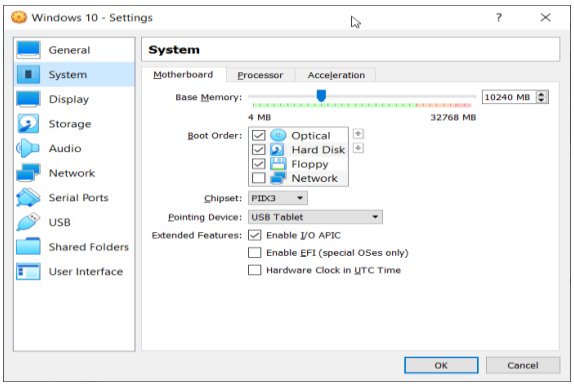

Figure 1 Setting up resource

Install commonly used software

Install a few commonly used programs, such as Word, browsers, and other programs. Here, we must demonstrate that the computer in question is a genuine personal device. Open a few files to gather logs and some temporary files. Numerous virus types examine this. Making a log of changes to the file system and registry can be done using Regshot or Process Monitor. Keep in mind that while these programs are running, malware can detect them.

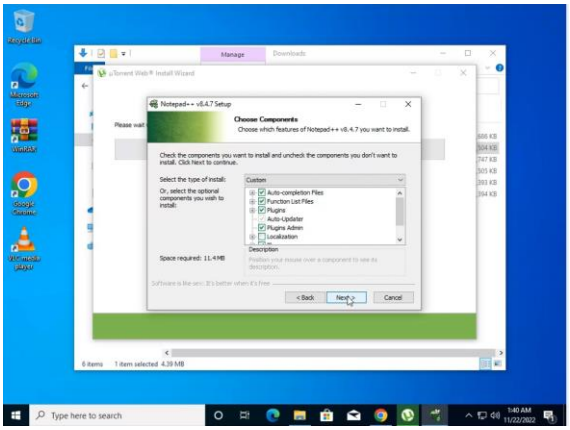

Figure 2 Setting up the softwares

Simulate an internet connection

Some malware types test their ability to connect to websites like Google. How can one make a malicious program believe it is online? We are able to intercept the requests that malware is making thanks to tools like INetSim and the FakeNet tool, which pretend to be a

genuine Internet connection. Network protocols between a malicious object and its host server should be checked. But first, use WireShark to determine what the analyzed sample is connecting to. Additionally, it requires some effort to prevent this tool from being taken over by malware.

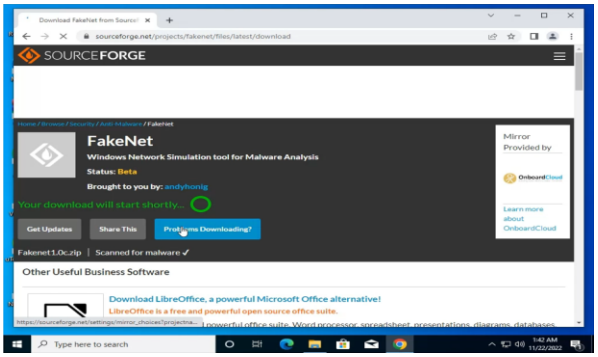

Figure 3 Setting up fakenet internet simulator

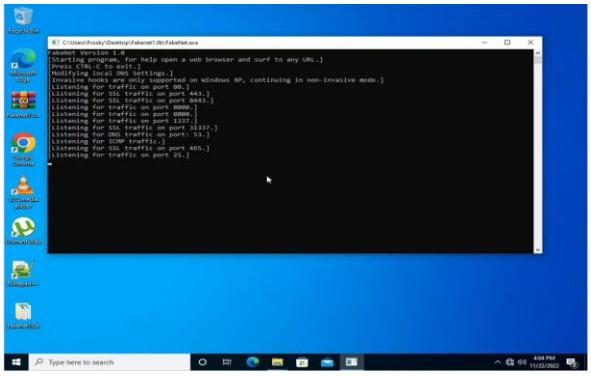

Figure 4 Setting up fakenet internet simulator 2

Configuring Network settings

Making use of a different network system is an additional precaution. It's crucial to keep any other computers on your network from becoming infected. Get a VPN service and configure it properly. You cannot permit a real IP address to be the source of the traffic leak. We must modify the network settings of the operating system we installed using the virtualization software to stop the malware we will analyze from infecting other devices on the network. The "Host-only" option must be chosen when entering the "Network" settings in the settings section.

