Comprehensive Review of Literature on Malware Attacks: Trends and Insights

[1] Habiba Habeeb and [2] Insha Rafique

University of Management and Technology, Lahore

Correspondence:
Habiba Habeeb: Habiba.habib@umt.edu.pk

Article Link: https://www.brainetwork.org/index.php/jcai/article/view/18

DOI: https://doi.org/10.69591/jcai.v1i1.18

JCAI
ISSN (P):3007-3049
ISSN (E):3007-3057
JOURNAL OF COMPUTING AND ARTIFICIAL INTELLIGENCE
ARTIFICIAL INTELLIGENCE
Published by
Beyond Research Advancement and Innovation Network

Citation: Habib, H., & Rafique, I. (2023). Comprehensive review of Literature on Malware Attacks: Trends and Insights, Journal of Computing and Artificial Intelligence, 1(1), 66-83.

Conflict of Interest: Authors declared no Conflict of Interest

Acknowledgment: No administrative and technical support was taken for this research

Article History

Submitted: Mar 10, 2023
Last Revised: Apr 10, 2023
Accepted: May 28, 2023

Volume 1, Issue 1, 2023

Funding
No

An official Publication of
Beyond Research Advancement & Innovation Network, Islamabad, Pakistan

Comprehensive Review of Literature on Malware Attacks: Trends and Insights

Habiba Habib[1*], Insha Rafique[2]

University of Management and Technology, Lahore

Abstract

Modern malware is very smart and designed to attack the target. Most of these sophisticated malwares are quite persistent and have escape mechanisms. Software that is malicious is bad. Researchers in both academia and industry face significant hurdles as a result of this malware's ability to harm computers without the knowledge of their owners. Given that harmful software refers to any software that exploits computer-based systems with malware in order to compromise data security, privacy, and availability, the aim of this study is to examine the literature that has been produced on malware attacks. This will allow us to carry out an analysis of the literature and see how the research has progressed in terms of quantity, content and means of publication. One cannot understand all aspects of most malware programs as they are so large and sophisticated. The correct implementation and use of anti-malware programs, as well as the education of Internet users about malware attacks, are crucial steps in defending the identity of online shoppers from malware attacks. Some of the shortcomings of screening approaches that need to be corrected for better screening were noted in critical appraisal of the study.

Keywords: Malware, malware attack, anti-malware tools, malware detection, mobile malware

Introduction

Viruses, Trojan horses, spyware, and other intrusive code fall under the broad category of malware, which is widely used today. Malware analysis is a multi-step procedure that reveals information about the functionality and structure of the malware, allowing the expansion of a fix. To harm computer resources, dangerous software known as malware infiltrates remote targets, steals sensitive data and gains access to the system. Malware can operate in the background of the system in various forms (.exe, scripts,

[*] Corresponding Author: Habiba.habib@umt.edu.pk

dlls, files, macros, etc.). According to recent estimates, 80% of cyber-attacks worldwide [1, 2] are caused by modern malware

Literature Review

A malware instance was defined as software with a malicious goal by Christodorescu [2]. In this context, the term "malware" refers to a class of malicious code that includes viruses, Trojan horses, worms, and spyware. There is a strong network for code sharing and some malware writers struggle to read and understand older methodologies Malware writers employ generators, incorporate libraries, and copy other people's code. [Arief and Besnard], 2003. Malware was simply defined as malicious code in the investigation of [1,2,3]. By this definition, malware is any piece of software created with malicious intent. Many of the attack and damage paths used by malware are not covered by this concept. Examining the collection of information and two data leaks. In the SOA project literature review, [4] defined malware as any rogue program that interferes with an organization's database. Malware was described as "software with malevolent intent or malicious effect" by Aycock (2006) appeared in 1986 [1]. Spam messages, web fraud, theft of personal information such as MasterCard numbers, and other nefarious tasks such as ransomware are carried out using malware [2]. Rogue antivirus software. Accordingly, this study defines malware as any computer code created with malicious intent to breach security defenses and exploit vulnerabilities in IT infrastructures and digital devices, resulting in information gathering, data loss, information leakage, file infections buffer overflows, interruption of computer operations, and subsequent physical or operational damage, or both. Since then, numerous articles have been written about malware such as Trojan horses, worms, and viruses. But turning our attention from fiction to reality reveals that both malware and antivirus are major commercial enterprises today [5].

Malware Analysis

Malware analysis is a procedure used to look at the behaviours and components of malware and, if possible, to locate the attacker. Malware analysis is a multi-step procedure that reveals information about the functionality and structure of malware [6]. Static, dynamic, and hybrid examination are the three main methods for malware investigation. Each examination strategy has specific advantages and disadvantages, which have been discussed in this section.

