

Dynamic Analysis of a Window-Based Malware Using Automated sandboxing

Dr. Chandrashekhar UPPIN

HOD, Department of Computer Science, Faculty of Computing and Applied Sciences
Baze University Abuja, Nigeria.

Abstract: Malwares are one of the most dangerous security threats in today's world of fast growing technology. Now, it is not impossible to remotely lock down a system's files for ransoms even when it is located overseas. This threat was accelerated when the world was introduced to cryptocurrency (for e.g., Bitcoins). It allowed the attackers to hide their tracks more efficiently. From a simple idea of testing the efficiency of a computer system to the most critical and sophisticated cyber-attack, malwares has evolved over the years and appeared time to time. Even with the smartest technologies today where we are trying to include Machine learning and Deep learning to every field of our life, the attackers are already developing more sophisticated malwares using the same Machine learning and Deep learning techniques. This raises the question on the security of the cyber-world and how we are able to protect it. In this work, we are presenting an analysis on a recent and most critical Windows malware called "LockerGoga". Both static and dynamic analyses are performed on the malware to understand the behavior and characteristics of the malware.

Keywords: Malwares, Machine learning, Deep learning, LockerGoga, Cryptocurrency, Static and dynamic analyses.

Date of Submission: 14-06-2019

Date of acceptance: 29-06-2019

I. Introduction

Today we are facing an Armageddon of cyber-battle that is constantly threatening the very existence of the Cyber-world. We already saw in last few years that how much capable the malwares are. Taking over a system remotely or breaking down the whole internet is not a big deal now. They are now strong enough to take full control of infected host or network connection bypassing the security features installed [1]. Malwares typically can steal information from a host computer or network, take remote control of a system or can even increase the CPU usage of a system in such a rate that the system can crash [2]. Every day, critical malwares which are more advanced than the previous one are being reported world-wide. The attackers are learning from the loopholes of their past malwares and implementing new advanced technologies to overcome it [3]. Malware analysis is the first step towards finding an effective way to limit the effects of the malwares. To identify the characteristics and behavior of a malware, it is very important to know the structure of the malware, which can be done through a thorough analysis of the malware in an isolated environment. But even before starting analyzing a malware, we need to have basic ideas like what a malware really is and how it affects other systems [4].

A malware can be defined as a sophisticated and accurately designed sequence of malicious code that can be executed remotely or can run automatically when the necessary environment conditions are met, to initiate and carry out its pre-defined list of malicious activities on a host or network connection. There are different types of malwares that are raising havoc to the cyber-world. For example, virus, spyware, adware, rootkits, Trojan horse, worm, ransomware, Keylogger etc. [5]. They all work differently to achieve different goals, for e.g., Trojan horse can act as a normal program initially but when conditions are met, it can make serious damage where ransoms lock down users' systems and asks for ransoms for providing the decryption key [6].

Some malwares are the extended advanced version of their predecessors. In this work, we are analysing LockerGoga, a Windows malware which recently struck, is actually the successor of the Odin ransomware.

About LockerGoga:

LockerGoga is a malicious program categorized as ransomware which first appeared in January 24, 2019 at Romania. Cyber criminals who designed this computer infection use it to encrypt data stored on computers and blackmail users by demanding ransom payments in return for decryption tools. LockerGoga adds the ".locked!?" or ".locked" extension to each encrypted file. It uses RSA-2048 algorithm.

II. Literature Review

We have done the literature review on the previous works to find out the problem statement:

Table I: Literature Review

| Sl. No. | Name of papers | Authors | Date of publication | Objective |
|---------|---|---|---------------------------|--|
| 1 | Malware Architectural View with Performance Analysis in Network at Its Activation State | Sisira Kumar Kapat, Satya Narayan Tripathy | February, 2019 | Focuses to analyze malware architecture to give a detailed study of malware. It classifies malwares into four categories as per their architecture at the time of infection also observes the performance of network at the time of infection. |
| 2 | A Survey on malware analysis and mitigation techniques | S. SibiChakkaravarthy, D. Sangeetha, V. Vaidehi | January, 2019 | This paper presents a detailed study on sophisticated attack and evasion techniques used by the contemporary malwares. |
| 3 | Efficient dynamic malware analysis using virtual time control mechanics | Chih-Hung Lin, Hsing-KuoPao, Jian-Wei Liao | December, 2018 | It proposes a virtual time control mechanics based method which utilizes a modified Xen hypervisor, in which a virtual clock source is generated to accelerate the sandbox running. |
| 4 | Malware Analysis and Mitigation in Information Preservation | Aru OkerekeEze, ChiaghanaChukwunonso E | August, 2018 | Analyses the Behavior-Based Detection methods and mechanism to build behavior-based malware detection and classification methods. |
| 5 | A survey of malware behavior description and analysis | Bo Yu, Ying Fang, Qiang Yang, Yong Tang, Liu Liu | 8 th May, 2018 | Presents a survey on malware behavior description and analysis considering the aspects of malware behavior description, behavior analysis methods, and visualization techniques. |
| 6 | Improving the effectiveness and efficiency of dynamic malware analysis using machine learning | Leonardo De La Rosa | April, 2018 | Introduces a next-generation sandbox that uses machine learning to create an adaptive malware analysis platform. |
| 7 | RARE: A Systematic Augmented Router Emulation for Malware Analysis | Ahmad Darki, Chun-Yu Chuang, Michalis Faloutsos, Zhiyun Qian, and Heng Yin | March, 2018 | Proposes RARE which is a systematic approach to analyze router malware and record its behavior focusing on home-office routers. |
| 8 | A Survey On Automated Dynamic Malware Analysis Evasion and Counter-Evasion | Alexei Bulazel, BülentYener | November, 2017 | Reviews fingerprint-based evasion techniques, evasion detection, evasion mitigation, offensive and defensive evasion case studies. |
| 9 | Empowering Convolutional Networks for Malware Classification and Analysis | BojanKolosnjaji, GhadirEraisha, George Webster, ApostolisZarras, Claudia Eckert | May, 2017 | Analyses the performance improvements achieved in the area of neural networks to model the execution sequences of disassembled malicious binaries. |

III. History Of Malware Attacks In Windows

The concept of malwares was introduced for simply testing the limitations of a computer system. At first, they were not intended to harm any system [1]. The idea was proposed by John Von Neuman in 1949. He proposed a system for self-reproducing automata that can test the efficiency of a system. In 1971, Robert H. Thomas developed Creeper Worm, the first malware that can replicate itself and spread through systems which ran on TENEX operating system. When spread to a system, it shows the message: “I’m the Creeper: Catch me if you can”.

In 1974, the Rabbit virus was created by an unknown user which was followed by the Animal virus, which was another extension of the Rabbit virus [2-3].

In 1982, Elk Cloner made its appearance which opened the doorway to modern sophisticated viruses and malwares. Since then, the rise of the malwares started to accelerate in a quick pace. Many of the history’s most dangerous worms, viruses, Trojans were developed in that time period.

Then came the age of ransomwares. In 1989, the AIDS ransomware (aka, PS Cyborg) appeared which was the first ransomware attack, targeted on WHO’s international AIDS conference. This was followed by the Archiveus Trojan in 2006, which encrypted everything in My Documents section of a computer [5].

In 2008, when Bitcoins were introduced, this accelerated the pace of the ransomware attacks. The attackers now had the facility to demand ransoms in crypto-currencies which allowed them to hide their tracks.

This next development in the history of ransomware meant that attackers no longer needed to encrypt the hijacked files, instead a fake Windows Product Activation screen, forcing users to call a number in search of an activation code at an international premium rate.

In 2012 a new Trojan called Reveton was developed, this had perhaps the most widespread impact so far as it spread across Europe.

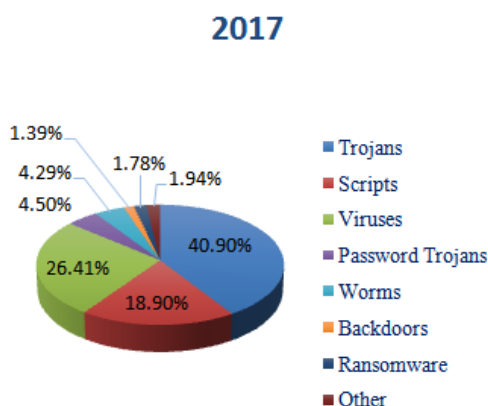


Fig. 1: Malware statistics (2017) by AV-Test

In 2013, the world faced the rise of CryptoLocker, an attack that would give those infected a strict 72 hours to pay \$400 in Bitcoin or else their encrypted files would be erased without mercy.

The TeslaCryptransomwareappearedin 2016, whichtargeted files associated with video games — saved games, maps, downloadable contents.

When the world was recovering from previous attacks of ransomwares, in 2017, yet another new level of damage was caused by a new ransomware called “WannaCry”. This ransomware caused a high range of havoc in the whole world.

Short after WannaCry, the Petya and NotPetya ransomware attacks again raised havoc to the whole world which swept through hospitals, banks and governments in several countries.