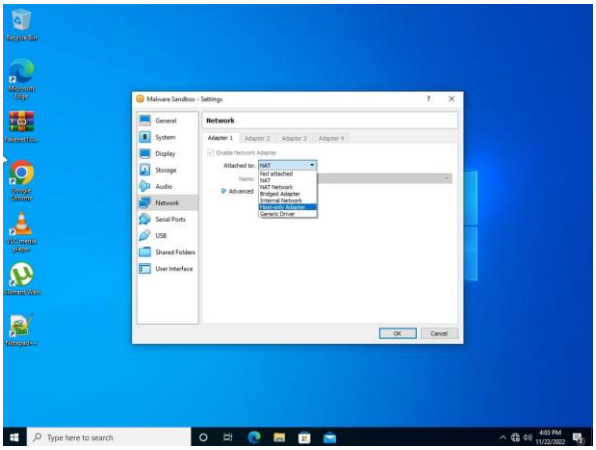

Figure 5 Sandbox network configuration

- NAT: Allows you to access the Internet through the network interface of your physical device.
- Bridge: Allows you to access the internet by obtaining its own IP address from your modem like your physical device.
- Host-only: It is included in the private network created by the virtualization environment. Internet access is not available in this option.

To prevent malware from infecting additional devices on the same network, you should make the virtual operating system appropriate for malware analysis.

Disable Antivirus Software

To stop antivirus software from interrupting our analysis by blocking or removing the malware we want to examine, we must turn it off.

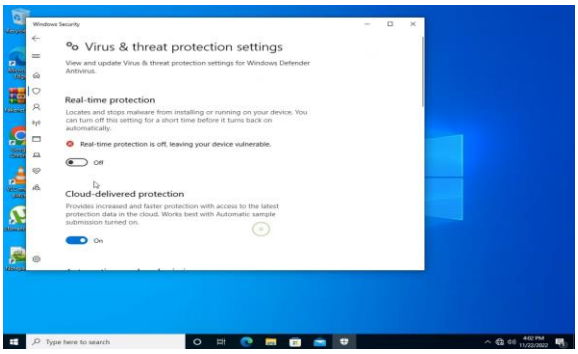

Figure 6 Disabling Windows defender antivirus

Disable Firewall

Disable the firewall from blocking the malware from accessing the internet so we can carefully analyse the packets being sent.

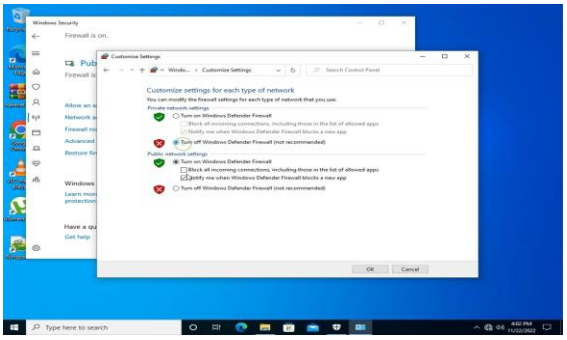

Figure 7 Disabling Firewall in windows 10

Disable Updates

Multiple vulnerabilities may be used by malware. We must stop our virtual operating system from getting security updates while performing our dynamic analysis so that the malware can continue to operate and successfully exploit such flaws. We must therefore disable our operating system's automatic update feature.

Disable Hidden Extensions

In the Windows operating system, well-known file extensions are by default hidden. To see the exact name of the file we want to analyze, we must disable this feature.

Disable Hidden Files & Folders

The Windows operating system does not by default display hidden files. Malware makes use of this feature to make it difficult to detect. We must disable this feature in order to clearly see what is happening in the file system.

Snapshots

Malicious software changes the system in a number of ways when it is run. While analyzing a new malware, if you do not restore the operating system to its initial state, you might mistake it for the malware you previously ran. Installing a fresh virtual operating system each time we want to study malware will be very challenging. Our work is made very simple by the virtualization software's Snapshot feature.