Behavior Analysis

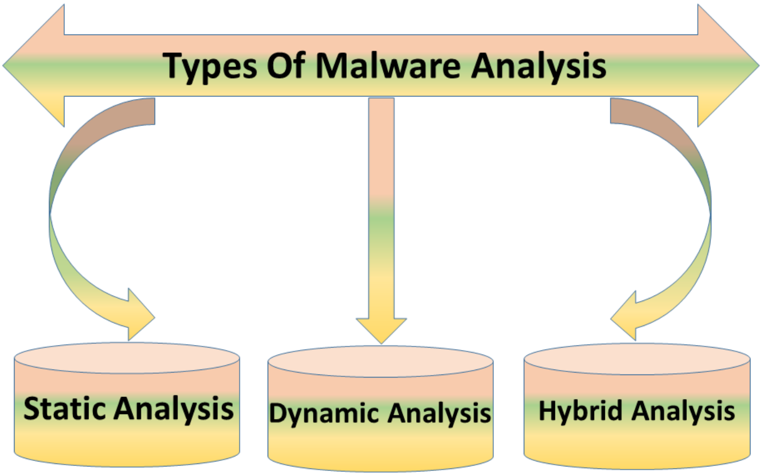
Examining malware behavior, such as registry operations, network operations, file transfers, and directory operations in the behavior analysis. Malware development occasionally introduces unintended problems [7]. Debugging is thus possible via behavior analysis. It's crucial to note this erroneous information. Debugging occasionally reveals details on the assailant Tools for behavior analysis include Process Monitor, Process Explorer, Regshot, Wireshark, CaptureBat, Cuckoo Sandbox, Anubis, Volatility, and others.

A. Code Analysis:

Dynamic and static analysis can be used to identify the targets, function, and operations of malicious software as well as how it is delivered.

Figure 1

Types of Malwares Analysis



B. Static Analysis

This analysis of programming that was not actually executing a program [6] is being done. Reproduction of this test involves a variety of processes. While others rely on the features of dual reporting, removing the "byte code courses of action" of a merged one, eliminating dual reporting and separating Op code progressions, to remove "control flow graphical ciphers" within the match registry and removing API calls, the like, and the like. Each of them covers the list of abilities, and some or many are used in malevolent ways [7].

C. Dynamic Analysis

It is the programming analysis that takes place while the programmed is running [7]. In one of the tests, such as API calls, structure calls, address tracking, dirty evaluation, vault modifications, memory creation, etc., may result in a piece of information. A step in the malware recognition process is described below using a dynamic one discussed previously.

D. Hybrid Analysis:

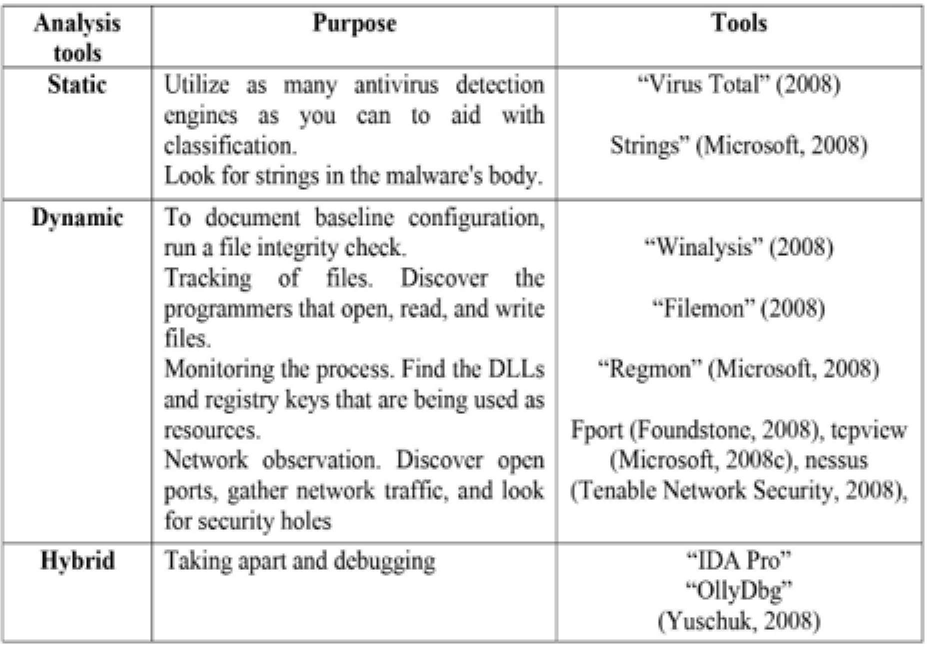
To overcome the drawbacks of each methodology, hybrid analysis uses approaches from both. Data leakage in Android apps can be found using a hybrid model 59 that employs both static and dynamic analysis methods. Static analysis investigates the root cause of an application's data loss in this way. On the other hand, a dynamic method observes the behavior of an application while it is running to find the malware [8].

E. Stages of Malware Analysis:

Malware analysis consists of four steps, which are sometimes represented as a pyramid diagram that gets more sophisticated as you proceed through the stages. We'll outline each of the four phase’s malware Analysis from the ground up for convenience's sake. Malware analysis is a procedure used to look at the behaviours and components of malware and, if possible, to locate the attacker [9].