Attackers, who are willing to spread their malware attacks, always tend to implement it on a large-scale basis. To do so, they always implant their malwares on the most vulnerable software eco-systems that are most widely in use. Windows operating system, as we know, is the most widely used operating system, which also happens to be the most vulnerable. This consistently made the operating system to be on the target list of the attackers over the past years. In 2017, over 67 percent of all malware attacks were aimed at Windows systems. In 2018, the percentage increased to 73.80 percent.

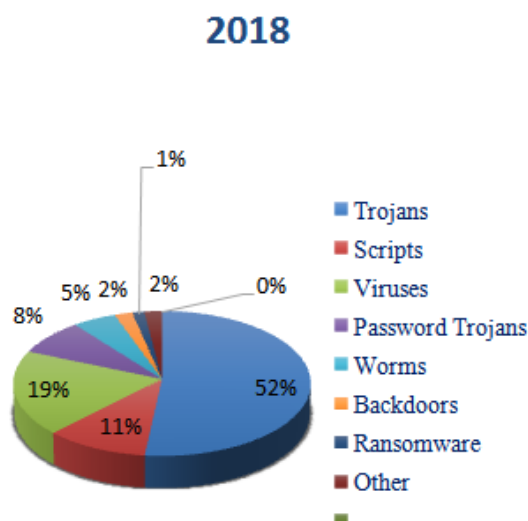


Fig. 2: Malware statistics (2018) by AV-Test

IV. Problem Statement

Many technologies have been developed over the years to cope with the increasing threat of the cyber-security but the ration of cyber-crime is increasing instead of decreasing. Heuristic-based tools use rules to examine suspicious codes and classify them as malware. This approach is limited, however, due to the fact that it relies on the sequence of repeated code that is indicative of malicious intent. Hence, in this work, we are presenting a view on the combined approach of static and dynamic analyses with tools based on real-time extraction.

V. Environmental Setup

To ensure a secure analysis of the LockerGoga malware, the environmental setup should be proper. We have created a laboratory setup for this purpose. We are using Oracle Virtual Box to create a virtual environment for the malware. We need to use FakeNet tool to prevent it from recognizing that it is running inside a virtual environment as LockerGoga has an inbuilt anti-VM trick installed.

We installed the virtual box on the host machine which is running Ubuntu 16.04 LTS. Within the virtual box, we installed Windows 7 Ultimate operating system to analyze the malware.

We have setup Cuckoo sandbox in Ubuntu for automated analysis of the malware. The configurations of the host and guest machines are as below:

TABLE II: HOST MACHINE CONFIGURATION

| Host Machine | |
|---------------------|------------------|
| Model | Lenevo Z360 |
| Processor | Intel I3 |
| RAM | 8 GB |
| Operating System | Ubuntu 16.04 LTS |
| System Type | 64-bit OS |

TABLEIII: GUEST MACHINE CONFIGURATION

| Guest Machine | |
|----------------------|--------------------|
| Operating System | Windows 7 Ultimate |
| System Type | 64-bit OS |
| Internal storage | 50GB |
| RAM | 2 GB |

VI. Tools For Malware Analysis

As we are experiencing more highly advanced malware attacks everyday around the world,it has become necessary to advance the level of malware analysis process. As we know, Windows operating system is most vulnerable to the malware attacks, it is highly recommended to use extra caution while handling malware samples in Windows environment.

The following are the tools required to analyze Windows malware for static and dynamic analysis which is followed by Linux and android tools to analyze malwares:

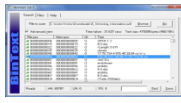
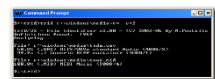



Cuckoo sandbox:

A sandbox is a type of software testing environment that enables the isolated execution of software or programs for independent evaluation, monitoring or testing. In an implementation, a sandbox also may be known as a test server, development server or workingdirectory.

Cuckoo sandbox is a leading open-source automated malware analysis environment that is maintained by “Elite group” which virtually analyses the malware in an isolated environment and provides a detailed report about the behavior of the malware through both static and dynamic analysis.