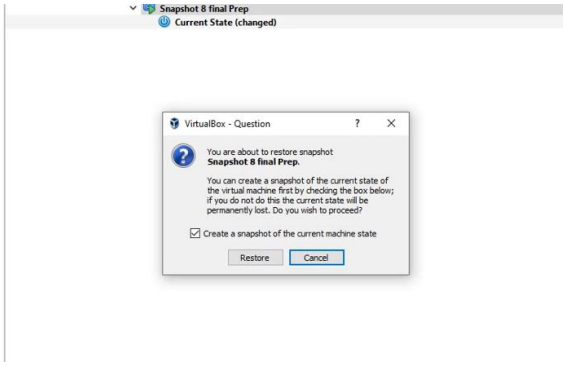

Figure 8 creating a snapshot in VirtualBox

The current state of your virtual device is saved when you take a snapshot of it using the virtualization environment. The device will then be restored by going back to this snapshot. You can take a snapshot after installing the necessary malware analysis tools, analyze the malware, and then restore the operating system to its initial state using this snapshot.

3. Prepare the tools

Disassemblers

A program must be translated into 0s and 1s that the machine can understand in order for it to be run on a computer. Compilation languages like C and C++ are examples of such languages. Compile is the name of this procedure. It is nearly impossible to analyze malware on 0 or 1s when trying to do so. The compiled software is changed by disassembler software into readable and understandable assembly language. Hex Rays' IDA Disassembler software is utilized in this study due to its versatility, usability, and support for a wide range of file formats

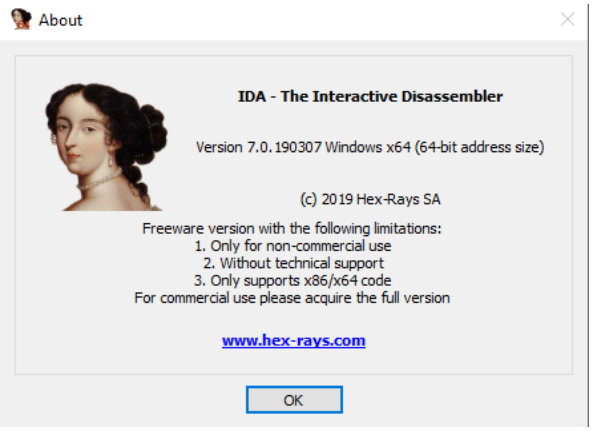

Figure 9 IDA Disassembler software of Hex Rays

Debuggers

Debuggers are programs that let us watch and change a program's operation step by step as well as watch and control the program's registers and stack while it is running.

x64dbg

An open-source Windows binary debugger called X64dbg is designed for malware analysis and executable reverse engineering. There are numerous features available, and a robust plugin system is included. Due to its simplicity of use and intuitive GUI interface, X64dbg has grown to be one of the most widely used malware debugging programs.

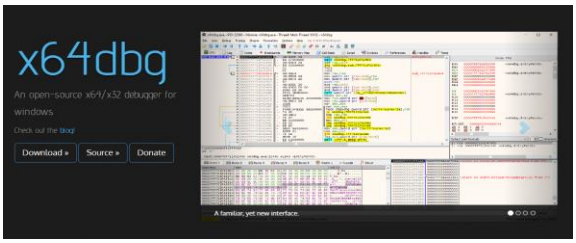

Figure 10 x64dbg Homepage

File viewers, editors and identification tools

Portable Executable File Format (PE) file editors display the data in the files in a readable format. A malware analyst might find information in Portable Executable File Format useful. If you examine the "Machine" information in the Image File Header, for instance, you can determine whether the malware was designed to target 32-bit or 64-bit operating systems.

1. CFF Explorer
2. PEView
3. PEiD
4. BinText (not a File Editor but it can show you strings inside PE File)
5. DocFileViewerEX

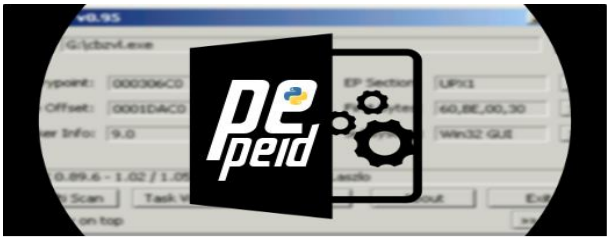

Figure 11 PEID Tool

Network analysis tools

Malware engages in network activities for a variety of purposes, including data theft, command reception from command control servers, and network propagation. The malware analyst needs to have a tool in their toolbox that can analyze network activities in order to

monitor and track the malicious software's network activities. Here are some tools for network analysis.