Table 1

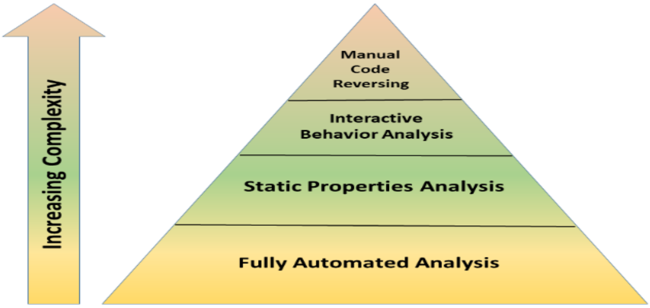
Types of Malware Analysis tool



Analysis tools	Purpose	Tools
Static	Utilize as many antivirus detection engines as you can to aid with classification. Look for strings in the malware's body.	"Virus Total" (2008) Strings" (Microsoft, 2008)
Dynamic	To document baseline configuration, run a file integrity check. Tracking of files. Discover the programmers that open, read, and write files. Monitoring the process. Find the DLLs and registry keys that are being used as resources. Network observation. Discover open ports, gather network traffic, and look for security holes	"Winalysis" (2008) "Filemon" (2008) "Regmon" (Microsoft, 2008) Fport (Foundstone, 2008), tcpview (Microsoft, 2008c), nessus (Tenable Network Security, 2008),
Hybrid	Taking apart and debugging	"IDA Pro" "OllyDbg" (Yuschuk, 2008)

Figure 2

Code Analysis Types



F. *Fully Automated Analysis:*

The use of detection models created by others for automated malware analysis examination of previously identified malware samples found in the wild. This approach is best for processing malware at scale and immediately determining how a sample will affect the network architecture. The study is useful in predicting potential effects in the event that virus gets within the network. It then produces a report that is simple to read with immediate fixes for security teams [10]. The best method for handling malware at scale is this.

G. *Static Properties Analysis:*

Without running the malware, static properties analysis examines the metadata of a file. You normally carry out this process in a remote setting that is cut off from the internet, such a virtual machine. Theoretically, static property analysis should give a malware analyst a good idea of whether to carry out further research or call it quits [11].

H. *Interactive Behavior Analysis:*

The following stage, behavior analysis, involves running the malware sample in isolation while the analyst monitors its interactions with the system and any modifications it makes. A malware piece will when a virtual environment is detected, programmers frequently refuse to operate or are designed to execute only with user input [12]. The following are some examples of actions that should immediately raise a red flag:

1. Including new or updated files
2. Implementing new procedures or services
3. Altering system settings or registry settings

I. *Manual Code Reversing:*

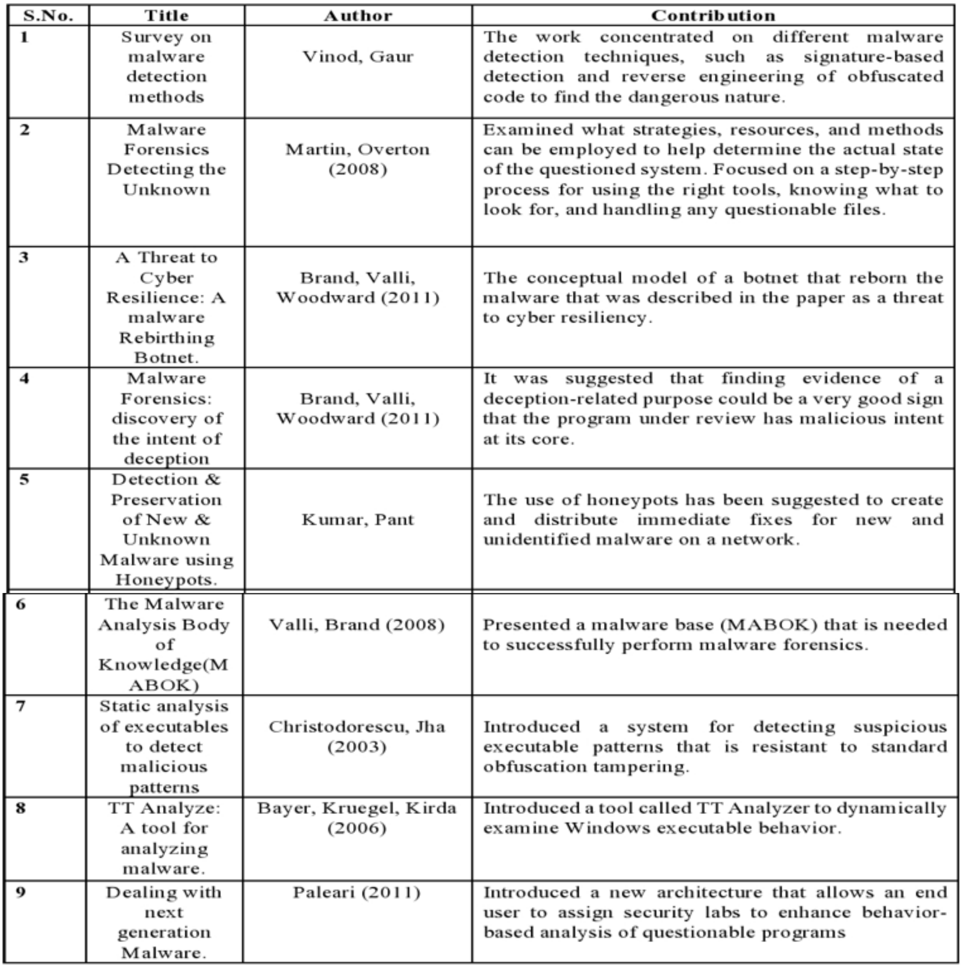
Debuggers, compilers, disassemblers, and other specialized tools are used by Analysts to manually reverse-engineer the code. By doing so, you can decrypt encrypted data, figure out how the malware algorithm works, and find any malware features that haven't yet been made apparent [13].

