An Emerging Malware Analysis Techniques and Tools: A Comparative Analysis

Arkajit Datta[1], Kakelli Anil Kumar[2], Aju. D[3]
[1]Btech-Student, SCOPE, Vellore Institute of Technology, Vellore, India
[2,3] Associate Professor, SCOPE, Vellore Institute of Technology, Vellore, India

***Abstract*— The term Malware denotes malevolent software. This type of software is installed in the system without the awareness of its owner. This malicious software is being installed by the third party to steal, damage, corrupt the important and personal data of the user. The Malware writers have to get an upper edge in spreading their spiteful software such as – worms, trojan horses, spyware, viruses, rootkits, cookies, adware, and many more through the world of immense networking – the Internet. Antivirus vendors are receiving a thousand potentially malicious software every day which can affect the systems. Using Antivirus scanning (AVS) and firewalls to detect the malware is not enough as the malware writers are always way ahead creating numerous unsolvable and challenging threats for the computer society. This survey provides a rundown on the study of Malwares, tools, and techniques which can be useful to analyze malicious software, focusing on the online available malware analysis tools which work on cloud computing in giving results. Further, we focus on devising an Analysis Algorithm which aims and suggests various tools to execute the malware analysis procedure.**

Keywords*— Malware Analysis Techniques, Sandboxing, Static Analysis, Dynamic analysis, Detection Methods

I. INTRODUCTION

The Internet has become an indispensable part of our life as it connects us to the entire world virtually but it opens the path for various people with malevolent intents, who strive to attack and harm legitimate users in different ways for various reasons. Most of the time the reason is money. A point to be noted in this respect is the use of malevolent software which is installed in the computer without the consciousness of its owner – this software can steal confidential data and also allow remote access which may cause the denial of services (DNS) in the system [4]. To protect legalized users from various threats, security vendors such as antivirus software provide detection and analysis procedures. Various online tools can dynamically analyze the malware and detect it, the tools use cloud computing hence they are more efficient and safer. The main idea of this study is to identify various online malware analysis tools and compare them based on their analysis. The rest of the paper is categorized in the following way - Section 2 describes the literature survey and the background research work. Section 3 describes the malware analysis and detection procedure. The overview of this study is presented in section 4. Section 5 contains the execution and algorithm development. Section 6 states the conclusion.

A. Objectives

The main objectives followed as –

1. Analyzing online and offline dynamic malware analysis tools.
2. Comparing the results based on the methods of analysis, the correctness of results, and time required to analyze the malware.
3. Gathering the reports of the malware analysis from the sites.

II. LITERATURE SURVEY

The research papers related to malware analysis stated various tools and techniques which can be potentially followed to detect and analyze the malware. There are two basic methods of analyzing the malware, one is Static and the other is Dynamic [2]. Most of the studies derives that, compared to the static analysis, dynamic analysis is much more efficacious and accurate [3]. Specific malware cases may show the characteristics of various sections at the same time. The tools must be powerful enough to detect different malware efficiently. There are different techniques identified under dynamic analysis – Process Call Monitoring, Process Parameter Analysis, Tracking of information, Instruction Tree, Auto-Start Extensibility.

Due to immense malware obfuscation, the results from static and dynamic analysis seem to be insufficient, this directs the security analysts to use machine learning, deep learning, and neural network techniques. Machine learning techniques for malware analysis have seen immense growth in the research field these days, there are features collected from static and dynamic analysis to predict the malware [26]. Even neural networks can be used to predict the malware from the raw bytes of the file [26]. The malwares could be divided into various families and then the resulting dataset should be used for machine learning, various ML algorithms were compared on various tools and the results state that Random Forest is the best algorithm for analyzing the dataset [27]. As there is an increasing number of malwares these days, researchers are suggesting the use of data visualization with ML which will increase the accuracy. The accuracy can be achieved up to 97.73% and the false positives are reduced by 81.17%, this technique is named Visual - AT [28]. Machine Learning can also be used to facilitate the analysis of Linux-based malwares it was found that – crypto-mining malware is permeating the IOT infrastructure, the level of sophistication is increasing and there is a rapid proliferation of new variants with minimal investments in infrastructure [29]. Reverse engineering can be used to understand the unlabeled samples and then using machine learning can help us predict which cluster of malwares is similar to this [29]. Due to the immense obfuscation of the malware the n-gram features of the malware are diluted, a new methodology of using n-gram features of the dissembled code and then using Machine Learning model for analysis [31].