1. Wireshark
2. Fiddler

Figure 12 Wireshark Packet Analysis Tool

Other Dynamic tools

Please be aware that dynamic malware analysis can jeopardize the security of your network and system because you will be running actual malware to examine its operation. We advise you to only run malware on dedicated systems or virtual machines in remote networks disconnected from the internet. Our malware analysis machine does not require an internet connection because there are numerous tools available for simulating an internet connection. In this article, we'll discuss a few of these tools. It is not guaranteed that the host or your network is completely safe even though we are running the malware in virtual machines because malware developers constantly come up with surprising new ways to infect systems and make malware analysis more challenging.

Figure 13 Windows Sysinternals

As previously mentioned, I'll be examining Process Hacker, Wireshark, and Regshot for dynamic malware analysis. We will provide a download link for the tools for your convenience. Along the way, we will be updating this list, so be sure to sign up for our newsletter.

4. Malware analysis approach

Static Approach

Malware Header Analysis

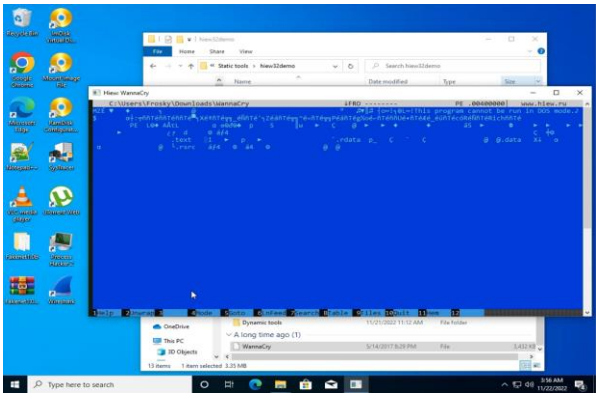

Figure 14 Malware Header analysis using Hiew

The Malware header shows that it's a 'PE' extension, PE stands for Portable executable file. And further we can find '.text', '.rdata', '.data', '.rsrc' strings present in the header. A screenshot of the various sections for the binary has been included under the Packing section. The '.text' section primarily contains executable code, '.rdata' section has globally accessible read-only data and '.data' section has globally accessible data.

Malware Developer info

The malware was developed using Visual C++ language version 6.00. The binary appears to not be packed due to a few indicators. Since it shows the language we can assume that the malware isn't packed. Also if the strings are readable and if the tool doesn't show a name of the packer here it is not packed.

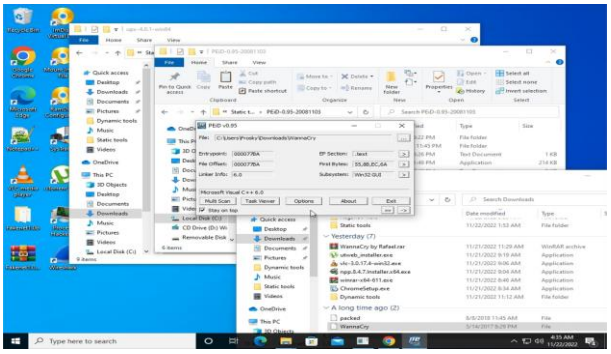

Figure 15 Malware Development info using PEiD

Unpacking packed malware

I have downloaded a packed malware to continue my observation. The PEiD tool showed that it's packed using upx-4.0.1. To unpack a packed malware we need the same packer, so I had download the packer and unpack the malware.