Methodology

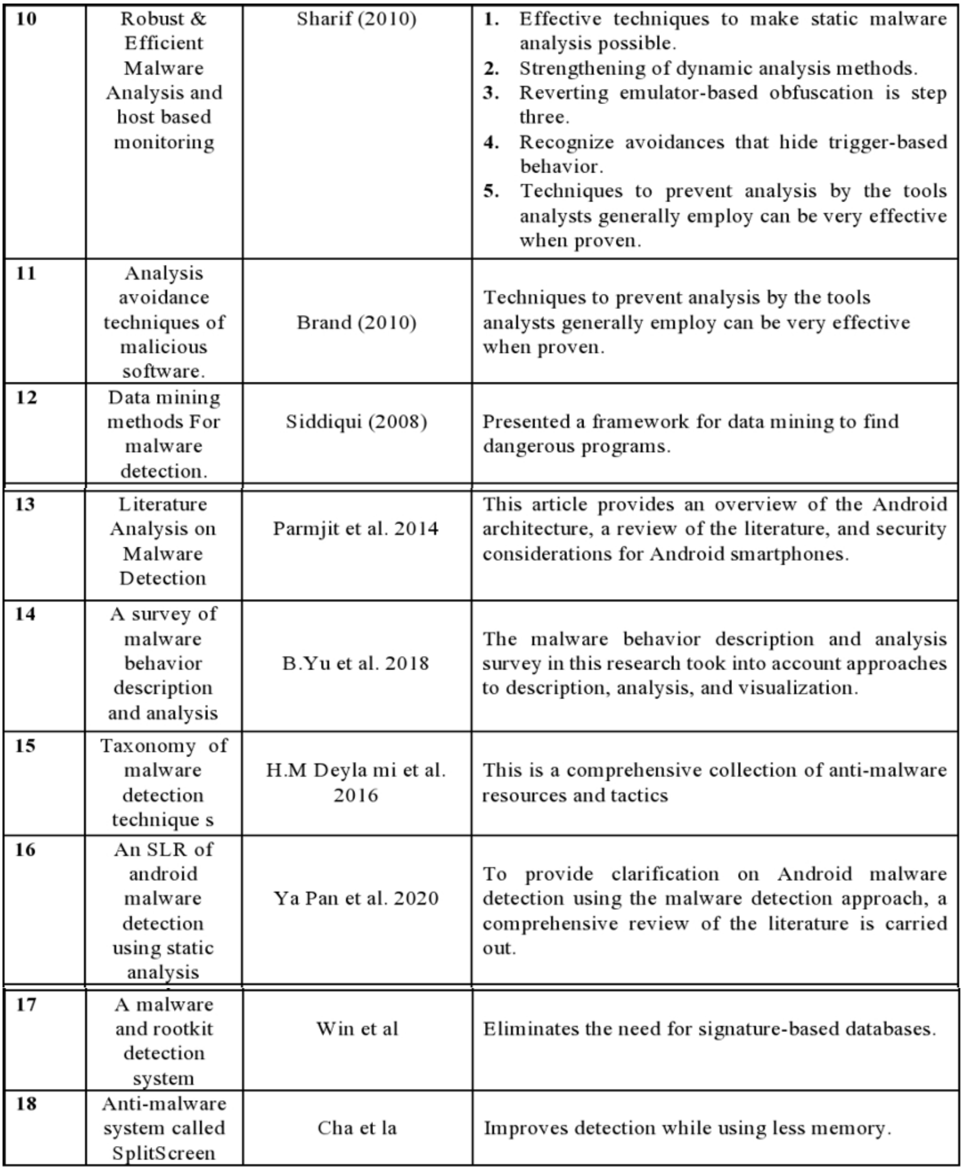
The methodology used for this part is based on recommendations made by a variety of authors. The growing social and economic effects of malware prompted numerous research initiatives, resulting in a large volume of research publications [14]. The scientific community has attempted to systematize these works over time by creating a series of surveys, often distinguished by various taxonomies and borders for example, while other documents cover the specific domain targeted by the virus (such as Internet of Things or mobile devices), many surveys focus on specific characteristics of malware (such as detection or evasion strategies). Several studies to find malware in the cloud environment are described in the literature review. Based on the core concept, the algorithms employed, and the feature extraction techniques, various malware detection strategies were examined [15]. Examination of this research reveals that none of the detection techniques were able to identify all malware, despite each having advantages over the others and performing better for specific cloud data sets. Split Screen is a new malware detection technique that Cha et al. suggested [16]. A second detection stage is used in this distributed malware detection system before the signature matching stage. The two-stage detection process in Split Screen is broke was included in ClamAV as an enhancement, increasing scanning performance with a signature collection that is more than twice as large while using only half the RAM. As the authors pointed out, as the number of signatures increases, Split Screen runs faster and uses less memory. The suggested approach is adaptable to many other low-end portable and consumer devices. The virtualization infrastructure that underpins cloud computing services has been the target of cyberattacks, according to research by Indirapriyadarsini [17]. They recommended a malware and rootkit detection solution to safeguard visitors from a variety of dangers. The guest kernel also included external monitoring, system call hashing, and other features. based on Support Vector Machines on the host. The suggested method design aims to put in place a system that recognizes the source of host attacks instantly and without the need for a signature database. They commended the suggested tactic's success against well-known user-level malware and kernel-level rootkit attacks as proof of its efficacy. For malware detection in the cloud, Gupta et al. introduced a cutting-edge paradigm [18]. Goals of this study to use various ways to identify harmful activities and alert guest virtual machines. This study combines the methods to detect Symbolic patterns, behavioral patterns, and patterns in behavior, symbolic representations, and DNA sequences. During the malware identification process, they used a file's DNA sequence to locate it. In the phase of symbolic detection, where data was organized according to file kinds, they employed symbols to find malware files. VMI-based techniques that can be used to examine IaaS cloud assaults were characterized by Chen et al. [19]. They focused on the implementation of virtual machines in IaaS cloud attacks that directly encrypt virtual machines. The target, source and direction of the attacks are taken into account in the classification process. They provide an overview of the possible attacks in which each actor would be in danger. CloudEyes is a solution for cloud-based malware detection as described by Sun et al. Devices have access to efficient security and data privacy. with minimal resources through CloudEyes. Based on