III. MALWARE ANALYSIS AND DETECTION METHODOLOGY

To understand the maliciousness of the malware, it should be analyzed in two methods – 1. Static Analysis and 2. Dynamic Analysis. For detecting, if a given program is malware or not, it should be initially analyzed Statically and then dynamically [4]. The flow chart of various malware detection techniques has been shown in figure no. 1.

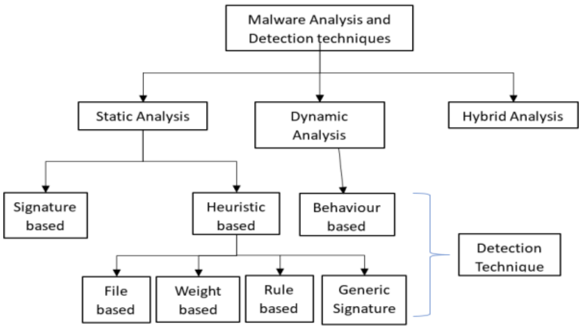

Fig. 1. Hierarchal representation of Malware analysis and detection techniques

A. *Malware Analysis Methods*

Static Analysis. By this method, we inspect the software without running it. In static analysis, the procedure is broken down by the technique of reverse engineering [3]. The static analysis compares the program with a huge database of various signatures using string and hashing mechanisms [4]. Though the static analysis can easily analyze and detect the known malware but fails for complex and new malware. Some of the advanced static analysis methods can analyze complex malware but these processes are quite cumbersome and require a lot of advanced knowledge in operating systems and disassembly.

Dynamic Analysis. Inspecting the characteristics of the strategies that have been executed while the program is being run determines dynamic analysis. It can be executed by various methods like monitoring the function calls, analyzing the function parameters, tracking the information flow, tracing the instructions, and AutoStart Extensibility Points [2]. The changes in files and registry, networking activities, and access to the RAM and various HDD (hard-drive) can also be analyzed and used for dynamic malware analysis. For detecting a new malware dynamic analysis is always preferred over static because even though the structure of the malware changes the behavior and the characteristics will never change and always remain the same which helps dynamic analysis to detect the malware easily [4]. In current times it can be observed that the malware has grown more intelligent, they stop executing as soon as it detects any behavioral analysis is being done. In these cases, Machine Learning outperforms both dynamic and static analysis methods [6].

B. *Malware Detection Methods*

Signature Based Detection Technique. Signature is a distinctive characteristic of a particular malware using which it can be easily detected. This technique is also known as – pattern matching, string matching, mask matching, and fingerprinting technique. A signature is a bit of sequence injected in the application program by malware writers, which uniquely identifies a particular malware [3]. Also, hashing done to the code of the malware with different algorithms such as SHA 256, MD 5, MD 4, etc. which is being used for comparing process later on.

Behavior-based Detection Technique. In this technique, the behavior of the program is used to decide whether a particular is malware or not. It doesn't look for the code and the code sequence. The main disadvantage of this method in Sandbox or a VM machine the malware doesn't always show its full potential and all the behaviors. So, it has been seen that this can lead to some incorrectly or wrong malware analysis [4].

Heuristic-Based Detection Technique. Known as the proactive technique, it is similar to the signature-based technique in detecting malicious code. For detecting the characteristics of the malware, the method of machine learning is been used. In this method, the API system calls, Opcodes, N-Grams, Control flow Graphs, and hybrid features are been implemented. The new feature comes here that – the malware detector will now search for the commands and instructions in the application program which were not been discovered earlier this helps in discovering various novel viruses and maintaining their signatures.