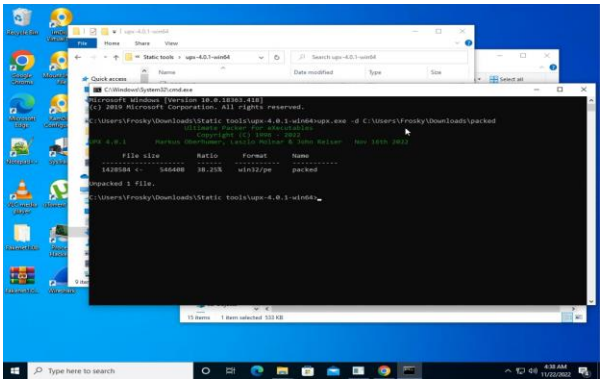

Figure 16 Unpacking packed malware using upx-4.0.1

String analysis

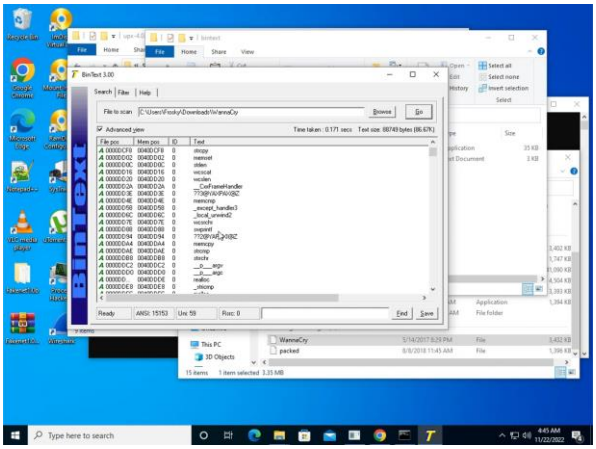

Figure 17 Malware string analysis using bintext

As per analysis I managed to find the strings inside the malware, there were some suspicious functions and strings are found on the malware file.

CFF File Analyzer

CFF file analyzer given some important piece of information about the malware. The malware modified date is also included. The hashes are also present and the hashes were found malicious according to virustotal. Some important dll files were found suspicious from the analysis.

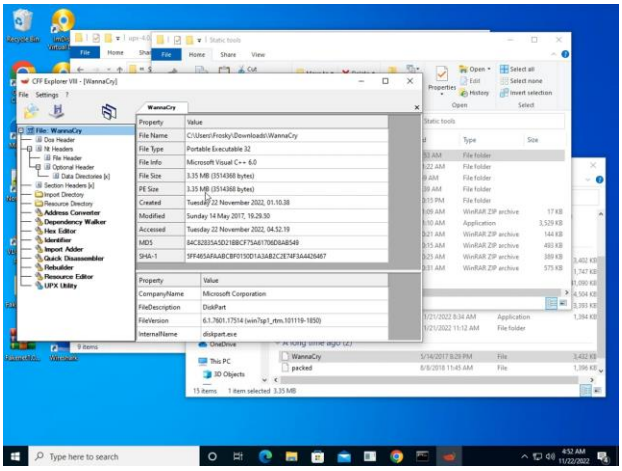

Figure 18 File analysis using CFF Explorer

Malware Reverse engineering

Assembly code analysis of the malware was done by IDA disassembler tool. This analysis has shown that some cryptographic functions are being executed throughout the execution process.

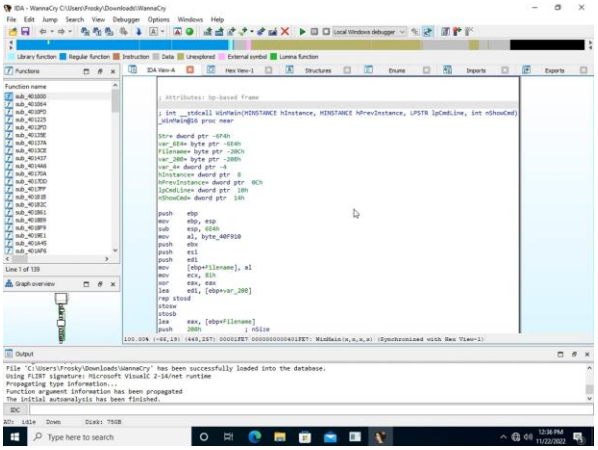

Figure 19 Disassembling malware using reverse engineering

Dynamic Approach

Debugging Malware

While debugging using 'X64dbg' it has confirmed that some functions are being called such as encrypting and generating cryptographic keys. Therefore we can confirm that it's a ransomware.