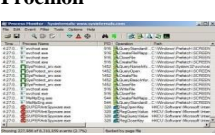




A. Windows Malware Static Analysis Tools:

TABLE IV: STATIC ANALYSIS TOOLS

| | | |
|---|--|---|
| 1 | BinText  | It is a small but very powerful text extracting tool which uses Binary-to-text extracting mechanism and provides output in simple plain text which can include ASCII text, Unicode (double byte ANSI) and resource strings. It can extract text from any format of file. |
| 2 | TrID  | It is used to identify file types from their binary-signatures and has no fixed rules for the same. Instead, we can train it for learning new file identification types. |
| 3 | UPX  | UPX (Ultimate Packer for Executables) is a freeware and open source executable packer for compression which uses UCL data compression algorithm. |
| 4 | XORSearch  | It is an open source executable tool which searches for XOR (0-255), ROL (0-7), ROT (1-25) or SHIFT encoded strings in a file using brute-force mechanism. It displays all the critical information about outbound communication of the malware like IP address, URL etc. |
| 5 | Exeinfo PE  | Exeinfo PE is a program that verifies .exe files and analyses their properties. It provides the exact size of the file and the point of entry of the malware. It also provides information about the packing, language used to create the malware etc. |

B. Windows Malware Dynamic Analysis Tools

TABLE V: DYNAMIC ANALYSIS TOOLS

| | | |
|---|--|--|
| 1 | FakeNet  | It simulates a fake network so that the malware interacting with a remote host continues to run without learning that it is running within a virtual environment, allowing the analyst to observe the malware's network activity. |
| 2 | Procmon  | Procmon (Process Monitor) is a freeware from Windows Sysinternals, which monitors and displays in real-time all file system activities. It combines two older tools, FileMon and RegMon to monitor the process activities of the malware. |
| 3 | ProcDOT  | It processes procmon's logfiles and PCAP-logs (Windump, Tcpdump) to generate a graph via the GraphViz suite. This graph visualizes activities that are related to it. |
| 4 | Wireshark  | It is used to analyze the structure of different network protocols and has the ability to demonstrate encapsulation. It is also useful to capture network traffic to analyze the incoming and outgoing logs. |
| 5 | Process Explorer  | Process Explorer (PE) is a freeware task manager and system monitor. It provides the functionality of Windows Task Manager along with a rich set of features for collecting information about processes running on the user's system. It can be used as the first step in debugging software or system problems. |
| 6 | RegShot  | RegShot is an open-source registry compare utility that allows us to quickly take a snapshot of our system's registry and then compare it with a second one - after executing the malware. It provides information about the changes and effects on the registry. |

VII. Our Approach

While performing a malware analysis of any operating system, it is crucial to take necessary precautions and to have a backup of the system. Setting up the proper environment for analyzing a malware is the most important thing. The following are the steps that were followed for performing analysis of LockerGoga malware.

STEP 1: The first thing we need to do is to setup a Virtual environment for performing the operations. For the same, we are using Oracle virtual box, which we installed in Linux 16.04 LTS operating system. We then installed Windows 7 Ultimate within the Virtual box to analyze the malware.

STEP 2: Now we need to download the appropriate tools for both static and dynamic analysis of the malware (as mentioned in table 1 and table 2). It is important to remember that we always need to perform static analysis before moving on to dynamic analysis. The tools are freely available on internet.

STEP 3: Once all the tools are downloaded and installed, we need to take a snapshot of the whole environment for backup which will be needed after we perform dynamic analysis of the malware to restore back to the original state of the environment.

STEP 4: Now, using the static analysis tools we need to perform the analysis one by one. We have downloaded the sample of LockerGoga and extracted it. We need to simply drag and drop the bin (binary) file to the tools except TrID and XORsearch. These two tools are command based and the following commands should be used:

TrID:

```
trid <file location with file name>
```

XORSearch:

```
xorsearch.exe <file name> -http
```

Following are some of the screenshots taken of the static analysis of LockerGoga:

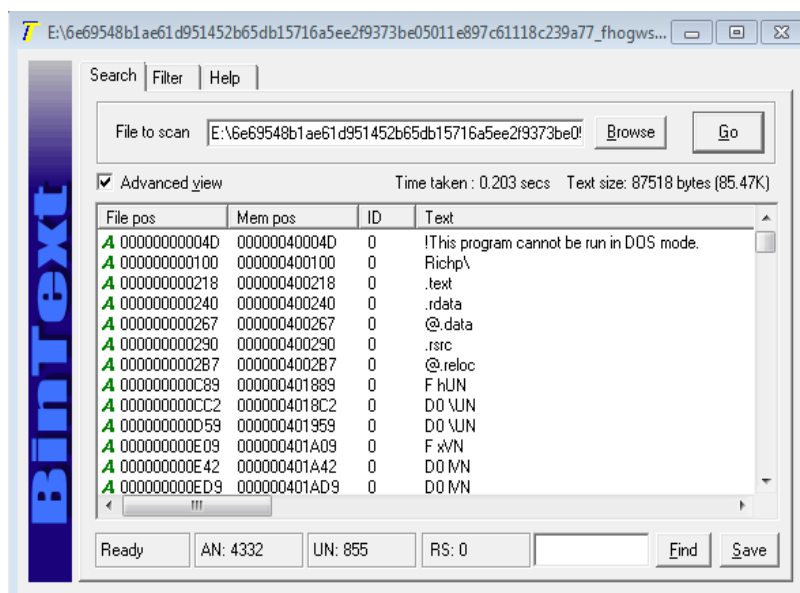


Fig. 4: BinText results

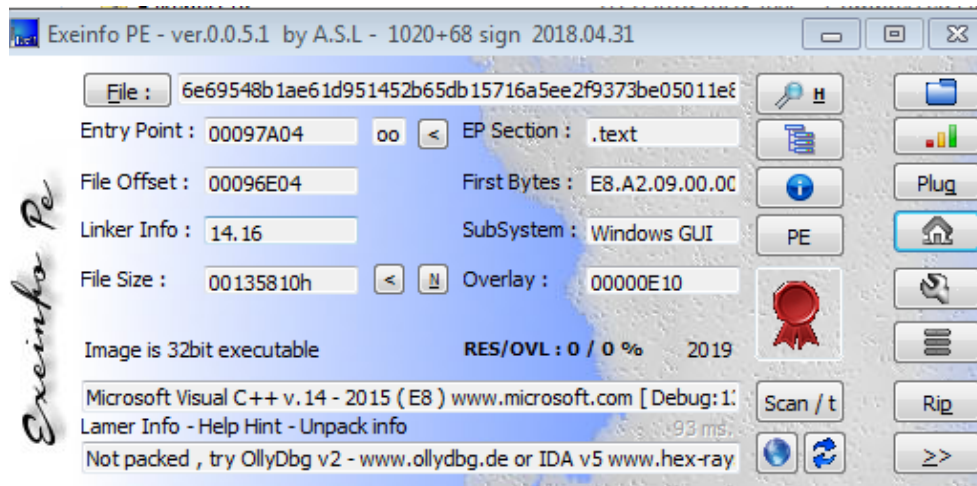


Fig. 5: Exeinfo PE results

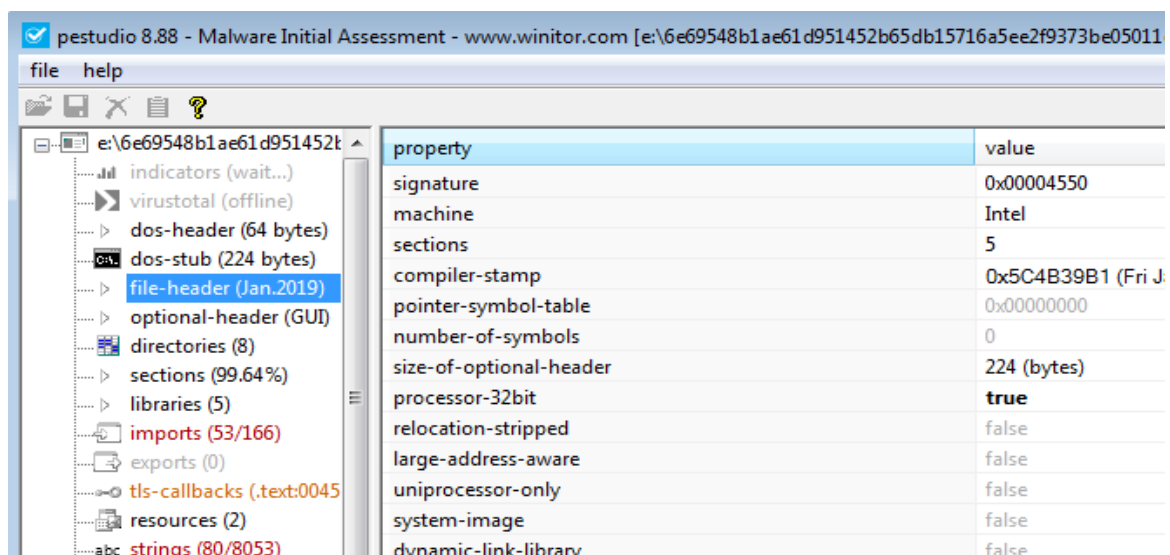


Fig. 6: PESTudio results

STEP 5: Once the static analysis is done, now we need to move to the dynamic analysis of the malware. The dynamic analysis is performed by running the malware in a secured environment (it's important to keep the snapshot of the Virtual environment, a mentioned in STEP 3). We need to start the dynamic analysis tools in the following order:

FakeNet → Procmon → ProcDOT → Process Explorer → RegShot (1st shot) → Execute the malware (LockerGoga, in this case) by extracting the bin file and renaming it with .exe extension → RegShot (2nd shot) → Compare both registry shots → WireShark.