C. *Malware Detection Tools*

There are many tools for detecting malware and some of them have been identified and presented in the following tables. The tools are divided into two broad categories of static and dynamic malware analysis.

TABLE I. STATIC ANALYSIS TOOLS

Sr. no.	Tool's name	Illustration
1	PE id [4]	This tool helps in identifying complicated malware. It works on the signature-based detection process having almost 600 fingerprints of different malware.
2	PE view [9]	Information about the file headers and portable executables are being provided by this tool. Various description of the malware is accumulated from this tool including – compile-time and import/export functions.
3	PE explorer [4]	Having similarity with PE View this tool provides features - Files packed from malware packers such as UPX and Ns Pack can be unpacked.
4	Bin Text [10]	Character strings of a binary file can be searched and displayed using this tool
5	UPX [9]	Malware samples can be compressed using this tool. The tool uses the method of packing the executables
7	Dependence Walker [4]	This tool was implemented by Microsoft for static analysis which explores the DLLs and functions imported by the malware
8	Resource Hacker [11]	This tool is specially made and used in the Windows operating system. The tool is used to view, modify, add, and extract resources for both 32bit and 64bit Windows executables.
9	IDA pro [11]	This tool is very famous among malware analysts, reverse engineers, and vulnerability testers. This gives an interactive disassembling feature.
10	Hex Editors [4]	Binary files are viewed and edited using this tool.

TABLE II. TOOLS USED FOR STATIC MALWARE ANALYSIS

Sr no.	Tool's Name	Illustration
1	Process Explorer [12]	Having similarities with the task manager, this tool provides the currently running processes in a hierarchical view of processes.
2	Process Monitor [13]	Real-time file creation, file read, file writes, and file closure can be seen using this tool. This tool has other functions like – 1. Monitoring the registry and activity changes 2. Tracking processes and networks.
3	Reg shot [14]	Two registry snapshots are taken, one before and after the process to analyse the changes so that if any malicious activity is present, it can be easily detected by the reg shot.
4	Net cat [15]	A Tool to monitor inbound and outbound connections.
5	Wireshark [16]	This is a network packet analyser that has the potential to capture the network traffic generated by malware as soon as it was executed.
6	Olly Dbg [17]	When the source is not available this x86 debugger is used for the binary code analysis.
7	Burp Suite [18]	Security of web applications is tested using this tool. This tool can track various server requests posted by the malware to any remote server. All the HTTP and HTTPS requests made by the malware can be intercepted by this tool using the Man in the Middle Attack.
8	Sandboxes [4]	Sandbox is a virtual container that can be used for analysing untrusted programs/malware in a virtual system (installed inside or outside the main system which is relatively a safer environment) without hurting the main system. The sandboxes use both static and dynamic approaches to analyse the programs.

IV. OVERVIEW OF THE WORK

This section contains the description of the problem statement and the objectives identified to proceed.

A. Problem Statement

It has been found that Dynamic Malware analysis is always better than Static Malware analysis, but for analyzing the malware in the computers the malicious codes must be executed such that the tools can keep a check on the activities taking place in the computer. This procedure has a high probability of damaging the computer as the malware runs freely on the computer. There are various online-based malware analysis tools identified which works on cloud computing and online virtual machines. These tools provide efficient malware analysis and no harm to the user's computer. The goal of this paper is to analyze the different dynamic malware analysis tools mainly the online-based ones and compare them among each other for identifying the best.

The malware for the experiment purpose was downloaded from Tekdefence.com. The malware 1.exe was used for analysis in all the tools such that the results could be compared more efficiently.

V. IMPLEMENTATION, COMPARISON AND ALGORITHM ANALYSIS

The online and offline software's identified has been shown in table no. 3.