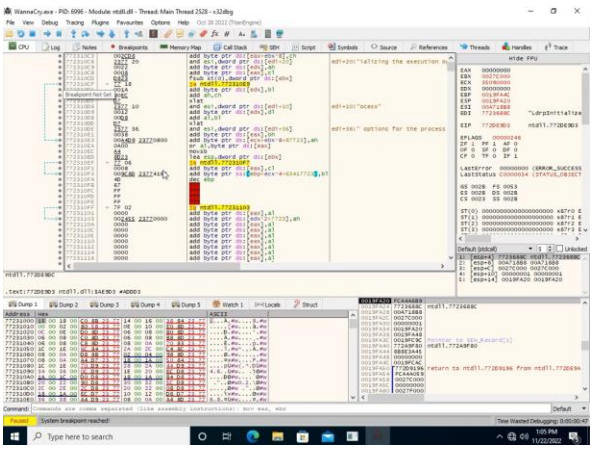

Figure 20 Debugging malware using 'X64 dbg'

Process Monitoring

After the execution of the malware file some process have been created. By observing those processes it has confirmed that some registry events are being executed inside those processes. Some key changes are present on the below figure.

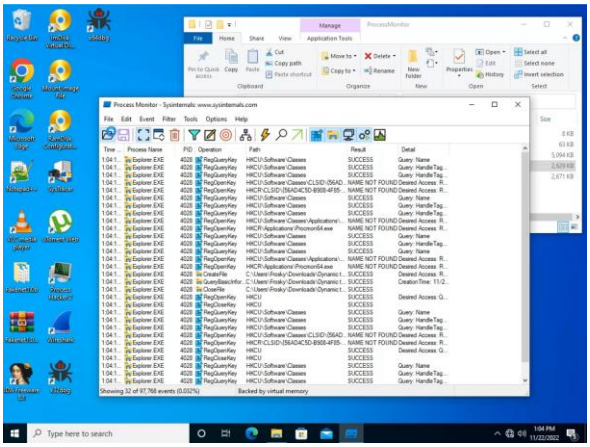

Figure 21 Process Monitoring Using Procmon

Source code Browsing and analysis

Ghidra can analyze the code and behavior by scanning the the malware file. It also organizes all the contents and headers in a structural way that can be easy when analyzing the malware. All the functions were arranged in a 'tree' form that we can observe. Imports and namespaces are also can be observed separately.

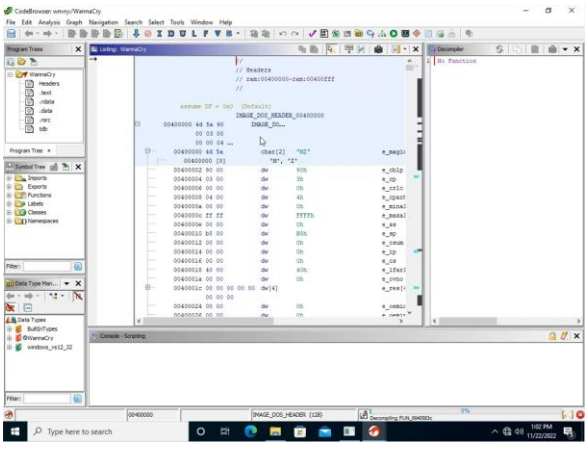

Figure 22 Code Browsing using Ghidra

Registry analysis

Some registry keys that were added or modified are shown below.

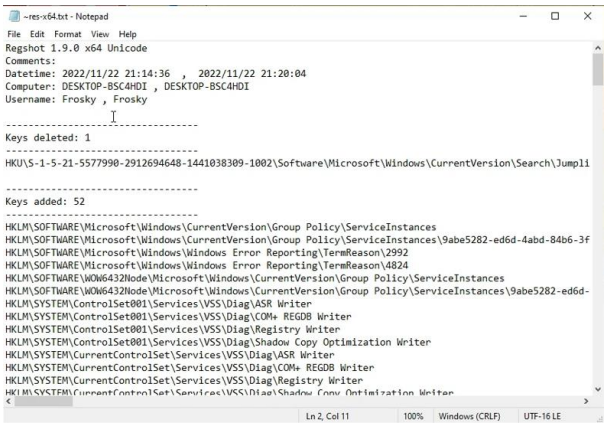

Figure 23 Regshot Comparison

The above information was retrieved using regshot by snapshotting before and after running the malware. The key changes are recorded as shown.

