iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** X **Month of publication:** October 2021

DOI: https://doi.org/10.22214/ijraset.2021.38416

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue X Oct 2021- Available at www.ijraset.com

A Methodological Study on Malware Analysis

Harshitkumar R. Panwala[1]
[1]*School of Computer Science and Engineering, VIT-AP University, Amravati, Andhra Pradesh, India*

Abstract: Malware is an executable binary that is designed to be malicious. Malware can be used by attackers to carry out a range of malicious operations, such as spying on the victim using keyloggers or remote access tools (RATs) or deleting or encrypting data for "Ransom" payments. Malware is software that is designed to carry out malicious operations, and it comes in a variety of forms. Malware's impact, according to studies, is escalating. There are several tools available for malware analysis. The present study is the analysis of the malware known as "Malware Analysis". Malware analysis is the study or process of extracting as much information as possible from a malware sample in order to determine its operation, origin, and potential impact. The information obtained aids in determining the functioning and scope of malware, as well as how the system got infected and how to guard against future attacks.

I. INTRODUCTION

Cybercriminals utilize malware, or malicious software, to inflict considerable damage on their victims. It can harm the server, the host system, or the network. Attackers, hackers, and nation-states are all examples of cybercriminals. The harm created could interrupt a computer's or a network's regular operations, steal crucial and secret data, and circumvent access rules to obtain access to confidential regions. It has the potential to hurt victims in unimaginable ways. Individuals, organizations, businesses, governments, and even key bodies working to change the world could be among the victims. Every day, roughly 200,000 malware samples are caught, according to a report. This necessitates a robust procedure that can detect harmful content early on and assist in the creation of a process that can either avert the problem or mitigate the damage. Malware can be classified in the following aspects:

1) *Virus:* To perform the malicious function, it is a kind of program, that gets attached to other programs themselves.
2) *Trojan:* It makes duplicates of themselves and steals data. It is a stand-alone malicious program that attempts to infect other computers in a completely automated manner without the assistance of other forces.
3) *Worms:* A worm is a self-replicating malware computer software that accesses computer and network resources without the consent of an authenticated user. It is eating network bandwidth in the network. On the target machine, there is a security flaw.
4) *Spyware:* It is installed without the user's awareness in order for the attacker to be notified of the user's actions.
5) *Rootkit:* Rootkit is a type of malware that installs a backdoor into a computer system, modifies log files, and deletes data files.

Malware analysis is the act of detecting and reducing any potential threat posed by a virus in order to improve the security of a program, website, or server. Malware analysis is a critical procedure that any firm must go through nowadays to ensure that their data is safe and secure and that they are protected from any vulnerabilities. Malware analysis is the process of determining how a suspicious file or URL behaves and what its aim is. The procedure's output aids in the detection and mitigation of any potential threat. Malware analysis has many advantages for incident responders and security experts. A few of them are mentioned below.

A. Determines the extent of a security threat's impact.
B. Determine the source of the assault, as well as the malware's vulnerability, exploitation degree, and patching readiness.
C. Break occurrences down into categories based on the level of security threat they pose.
D. Identify hidden signs of compromise that need to be addressed.
E. When looking for a threat, any context can be enhanced.

II. CLASSIFICATION OF MALWARE ANALYSIS

The Malware Analysis can be classified into two categories, i.e., Static Malware Analysis and Dynamic Malware Analysis.

A. Static Malware Analysis

Static analysis is the process of analyzing software without running it. Different representations of a program can be subjected to static analysis techniques. Static analysis tools can also be applied to a program's binary representation. Some information is lost when a program's source code is compiled into a binary executable. The work of deciphering the code is made even more difficult by this loss of information.

Manually inspecting a binary without running it is the most common method of doing so. If the source code is provided, for example, various useful details such as data structures and used functions can be extracted. Once the source code is built into a binary executable, this information is gone, preventing further investigation. Static malware analysis employs a variety of methodologies. File fingerprinting, File format, AV scanning, Packer detection, Disassembly, etc. are examples of the methodologies.