TABLE III. COMPARISON OF VARIOUS MALWARE ANALYSIS TOOLS (BOTH ONLINE AND OFFLINE)

Name	Description	Detection Method	Time Required	Result
Reverss (Anlyz) [20]	It provides automated dynamic malware analysis and helps Cyber Intelligence Response Teams (CIRT) to solve complicated malware problems effectively and steadily.	Reverse Engineering, Cognitive Analytics, Swift Reversal, Real-Time Classification, and comprehensive reporting	11 days (SLOW)	Malicious
Any run [21]	This is a cloud-based malware analysis tool. The malware can be uploaded and is analysed in the online Virtual Machine	Signature-based, behavioural, and heuristic-based analysis	1 hour (FAST)	Malicious
Valkyrie Verdict [22]	This tool has an effective database of malware which are used for static behavioural analysis. The analysis is very quick for the same reason. The malware has to be uploaded and then all the analysis is done on the cloud	Signature-based static malware analysis	Less than an hour (VERY FAST)	Malware Malicious
Virus Total [25]	Suspicious files can be uploaded which are compared with different anti-virus software and results are given that whether the signature is available in Anti-virus software. Even the URLs can be attached for scanning purposes.	Signature-based and Heuristic-based	Less than an hour (VERY FAST)	Malicious
Hybrid Analysis [23]	This is an advanced security tool which takes the suspicious file uploads and URL. It takes a deeper understanding of Windows and program code. It does both static analysis and multi-scan analysis. It even compares the results with the falcon's sandbox and virus total. This tool gives a threat score which can be taken as a	Signature-based, behavioural, and heuristic-based analysis	Less than 1 hour (FAST)	Malicious

	metric. Also, incident response and risk assessment reports are being provided.			
Intezer Analyse [24]	This is an advanced security tool in which we can upload malicious files and get the result. This tool uses a unique way of malware analysis, which is more efficient. This tool is also used to protect clouds. It even checks the result in virus total and gives it in the result. Also, organizations use this tool for their security from malicious software and codes.	Genetic – Software mapping - High confidence alerts- No manual configurations, rules, and policies	Less than an hour (FAST) [Required to create an account which takes time to get approved]	Malicious
Ghidra (offline)	It helps to analyse malware such as viruses and can also identify potential vulnerabilities in the system and the network. It can apprehend assembly, disassembly, decompilation, scripting and graphing, and various features. It aids various processor instruction sets and executable formats and supports both automated and interactive modes.	Reverse engineering	Less than an hour (VERY FAST)	Malicious
Olly DBG (offline) [17]	When the source is not available this x86 debugger is used for the binary code analysis.	Reverse Engineering	Less than an hour (VERY FAST)	Malicious
Reg Shot (offline) [14]	Two registry snapshots are taken, one before and after the process to analyse the changes so that if any malicious activity is present it can be easily detected by the reg shot.	Reverse Engineering	Less than an hour (VERY FAST)	Malicious

A. *Comparative Analysis of novel Algorithms*

Throughout the study, we have seen different types of malware analysis methods which can be used to detect malicious code. But there must be an algorithm that can be followed such that the analysis procedure becomes much easier to execute. Taking the basic components of malware analysis, a simple and naïve way of analysis approach can be built as shown in figure no. 2.

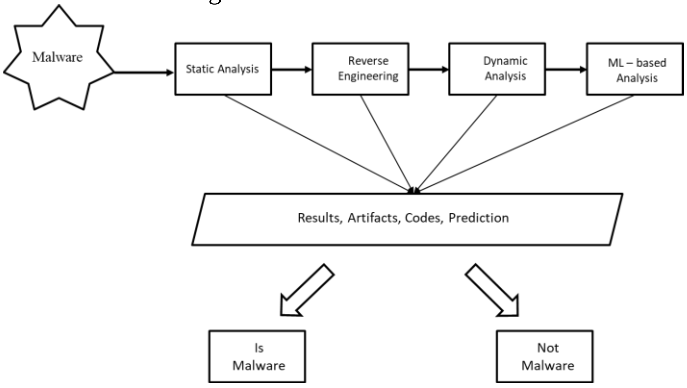

Fig. 2. Malware Analysis Sequence Representation using basic analysis techniques

This sequence can be formatted as an algorithm which will be represented as a flow chart depicting various tools which can be used at each step of the analysis. The tools have been selected based on the above comparison tables. The tools have to be numbered according to the order of higher precedence which was found from the study. The logic behind the flow chart is – The analysis starts from the basic static analysis, if the malware is detected in the very first step the result will be returned otherwise it'll be going through the other processes. The Flow chart is shown in figure no. 3. A functional algorithm snippet is being shown below.