the reversible structure, a new signature-based detection solution for cloud servers called Suspicious Bucket Cross Filtering has been proposed. It can offer proper and retroactive processing of harmful signature fragments to quickly characterize the suspicious file content in relation to the client summary's reversible sketch, a scan tool is used. Abdullahi [20] created a technique to prevent the transmission of malware in cloud systems. This study develops a two-tier epidemic model to prevent the spread of malware from one network to another and offers numerous layered defenses to address the problem. An intermediate monitoring server was used to create the malware detection system for multiple cloud servers in the proposed system, allowing malware to be scanned, detected, and removed before it is transferred to the cloud servers. A cloud-based malware detection tool called TrustAV was introduced by Follawo et al. [21]. In this approach a pattern matching method is used to identify tainted data. TrustAV promotes itself as a cloud-based solution and transmits malware analysis processes to a distant server. The study claims that even in cases when confidence is lacking, TrustAV can protect the transport and processing of user data. An instrument for detecting malware in the cloud is TrustAV was introduced by Boodai et al. [22]. In this approach a pattern matching method is used to identify tainted data. TrustAV promotes itself as a cloud-based solution and transmits malware analysis processes to a distant server. The study claims that even under dubious circumstances, TrustAV can protect the transmission and processing of user data. In order to locate malware on the system, Yadav explained how to use a WFCM-AANN unified malware detection approach [23]. Classification and clustering are covered in the study's two courses. In order to cluster the input dataset, the clustering module employs the WFCM method. into clusters. In the classification module, the discontinuous auto associative neural network receives the centroid of the clusters and is used to determine whether or not the information has been invasively entered. The author claims that the suggested classifier outperforms currently used classifiers in successfully identifying malware with a high detection rate. CloudEyes is a cloud-based malware detection tool, as described by Reshmi et al. [24]. For low-resource devices, CloudEyes provides effective security and data privacy. It has been suggested to use a new signature-based detection approach for cloud servers called cross-filtering of suspicious buckets. Processing of malicious signature fragments can be done correctly and retroactively. Using a scan tool, the suspicion of the file content regarding the reversible sketch client abstract is quickly defined. To safeguard data confidentiality and reduce the use of communication, an interaction system has been developed. Instead of sending the entire file, the client only sends the locations of the suspicious file segments [25]. They used suspicious and legitimate traffic to evaluate the performance of CloudEyes. The test results, the authors claim that their research demonstrates that CloudEyes is superior to other modern systems in terms of time usage and communication resource consumption. It's also practical and effective international conference journals and papers were chosen because they had been peer-reviewed before being published in journals or conference proceedings The table below provides a summary of the research articles (from serial numbers 1-10) and theses (from serial numbers 11–14) that were included in the review.