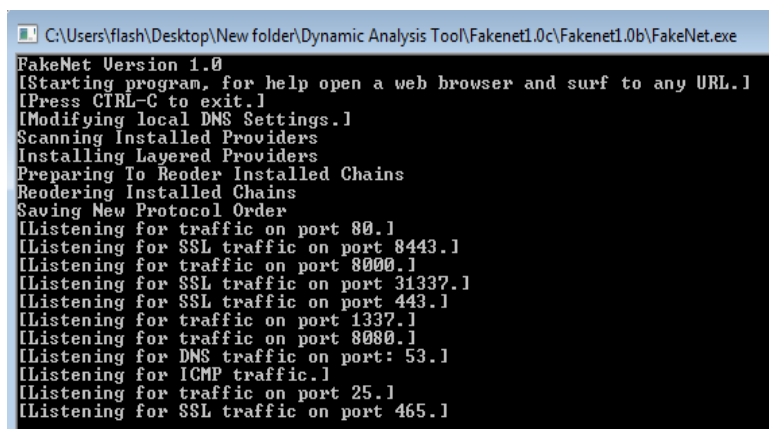


Fig. 7: FakeNet results

Once the dynamic analysis is done, it is important to remember to restore snapshot taken (STEP 3) to clean the environment state.

The following are some of the screenshots taken while dynamic analysis:

The figure 7 shows that the FakeNet server was started and it is using ports 25 and 465 for listening to requests.

The figure 8 shows the results found by Procmon (Process Monitor) after running the malware in virtual environment. As we can see, the malware is using svchOst.exe for spawning several processes (e.g., svchOst.25671 used for locking the host files) to perform different operations.

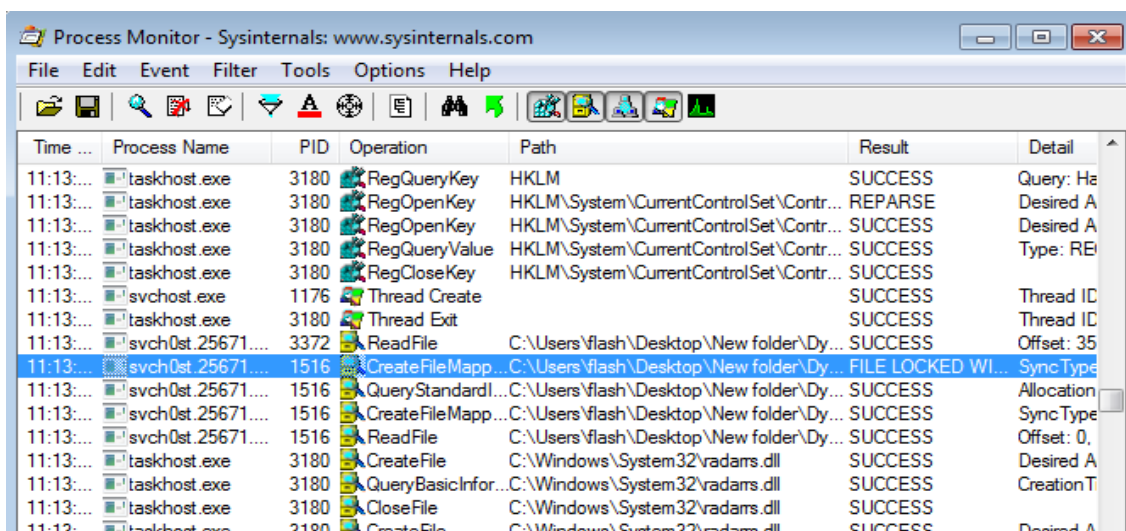


Fig. 8: Procmon results

VIII. Results

The following results were found by performing the analysis of the LockerGoga malware:

A. Sections:

It was found that the malware is packed. Packed or repacked malware is malware that has been modified using a runtime compression (or encryption) program. If the entropy of the .text field is greater than 5.0, it is assumed to be a packed malware. Here the .text file has an entropy value of 6.6251837006; hence it is a packed malware.

B. DLL files:

DLL (Dynamic Link Libraries) files are system files that contain instructions that other programs can use when needed. A single file can be used by different applications. The DLL files (.dll extension) that were found is shown in Table 3.