Code Snippet –

```
void Static_Analysis();
void Reverse_engineering();
void Dynamic_analysis();
void ML_based_analysis();
Malware_detection(Malware_Sample){
        if(Static Analysis(Malware_Sample)==True){
                return Malicious
        }
        else
if(Reverse_engineering(Malware_Sample)==True){
                return Malicious
        }
        else if(Dynamic_analysis(Malware_Sample)==True){
                return Malicious
        }
        else if(ML_based_analysis(Malware_Sample)==True){
                return Malicious
        }
        else{
                return (!Malicious)
        }
}
```

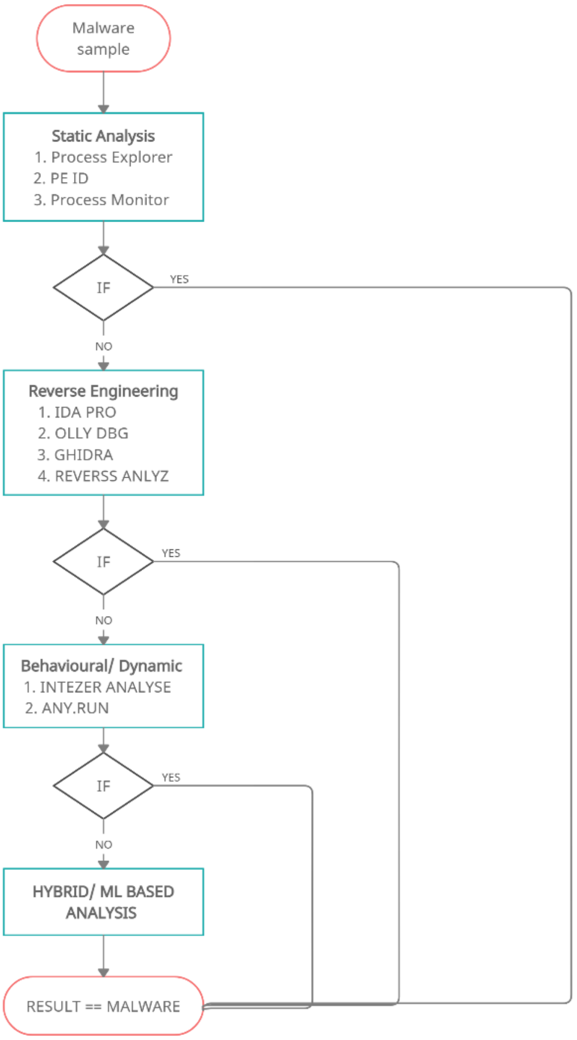

Fig. 3. Malware Analysis Sequence Representation using basic analysis techniques

VI. CONCLUSION

This research work has shown the different types of tools and ways in which a particular malware can be analyzed. Many kinds of research show that one single malware couldn't be analyzed in a single tool. Experimental results show that every malware analysis tool has a different metric and way to analyze the malicious code. Intezer Analyze software was recognized to be the best among the selected ones as it can take the output from Virus Total and simultaneously use a unique Genetic software mapping algorithm. This algorithm is comparing the genes with various other malware for coming to a solid result. The possible future work in this domain can be developing an algorithm by which we can use all the software simultaneously and get better results and protect the systems more efficiently.

REFERENCES

[1] Firdausi, I., Erwin, A., & Nugroho, A. S. (2010, December). Analysis of machine learning techniques used in behavior-based malware detection. In 2010 second international conference on advances in computing, control, and telecommunication technologies (pp. 201-203). IEEE

[2] Egele, M., Scholte, T., Kirda, E., & Kruegel, C. (2008). A survey on automated dynamic malware-analysis techniques and tools. ACM computing surveys (CSUR), 44(2), 1-42.