The primary benefit of static malware analysis is that it allows for a thorough examination of a specific binary. That is, it can cover all of a malware sample's possible execution routes. Furthermore, because the source code is not run, static analysis is often safer than dynamic analysis. It is, however, time-consuming, and so necessitates skill. The limitation of the Static Malware Analysis are that malware samples' source code is usually not widely available. As a result, the static analysis approaches for malware analysis that can be used are those that recover information from the malware's binary representation. Consider the fact that most malware attacks use the IA32 instruction set to execute their code. If the binary uses self-modifying code techniques, disassembly of such applications may yield confusing results.

B. Dynamic Malware Analysis

Dynamic malware analysis is the process of executing a malware sample in a controlled environment and watching its actions in order to analyze its destructive behavior. Dynamic malware analysis avoids the limitations of static malware analysis because it is performed during runtime and malware unpacks itself (i.e., unpacking issue). It is thus simple to observe a program's true behavior. The biggest disadvantage is so-called dormant code: That example, unlike static analysis, dynamic analysis typically only examines one execution path, resulting in insufficient code coverage. Furthermore, if the analysis environment is not appropriately isolated or controlled, there is a risk of compromising third-party systems. Furthermore, if malware samples realize that they are being executed in a controlled analysis environment, they may change their behavior or stop operating altogether.

There are mainly two basic approaches for Dynamic Malware Analysis. The first is “Analyzing the difference between defined points”. In this approach, A malware sample is run for a set amount of time, and then the changes done to the system are compared to the starting condition. In this method, a comparison report is used to describe malware behavior. The second is “Observing runtime behavior”. In this kind of approach, A specific program is used to monitor the malicious application's activity while it is running.

III. STAGES OF MALWARE ANALYSIS

Mainly fours stages are involved in Malware Analysis. All four stages are explained in this section.

A. Static Property Analysis

This comprises strings embedded in malware code that can be read rapidly and are needed to construct IOCs. It would not be necessary to run software to see it. This is the first level of investigation that will determine whether further investigation is required. It will assess whether more measures are required.

B. Interactive Behavior Analysis

This is used in a lab to analyse a malware sample. It tries to figure out what the registry, process, network activity, and file system are all about. It does memory forensics to determine how malware makes use of memory. If the virus is deemed to be suspect, a simulation can be built up to test the theory. It takes a long time and necessitates the use of a creative analyst with superior expertise.

C. Fully Automated Analysis

Fully automated malware analysis simply evaluates suspicious files and predicts the consequences if they infiltrate the network. It also generates an easily understandable report that provides security teams with quick replies. It's a fantastic technique to perform large-scale malware analysis.

D. Manual Code Reversing

Analysts decrypt any encrypted data and establish the logic by employing debuggers, disassemblers, specialist tools, and compilers to reverse engineer the code. It's an uncommon talent, and mastering it takes a long time. Several analysts opt to omit this stage, resulting in the loss of a wealth of information about the malware's nature.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue X Oct 2021- Available at www.ijraset.com

IV. METHODOLOGY AND RESULTS

The first step was to install a VM machine along with windows 7 as an operating system because many of the malware are compatible with it. A virtual machine, abbreviated as VM, is similar to any other physical computer, such as a laptop, smartphone, or server. It features a CPU, RAM, and disks for storing your files, as well as the ability to connect to the internet if necessary. VMs are generally conceived of as virtual computers or software-defined computers within physical servers, existing solely as code, while the elements that make up your computer (called hardware) are physical and tangible.

In the next step, after downloading the software and malware, the system was switched to a host-only network to prevent malware from spreading to my primary machine if it turns out to be a worm.

Furthermore, the HashTab tool was used. The HashTab supports numerous hash algorithms, including MD5, SHA1, SHA2, RipeMD, HAVAL, and Whirlpool, and provides OS extensions to generate file hashes. It has a simple drag-and-drop interface that makes comparing two files a breeze. The Hashtab is widely used to verify the integrity of a file downloaded from the internet, in addition to comparing files. The Hashtab is a global community and is currently being translated into around 27 languages.