Table 2

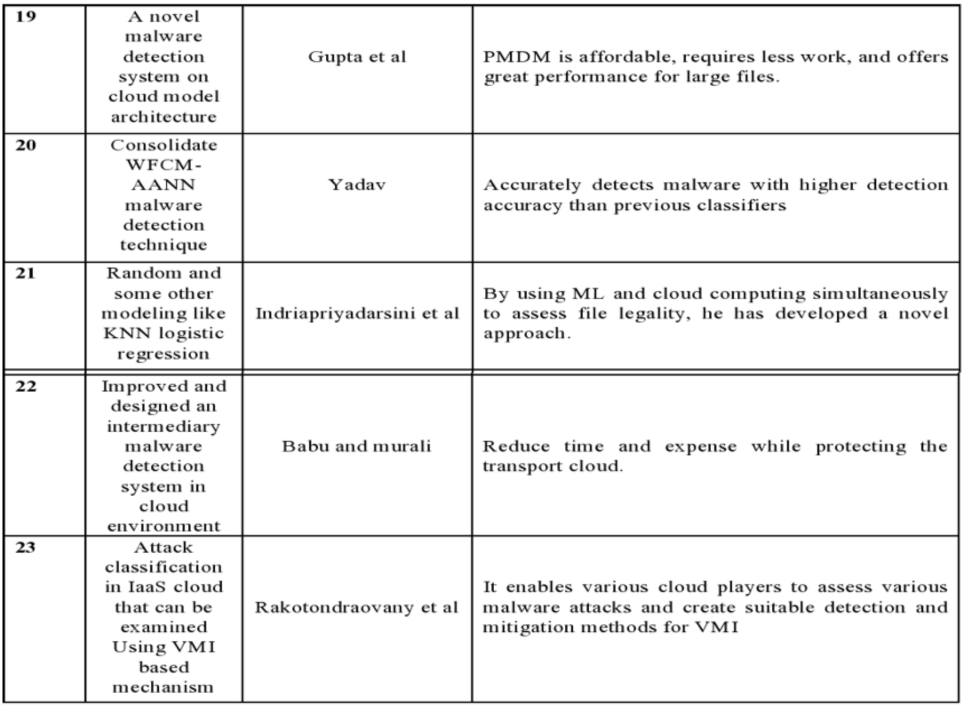
Related Work about Malware Analysis by various authors



S.No.	Title	Author	Contribution
1	Survey on malware detection methods	Vinod, Gaur	The work concentrated on different malware detection techniques, such as signature-based detection and reverse engineering of obfuscated code to find the dangerous nature.
2	Malware Forensics Detecting the Unknown	Martin, Overton (2008)	Examined what strategies, resources, and methods can be employed to help determine the actual state of the questioned system. Focused on a step-by-step process for using the right tools, knowing what to look for, and handling any questionable files.
3	A Threat to Cyber Resilience: A malware Rebirthing Botnet.	Brand, Valli, Woodward (2011)	The conceptual model of a botnet that reborn the malware that was described in the paper as a threat to cyber resiliency.
4	Malware Forensics: discovery of the intent of deception	Brand, Valli, Woodward (2011)	It was suggested that finding evidence of a deception-related purpose could be a very good sign that the program under review has malicious intent at its core.
5	Detection & Preservation of New & Unknown Malware using Honeypots.	Kumar, Pant	The use of honeypots has been suggested to create and distribute immediate fixes for new and unidentified malware on a network.
6	The Malware Analysis Body of Knowledge(MABOK)	Valli, Brand (2008)	Presented a malware base (MABOK) that is needed to successfully perform malware forensics.
7	Static analysis of executables to detect malicious patterns	Christodorescu, Jha (2003)	Introduced a system for detecting suspicious executable patterns that is resistant to standard obfuscation tampering.
8	TT Analyze: A tool for analyzing malware.	Bayer, Kruegel, Kirda (2006)	Introduced a tool called TT Analyzer to dynamically examine Windows executable behavior.
9	Dealing with next generation Malware.	Paleari (2011)	Introduced a new architecture that allows an end user to assign security labs to enhance behavior-based analysis of questionable programs



10	Robust & Efficient Malware Analysis and host based monitoring	Sharif (2010)	**1.** Effective techniques to make static malware analysis possible. **2.** Strengthening of dynamic analysis methods. **3.** Reverting emulator-based obfuscation is step three. **4.** Recognize avoidances that hide trigger-based behavior. **5.** Techniques to prevent analysis by the tools analysts generally employ can be very effective when proven.
11	Analysis avoidance techniques of malicious software.	Brand (2010)	Techniques to prevent analysis by the tools analysts generally employ can be very effective when proven.
12	Data mining methods For malware detection.	Siddiqui (2008)	Presented a framework for data mining to find dangerous programs.
13	Literature Analysis on Malware Detection	Parmjit et al. 2014	This article provides an overview of the Android architecture, a review of the literature, and security considerations for Android smartphones.
14	A survey of malware behavior description and analysis	B.Yu et al. 2018	The malware behavior description and analysis survey in this research took into account approaches to description, analysis, and visualization.
15	Taxonomy of malware detection technique s	H.M Deyla mi et al. 2016	This is a comprehensive collection of anti-malware resources and tactics
16	An SLR of android malware detection using static analysis	Ya Pan et al. 2020	To provide clarification on Android malware detection using the malware detection approach, a comprehensive review of the literature is carried out.
17	A malware and rootkit detection system	Win et al	Eliminates the need for signature-based databases.
18	Anti-malware system called SplitScreen	Cha et la	Improves detection while using less memory.



19	A novel malware detection system on cloud model architecture	Gupta et al	PMDM is affordable, requires less work, and offers great performance for large files.
20	Consolidate WFCM-AANN malware detection technique	Yadav	Accurately detects malware with higher detection accuracy than previous classifiers
21	Random and some other modeling like KNN logistic regression	Indriapriyadarsini et al	By using ML and cloud computing simultaneously to assess file legality, he has developed a novel approach.
22	Improved and designed an intermediary malware detection system in cloud environment	Babu and murali	Reduce time and expense while protecting the transport cloud.
23	Attack classification in IaaS cloud that can be examined Using VMI based mechanism	Rakotondraovany et al	It enables various cloud players to assess various malware attacks and create suitable detection and mitigation methods for VMI

Discussion

Remember that malware analysis is like a game of cat and mouse from the review that has been covered so far [26]. The results suggest that there are two main approaches to malware analysis, aside from a flaw in virtual machines: many researchers are unable to locate malware because it employs a variety of strategies to avoid analysis [27,28]. Malware authors respond to new malware analysis tools by creating new methods to evade analysis, thus identifying, analyzing, and ultimately developing remedies for them are separate academic issues. The efforts made to create, propagate, detect and spread malware are revealed in this investigation. The need to lessen their detrimental effects on the community is described in detail along with well-defined taxonomies [29].

Conclusion and Future Work

The usual risk of PC and correspondence frameworks damaging devices or stealing sensitive data is caused by malicious arracks. Malware attacks are growing more frequently as a trend. Networks are constantly attacked by malware due to their increased vulnerability. No one is immune from today's sabotage, whether it affects a single machine or an entire organization's network. In the end, we make some recommendations for future studies that will help in the development of more precise, effective, robust and scalable mechanisms for detection and attacks. A comprehensive security system is urgently needed to identify and stop the next malware attacks, according to a review of the literature and a practical analysis of available mitigation measures.

References

[1] A. Vance, “Flow based analysis of Advanced Persistent Threats detecting targeted attacks in cloud computing,” in *2014 First International Scientific-Practical Conference Problems of Infocommunications Science and Technology*, IEEE, 2014, pp. 173–176. Available: https://ieeexplore.ieee.org/abstract/document/6992342/

[2] L. Li *et al.*, “Understanding android app piggybacking: A systematic study of malicious code grafting,” *IEEE Trans. Inf. Forensics Secur.*, vol. 12, no. 6, pp. 1269–1284, 2017.

[3] A. F. A. Rahman, M. Daud, and M. Z. Mohamad, “Securing Sensor to Cloud Ecosystem using Internet of Things (IoT) Security Framework,” in *Proceedings of the International Conference on Internet of things and Cloud Computing*, Cambridge United Kingdom: ACM, Mar. 2016, pp. 1–5. doi: 10.1145/2896387.2906198.

[4] Y. Chen and F. M. Zahedi, “Individuals’ internet security perceptions and behaviors,” *Mis Q.*, vol. 40, no. 1, pp. 205–222, 2016.

[5] A. Shashwat, D. Kumar, and L. Chanana, “An end to end security framework for service oriented architecture,” in *2017 International Conference on Infocom Technologies and Unmanned Systems (Trends and Future Directions)(ICTUS)*, IEEE, 2017, pp. 475–480. Available: https://ieeexplore.ieee.org/abstract/document/8286056/

[6] B. Arief and D. Besnard, “Technical and human issues in computer-based systems security,” 2003. Available: https://kar.kent.ac.uk/58732

[7] H. J. Highland, "A history of computer viruses—Introduction," *Computers & Security*, vol. 16, no. 5. Elsevier, pp. 412–415, 1997. Available: https://www.sciencedirect.com/science/article/pii/S0167404897822456

[8] F. Mira, "A systematic literature review on malware analysis," in *2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*, IEEE, 2021, pp. 1–5. Available: https://ieeexplore.ieee.org/abstract/document/9422537/

[9] H. M. Deylami, R. C. Muniyandi, I. T. Ardekani, and A. Sarrafzadeh, "Taxonomy of malware detection techniques: A systematic literature review," in *2016 14th Annual Conference on Privacy, Security and Trust (PST)*, IEEE, 2016, pp. 629–636. Available: https://ieeexplore.ieee.org/abstract/document/7906998/

[10] S. K. Cha, I. Moraru, J. Jang, J. Truelove, D. Brumley, and D. G. Andersen, "SplitScreen: Enabling efficient, distributed malware detection," *J. Commun. Netw.*, vol. 13, no. 2, pp. 187–200, 2011.

[11] T. Y. Win, H. Tianfield, and Q. Mair, "Detection of malware and kernel-level rootkits in cloud computing environments," in *2015 IEEE 2nd International Conference on Cyber Security and Cloud Computing*, IEEE, 2015, pp. 295–300. Available: https://ieeexplore.ieee.org/abstract/document/7371497/

[12] M. K. Gupta, S. Shaw, and S. Chakraborty, "Pattern based malware detection technique in cloud architecture," *Inst Eng Manage Kolkata India*, 2016. Available: https://www.researchgate.net/profile/Sanjay-Chakraborty-2/publication/306291981_Cloud_Based_Malware_Detection_Technique/links/58a6b4bfaca27206d9a7b396/Cloud-Based-Malware-Detection-Technique.pdf

[13] N. Rakotondravony *et al.*, "Classifying malware attacks in IaaS cloud environments," *J. Cloud Comput.*, vol. 6, no. 1, p. 26, Dec. 2017, doi: 10.1186/s13677-017-0098-8.

[14] H. Sun, X. Wang, R. Buyya, and J. Su, "CloudEyes: Cloud-based malware detection with reversible sketch for resource-constrained internet of things (IoT) devices," *Softw. Pract. Exp.*, vol. 47, no. 3, pp. 421–441, Mar. 2017, doi: 10.1002/spe.2420.

[15] "15. Babu, N. M., & Murali, G. (2017, August). Malware... - Google Scholar." Available: https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=15.%09Babu%2C+N.+M.%2C+%26+Murali%2C+G.+%282017%2C+August%29.+Malware+detection+for+multi+cloud+servers+using+intermediate+monitoring+server.+In+2017+International+Conference+on+Energy%2C+Communication%2C+Data+Analytics+and+Soft+Computing+%28ICECDS%29+%28pp.+3609-3612%29.+IEEE.&btnG=

[16] R. M. Yadav, "Effective analysis of malware detection in cloud computing," *Comput. Secur.*, vol. 83, pp. 14–21, 2019.

[17] P. Indirapriyadarsini, M. U. Mohiuddin, M. Taqueeuddin, C. S. Reddy, and T. Koushik, "Malware detection using machine learning and cloud computing," *Int J Res Appl Sci Eng Technol*, vol. 8, no. 6, pp. 101–104, 2020.

[18] D. Deyannis, E. Papadogiannaki, G. Kalivianakis, G. Vasiliadis, and S. Ioannidis, "TrustAV: Practical and Privacy Preserving Malware Analysis in the Cloud," in *Proceedings of the Tenth ACM Conference on Data and Application Security and Privacy*, New Orleans LA USA: ACM, Mar. 2020, pp. 39–48. doi: 10.1145/3374664.3375748.

[19] H. Chen, H. Leung, B. Han, and J. Su, "Automatic privacy leakage detection for massive android apps via a novel hybrid approach," in *2017 IEEE International Conference on Communications (ICC)*, IEEE, 2017, pp. 1–7. Accessed: May 19, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7996335/

[20] M. Abdullahi *et al.*, "Detecting cybersecurity attacks in internet of things using artificial intelligence methods: A systematic literature review," *Electronics*, vol. 11, no. 2, p. 198, 2022.

[21] O. I. Falowo, M. Ozer, C. Li, and J. B. Abdo, "Evolving Malware & DDoS Attacks: Decadal Longitudinal Study," *IEEE Access*, 2024. Available: https://ieeexplore.ieee.org/abstract/document/10471391/

[22] J. Boodai, A. Alqahtani, and K. Riad, "deep learning for malware detection: literature review," *J. Theor. Appl. Inf. Technol.*, vol. 102, no. 4, 2024. Available: http://www.jatit.org/volumes/Vol102No4/34Vol102No4.pdf

[23] A. Verma and C. Shri, “Cyber Security: A Review of Cyber Crimes, Security Challenges and Measures to Control,” *Vis. J. Bus. Perspect.*, p. 097226292210747, Feb. 2022, doi: 10.1177/09722629221074760.

[24] T. R. Reshmi, “Information security breaches due to ransomware attacks-a systematic literature review,” *Int. J. Inf. Manag. Data Insights*, vol. 1, no. 2, p. 100013, 2021.

[25] C. Senarak, “Port cyberattacks from 2011 to 2023: a literature review and discussion of selected cases,” *Marit. Econ. Logist.*, vol. 26, no. 1, pp. 105–130, Mar. 2024, doi: 10.1057/s41278-023-00276-8.

[26] F. A. Aboaoja, A. Zainal, F. A. Ghaleb, B. A. S. Al-Rimy, T. A. E. Eisa, and A. A. H. Elnour, “Malware detection issues, challenges, and future directions: A survey,” *Appl. Sci.*, vol. 12, no. 17, p. 8482, 2022.

[27] N. Z. Gorment, A. Selamat, and O. Krejcar, “A Recent Research on Malware Detection Using Machine Learning Algorithm: Current Challenges and Future Works,” in *Advances in Visual Informatics*, vol. 13051, H. Badioze Zaman, A. F. Smeaton, T. K. Shih, S. Velastin, T. Terutoshi, B. N. Jørgensen, H. Aris, and N. Ibrahim, Eds., in Lecture Notes in Computer Science, vol. 13051. , Cham: Springer International Publishing, 2021, pp. 469–481. doi: 10.1007/978-3-030-90235-3_41.

[28] M. U. Rehman, R. Akbar, M. Omar, and A. R. Gilal, “A Systematic Literature Review of Ransomware Detection Methods and Tools for Mitigating Potential Attacks,” in *Computing and Informatics*, vol. 2001, N. H. Zakaria, N. S. Mansor, H. Husni, and F. Mohammed, Eds., in Communications in Computer and Information Science, vol. 2001. , Singapore: Springer Nature Singapore, 2024, pp. 80–95. doi: 10.1007/978-981-99-9589-9_7.

[29] T. Jabar and M. Mahinderjit Singh, “Exploration of mobile device behavior for mitigating advanced persistent threats (APT): a systematic literature review and conceptual framework,” *Sensors*, vol. 22, no. 13, p. 4662, 2022.