Process Analysis

When the malware is executed I have analyzed the processes created by the malware and its information and events.

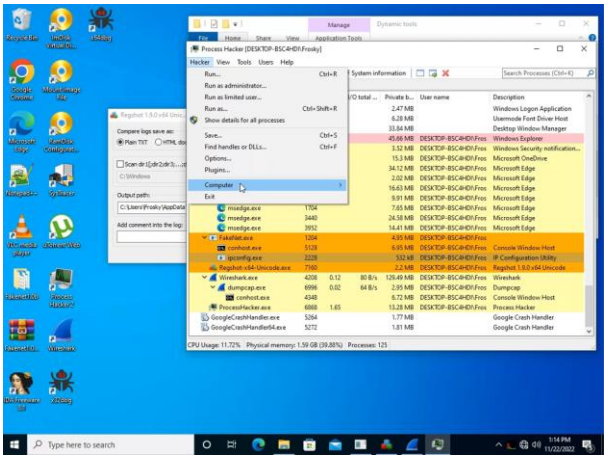

Figure 24 Analyzing process using Process Hacker

To avoid system disruption, the ransomware does not target any executable files (.exe and.dll). When the encryption process is complete, the ransomware will display a window (Wana Decrypt0r program) with decryption instructions.

Figure 25 Wana Decryptor 2.0 window

The ransomware in this Wanna Decrypt0r program demands that the victim pay $300 in Bitcoin to that specific address in order to decrypt the files.

Network Analysis

The wireshark has confirmed that the malware is trying to connect with its host by attempting to send packets.

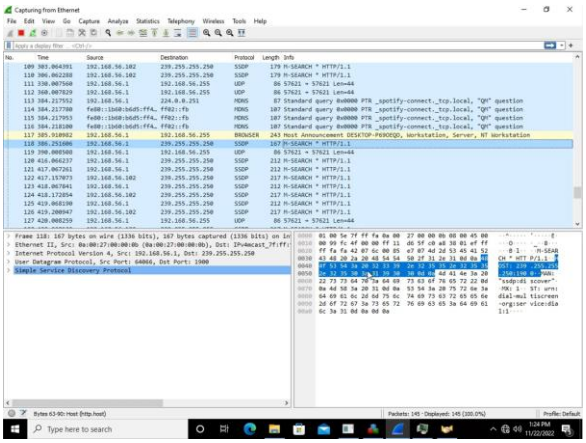

Figure 26 Network traffic analysis using Wireshark

5. CONCLUSION

Because it combines worm and ransomware features, WannaCry is a potent piece of malware [1]. The massive spread of WannaCry was made possible by its worm module. It can spread throughout the local and external networks of a system that is infected. By testing the malware in a controlled environment, we have examined the malware's propagation mechanism in this paper [2]. Additionally, the WannaCry ransomware module encrypts the victim's data and demands payment to decrypt it. In our analysis, we looked at the ransomware's cryptographic model, which we found to be well-designed and prevents the recovery of the decryption key [2]. This is because it combines the convenience of asymmetric key cryptography with the speed of symmetric key cryptography. Furthermore, we investigated the motive behind the WannaCry attack, which, contrary to previous assumptions, was not merely financial extortion but rather a state-sponsored cyberattack. Several things can be learned from this attack: To lessen the effects of such attacks, the system must be updated frequently and backup files must be kept on an external medium.

6. REFERENCES

[1] “WannaCry ransomware attack,” Wikipedia. Accessed September 17, 2018. [Online].

[2] LogRhythm Labs, “A Technical Analysis of WannaCry Ransomware” 2017. [Online].

[3] D. Brien, “Internet Security Threat Report Ransomware,” Symantec, 2017. [Online].

[4] WannaCry: Malware trends tracker (2022) WannaCry | Malware Trends Tracker. Available at: https://any.run/malware-trends/wannacry (Accessed: November 24, 2022).

[5] Mandiant WannaCry malware profile, Mandiant. Available at: https://www.mandiant.com/resources/blog/wannacry-malware-profile (Accessed: November 24, 2022).

[6] Counter Threat Research Team, “WCry Ransomware Analysis,” Secureworks, 2017. [Online]