In the later stage, VirusTotal was used. VirusTotal gathers data from a variety of antivirus products and internet scan engines to look for viruses that the user's antivirus may have missed, as well as to check for false positives. Up to 650 MB of files can be posted to the website or emailed (max. 32MB). Anti-virus software suppliers can obtain copies of files that were flagged by other scans but passed through their engine in order to improve their program and, by extension, VirusTotal's capability. Users can also search the VirusTotal dataset and scan questionable URLs. The Cuckoo sandbox is used by VirusTotal for dynamic malware analysis. PC World named VirusTotal as one of the top 100 products of 2007. It was also used in this case to determine whether the hash was malicious or not. It was flagged as harmful by nearly 65 out of 72 antivirus engines (Figure 1).

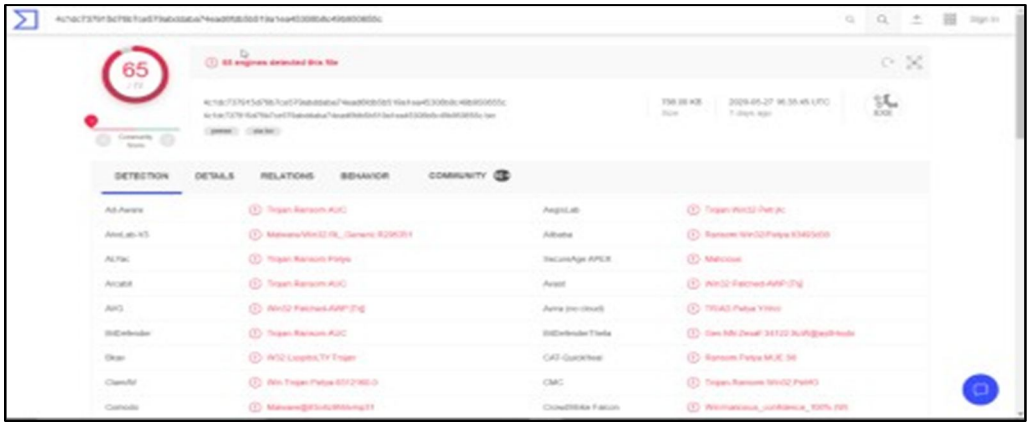


Figure 1: Malicious antivirus engines

By the application of PEiD (Figure 2), it can be determined that the executable's subsystem as well as the executable's entry point. It is a user-friendly application that uses its user-friendly interface to detect packers, cryptors, and compilers in PE executable files — it has a greater detection rate than other similar tools because it packs over 600 distinct signatures in PE files.

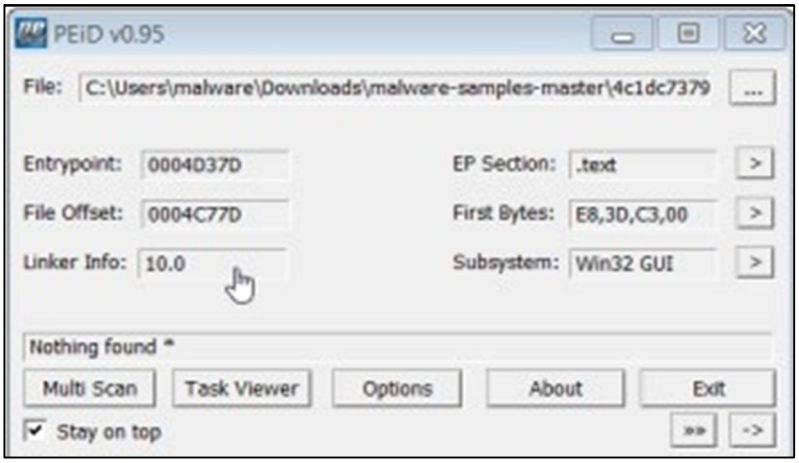


Figure 2: Application of PEiD

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue X Oct 2021- Available at www.ijraset.com