[3] Uppal, D., Mehra, V., & Verma, V. (2014). Basic survey on malware analysis, tools and techniques. International Journal on Computational Sciences & Applications (IJCSA), 4(1), 103.

[4] Aslan, Ö., & Samet, R. (2017, October). Investigation of possibilities to detect malware using existing tools. In 2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA) (pp. 1277-1284). IEEE.

[5] Sharma, A., Tripathi, G., Khan, M. S., & Kumar, K. A. (2015). A Survey Paper on Security Protocols of Wireless Sensor Networks. IJEIT.

[6] Ucci, D., Aniello, L., & Baldoni, R. (2019). Survey of machine learning techniques for malware analysis. Computers & Security, 81, 123-147.

[7] Gadhiya, S., Bhavsar, K., & Student, P. D. (2013). Techniques for malware analysis. International Journal of Advanced Research in Computer Science and Software Engineering, 3(4), 2277-128.

[8] Sharif, M., Yegneswaran, V., Saidi, H., Porras, P., & Lee, W. (2008, October). Eureka: A framework for enabling static malware analysis. In European Symposium on Research in Computer Security (pp. 481-500). Springer, Berlin, Heidelberg.

[9] Eilam, E. (2011). Reversing: secrets of reverse engineering. John Wiley & Sons.

[10] Prayudi, Y., & Riadi, I. (2015). Implementation of malware analysis using static and dynamic analysis method. International Journal of Computer Applications, 117(6), 11-15.

[11] Ligh, M., Adair, S., Hartstein, B., & Richard, M. (2010). Malware analyst's cookbook and DVD: tools and techniques for fighting malicious code. Wiley Publishing.

[12] Microsoft Sysinternals, https://docs.microsoft.com/en-us/sysinternals/downloads/process-explorer

[13] Microsoft Sysinternals, https://docs.microsoft.com/en-us/sysinternals/downloads/procmon

[14] Microsoft Sysinternals, https://docs.microsoft.com/en-us/sysinternals/

[15] Net Cat, Wikipedia, https://en.wikipedia.org/wiki/Netcat

[16] Wireshark homepage, https://www.wireshark.org/

[17] Olly DBG homepage, http://www.ollydbg.de/

[18] Port Swigger, Burp Suite, https://portswigger.net/burp

[19] Tekdefence.com – Malware database

[20] Analyz Reverss Homepage, https://sandbox.anlyz.io/dashboard

[21] Any.run homepage, https://any.run/

[22] Valkyrie Verdict, https://verdict.valkyrie.comodo.com/

[23] Hybrid analyse, https://www.hybrid-analysis.com/

[24] Intezer Analyze, https://analyze.intezer.com/

[25] Virus Total homepage, https://www.virustotal.com/gui/

[26] Gibert, D., Mateu, C., & Planes, J. (2020). The rise of machine learning for detection and classification of malware: Research developments, trends and challenges. Journal of Network and Computer Applications, 153, 102526.

[27] Mahajan, G., Saini, B., & Anand, S. (2019, February). Malware Classification Using Machine Learning Algorithms and Tools. In 2019 Second International Conference on Advanced Computational and Communication Paradigms (ICACCP) (pp. 1-8). IEEE.

[28] Liu, X., Lin, Y., Li, H., & Zhang, J. (2020). A novel method for malware detection on ML-based visualization technique. Computers & Security, 89, 101682.

[29] Carrillo-Mondéjar, J., Martínez, J. L., & Suarez-Tangil, G. (2020). Characterizing Linux-based malware: Findings and recent trends. Future Generation Computer Systems, 110, 267-281.

[30] Qamar, A., Karim, A., & Chang, V. (2019). Mobile malware attacks: Review, taxonomy & future directions. Future Generation Computer Systems, 97, 887-909.

[31] Pektaş, A., Eriş, M., & Acarman, T. (2011, August). Proposal of n-gram based algorithm for malware classification. In The Fifth International Conference on Emerging Security Information, Systems and Technologies (pp. 7-13).