TABLE VI: DLL FILES USED BY THE MALWARE

| DLL files | Description |
|--------------|--|
| ADVAPI32.dll | It is an advanced API service library that contains machine codes for the system to work properly. |
| KERNEL32.dll | It is a 32-bit DLL file found in Windows Kernel. It handles memory management, input/output operations and interrupts. |
| SHELL32.dll | It serves as a graphical user interface (GUI) for Windows operating system. |
| SHLWAPI.dll | It contains necessary functions for UNC and URL paths, registry entries and color settings. |
| ole32.dll | It contains core OLE (Object Linking and Embedding) functions. |

C. Registry keys:

Following the observation of the LockerGoga malware, it was found that it touches several registry keys to enter into a system’s core operating environment to execute its process of lockdown. Mostly, the HKEY_LOCAL_MACHINE (HKLM) registry section was being targeted to gather information about the system and the user and their security features to bypass them.

D. Functions called:

The malware uses a whole list of critical functions to gain access to restricted areas of the system’s operating environment (e.g., CreateFileA, ReadFile, GetCurrentPackageID, RemoveDirectoryW, WriteConsoleW etc.).

E. CPU usage:

The following graphs are showing the average CPU usage before running LockerGoga (Fig. 6) and after running LockerGoga (Fig. 7).

In fig. 6, we can see that there are only the usual processes running. As observed, it was found that the total CPU usage before running the malware was 0% and after running the malware was 21%, which is because the malware was using several new processes to adjust the CPU usage.

F. Other observations:

Observing the malware closely in our lab setup, it was found that the malware is using Boost library package version 1.68 for Crypto++. It is using FLIRT IDA (Fast Library Identification and Recognition Technology – Interactive Disassembler) for spawning processes which allows the IDA to recognize standard function calls, and enhance the output. It is IDA’s internal symbols identifier that searches through disassembled binaries in order to locate, rename, and highlight known library subroutines.

It is also using the svchOst.exe, which is a system process of Windows, to spawn new processes.

FLIRT eliminates the need to analyze functions that could be understood simply by reading the internal documentation or source code from the library it came from.

It reduces the amount of work required in order to reverse and understand symbol-stripped binaries by a considerable amount.

The malware is also using BufferedTransformation function call to use pipelining for transferring user information to the end points.

By analyzing we found that it is targeting SSE2 (Streaming SIMD Extensions 2) version of processors which is mainly produced by Intel.

We found that the following email address is used by the malware from the signer named “MIKL LIMITED”.

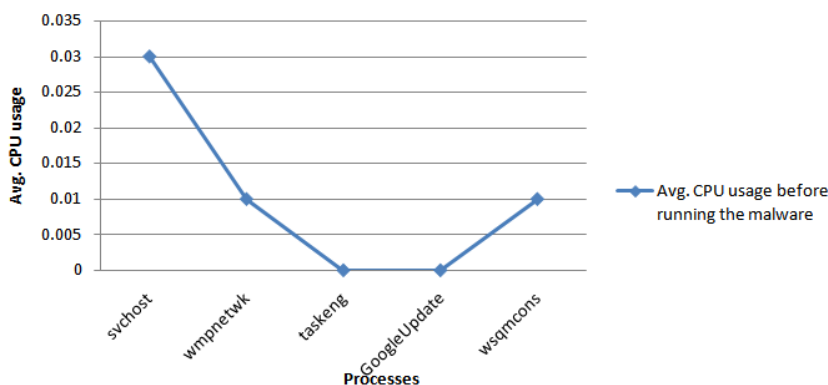


Fig. 9: Before running LockerGoga

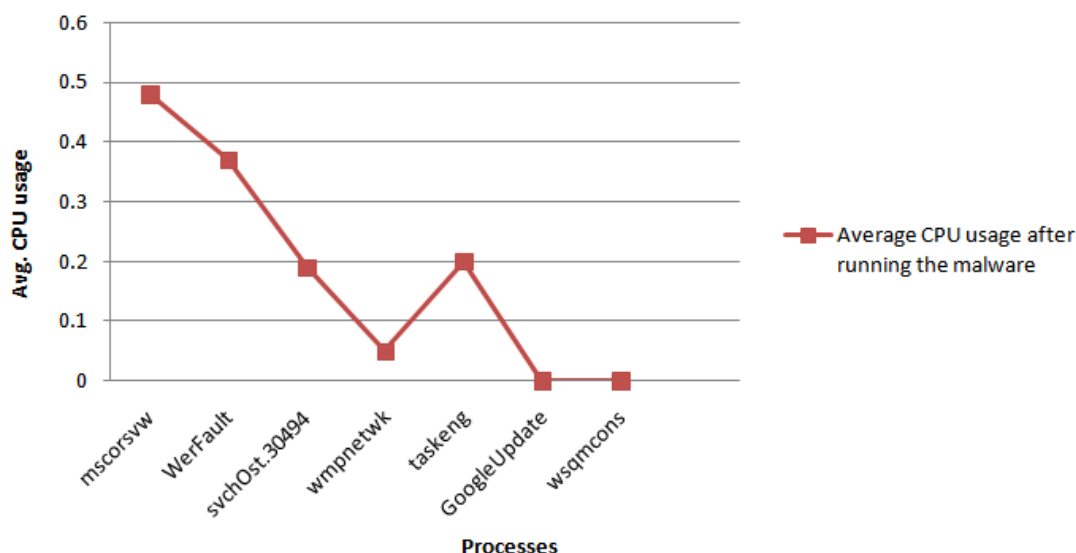


Fig. 10: After running LockerGoga

IX. Some Recommendations To Mitigate Malware Attacks

- We can use firewall for blocking all incoming connections from the Internet to services that should not be publicly available.
- To prevent computer infection by ransom ware-type (or other) high-risk infections, browse the web, install, download and update software with care.
- Do not open attachments that are included in emails received from unknown/untrustworthy or suspicious addresses.
- Update installed software (or the operating system itself), using implemented functions or tools provided by official developers.
- Disable AutoPlay to prevent the automatic launching of executable files on network and removable drives and disconnect the drives when not required. If write access is not required, enable read-only mode if the option is available.
- Avoid downloading software using untrustworthy, unofficial websites, third party downloads or other tools of this kind. Third party download or installation set-ups can include rogue applications that might cause computer infections (or other problems).

VI. Conclusion and Future Scope

Day to day malware is being spread via network like wildfire. However, preserving information and records in a system involves ensuring they remain accessible, usable and free from malware attacks. Information and records will deteriorate over time, whether they're paper, photographic, digital or audiovisual if they cannot be preserved from possible malware attacks. While the rate of deterioration differs, the lifespan of your information and records will depend on how they are managed and the preservation actions applied throughout their lifecycle. In this work, we are presenting the steps to analyze a malware in a secured environment with the example of the LockerGoga ransomware.

Although, the ratio of malwares is increasing at an alarming rate, this work provides a thorough study of tools for analyzing malwares. In addition, malware mitigation strategies are also listed.

It is likely that the methods presented in this work would have a significant impact in helping cyber-cleaning. The study highlighted the steps required for effective and good malware mitigation strategies; there is a need for follow-up research using the tools and different methods to help organization understand what is required to improve the effectiveness of their information preservation policy against malwares.

References

- [1]. Chakkaravarthy, S. S., Sangeetha, D., &Vaidehi, V. (2019). A Survey on malware analysis and mitigation techniques. *Computer Science Review*, 32, 1-23.
- [2]. Kapat, S. K., &Tripathy, S. N. (2019). Malware Architectural View with Performance Analysis in Network at Its Activation State. In *Cognitive Informatics and Soft Computing* (pp. 207-216). Springer, Singapore.
- [3]. Lin, C. H., Pao, H. K., & Liao, J. W. (2018). Efficient dynamic malware analysis using virtual time control mechanics. *Computers & Security*, 73, 359-373.
- [4]. Darki, A., Chuang, C. Y., Faloutsos, M., Qian, Z., & Yin, H. (2018, March). RARE: A Systematic Augmented Router Emulation for Malware Analysis. In *International Conference on Passive and Active Network Measurement* (pp. 60-72). Springer, Cham.

- [5]. Yu, B., Fang, Y., Yang, Q., Tang, Y., & Liu, L. (2018). A survey of malware behavior description and analysis. *Frontiers of Information Technology & Electronic Engineering*, 19(5), 583-603.
- [6]. De La Rosa, L. (2018). *Improving the effectiveness and efficiency of dynamic malware analysis using machine learning* (Doctoral dissertation, University of Delaware).
- [7]. Kolosnjaji, B., Eraisha, G., Webster, G., Zarras, A., & Eckert, C. (2017, May). Empowering convolutional networks for malware classification and analysis. In *2017 International Joint Conference on Neural Networks (IJCNN)* (pp. 3838-3845). IEEE.
- [8]. Bulazel, A., & Yener, B. (2017, November). A survey on automated dynamic malware analysis evasion and counter-evasion: Pc, mobile, and web. In *Proceedings of the 1st Reversing and Offensive-oriented Trends Symposium* (p. 2). ACM.

Dr. Chandrashekhar UPPIN. " Dynamic Analysis of a Window-Based Malware Using Automated sandboxing." IOSR Journal of Computer Engineering (IOSR-JCE) 21.3 (2019): 12-.22