First and foremost, debugging of the binary file was done. It was observed that it was an "exe file". Using the analysis tool, Razare2 (Figure 3), it was noticed that it installs the random program and makes changes to Windows default entry. When the binary file was opened in Notepad, it was discovered that there are few language options, which are common in Ransomware software. Another suspicious activity was discovered at EOF; it is requesting Windows Admin access to windows Settings. In addition, it was trying to access the other directories. Figure 3 depicts that it is trying to make changes to the registry of Windows. The red commands are just string concatenation followed by copying to multiple locations as demonstrated in figure 4.

Figure 3: Activity in Registry

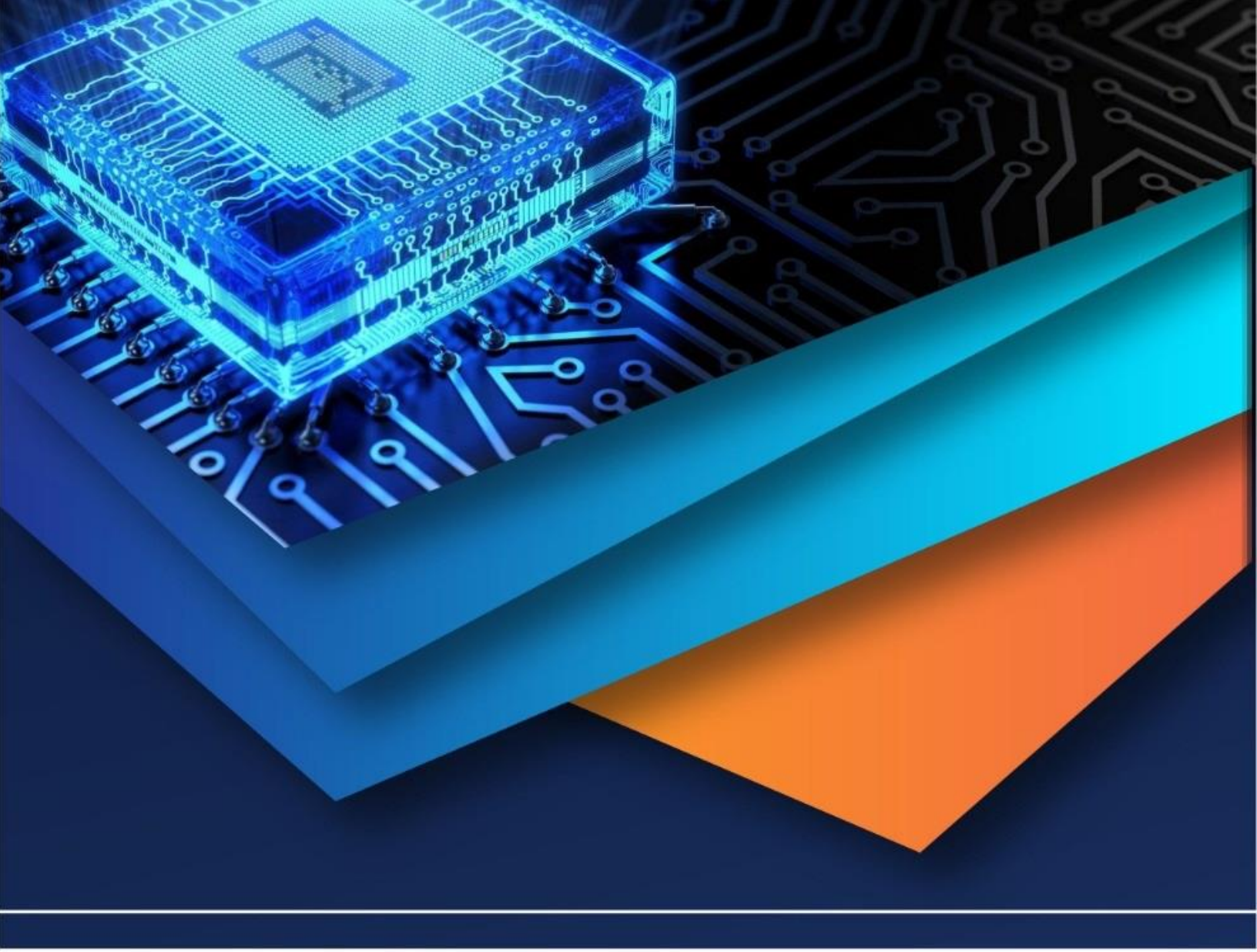
Figure 4: string concatenation

V. CONCLUSION

When malware is the source of a security threat, malware analysis enters the picture and plays an important part in developing an incident response. It guides users through the necessary stages for recuperation. It assists responders in determining the scope of a malware-related incident and identifying the affected hosts, servers, or systems. Malware analysis also produces actionable data that aids organizations in avoiding or mitigating the risks posed by malware. It aids in the prevention of further compromise. The current study explains the static and dynamic Malware Analysis along with its stages. Nevertheless, the main aim of the study was to conduct the Malware Analysis which was successfully conducted using Razare2.

REFERENCES

[1] Gray Hat Hacking 2nd Edition McGraw Hill by Shon Harris
[2] Symantec Corporation, Internet security threat report2013, Volume 18
[3] Ammar Ahmed E. Elhadi, Mohd Aizaini Maarof and Ahmed Hamza Osman, Malware Detection Based on Hybrid Signature Behaviour Application Programming Interface Call Graph, American Journal of Applied Sciences 9 (3): 283-288, 2012, ISSN 1546-9239, 2012, Science Publications
[4] Asaf Shabtai, Robert Moskovitch, Clint Feher, Shlomi Dolev and Yuval Elovici, Detecting unknown malicious code by applying classification techniques on OpCode patterns, Security Informatics 2012, 1:1,
[5] Imtithal A. Saeed, Ali Selamat, Ali M. A. Abuagoub, A Survey on Malware and Malware Detection Systems, International Journal of Computer Applications (0975 – 8887) Volume 67– No.16, April 2013
[6] Jonathan joseph bloun, adaptive rule-based malware detection employing learning classifier systems, Thesis Master of science in computer science, Missouri University of science and technology, 2011.
[7] Kirti Mathur, Saroj Hiranwal, A Survey on Techniques in Detection and Analyzing Malware Executables, International Journal of Advanced Research in Computer Science and Software Engineering, ISSN: 2277 128X, Volume 3, Issue 4, April 2013
[8] Pham Van Hung, An approach to fast malware classification with machine learning technique, Keio University,5322 Endo Fujisawa Kanagawa 252-0882 JAPAN, 2011
[9] Raja Khurram Shahzad, Niklas Lavesson, Henric Johnson, Accurate Adware Detection using Opcode Sequence Extraction, in Proc. of the 6th International Conference on Availability, Reliability and Security (ARES11),Prague, Czech Republic. IEEE, 2011, pp. 189195.
[10] R. K. Shahzad, S. I. Haider, and N. Lavesson, Detection of spyware by mining executable files, in Proceedings of the 5th International Conference on Availability, Reliability, and Security. IEEE Computer Society, 2010, pp. 295302.
[11] R. K. Shahzad and N. Lavesson, Detecting scareware by mining variable length instruction sequences,in Proc. of the 10th Annual Information Security South Africa Conference (ISSA11), Johannesburg, South Africa. IEEE, August 2011, pp. 18.
[12] Robiah Y, Siti Rahayu S., Mohd Zaki M, Shahrin S., Faizal M. A., Marliza R.,A New Generic Taxonomy on Hybrid Malware Detection Technique, (IJCSIS) International Journal of Computer Science and Information Security, Vol. 5, No. 1, 2009
[13] Robin Sharp, An Introduction to Malware, Spring 2012. Retrieved on April, 10, 2013
[14] Ronghua Tian, An Integrated Malware Detection and Classification System, Changchun University of Science and Technology, Thesis, August, 2011



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY