iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 **Issue:** XII **Month of publication:** December 2023

DOI: https://doi.org/10.22214/ijraset.2023.57649

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: 7.538
Volume 11 Issue XII Dec 2023- Available at www.ijraset.com

A Static Approach for Malware Analysis: A Guide to Analysis Tools and Techniques

Riya Nair[1], Kiranbhai R Dodiya[2], Parth Lakhalani[3]
[1]*Department of Forensic Science, PIAS, Parul University, Vadodara, Gujarat, India*
[2]*Research Scholar, Department of Biochemistry and Forensic Science department, Gujarat University, Ahmedabad, Gujarat, India.*
[3]*Assistant Professor, Department of Forensic Science, PIAS, Parul University, Vadodara, Gujarat*

Abstract: ***Malicious code presents a severe risk to computer systems, making work difficult for information security and cyber experts. Malware analysis is in great demand because of its importance and function in digital forensics and cyber security. Malware, often known as malicious software, is purposefully written software that harms or damages people, computers, servers, or networks. An overview of malware analysis methods and techniques in the fields of digital forensics and cyber security is given in this article. The study examines several malware types, their characteristics, and analysts' challenges in locating and analyzing them. It also highlights the importance of continuing this field's research and development to stay ahead of evolving malware threats. Trojan horses, worms, backdoors, rootkits, and adware are examples of malware. There are several methods for analyzing malware, but one of the most well-known is static analysis. This article will look at several methods for doing malware analysis and detection on corporate systems, as well as the resources available to assist with sample inspection to reduce the impact of malware assaults on an organization's operations. The investigator must first choose which methods and instruments to use for analysis. Static analysis, which includes malware scanners and detectors, is the first line of defense against malware. As technology advances, malware creators use various techniques to conceal their source code from scanners and detectors that search for strings, pattern matching, and other similar patterns to determine hash values that may be used to identify the infection. Malware experts decompress the packed file into unpacked one to examine obfuscated malware. This study examines efforts to investigate the many techniques and instruments malware uses in the real world.***

***Keywords*— malware, static analysis, portable executable, malware tools, malware detection, classification, code analysis**

I. INTRODUCTION

Malware is famous and among the most successful kinds of online fraud. "Malware," short for "malicious software," is intended to do damage, disrupt regular operations, or gain unauthorized access to a computer. Attackers frequently employ malware to steal confidential data, obstruct regular computer functions, or propagate harmful software to other computers. It can increase through malicious software downloads, email attachments, and exploiting operating system and application flaws. It is dangerous because different malware have their style of actions, and their intentions are likely to cause some damage that could be unrecoverable. Earlier in history, they were mainly created for fun and to annoy, such as slowing down their systems or annoying pop-ups. Still, eventually, malware became more malicious. It progressed to stealing, encrypting files or locking systems unless a ransom is paid, also known as ransomware, scamming them, tricking using various social engineering techniques, shutting down essential servers or making them slow as dos attacks can heavily impact the clients and reputation of the organization there is a specific type of malware which can create botnets in which malware are made to take part in the more significant attack and perform various attacks.

Most often, malware impacts the Windows operating system from Microsoft. Given that it is among the most well-liked and extensively used operating systems, malware programs are written by cyber attackers or malware authors, and they can be one of the most lethal attacks on an organization because malware often goes undetected in a network for several weeks or is hidden by obfuscating into some legitimate application. Every day, these malware authors get more competent and try to find a way to make it more perfect, but as the slogan goes, every criminal leaves their traces. Spyware is malicious software that monitors and records the activity on a target's computer. In addition, there are logic bombs that set instructions to execute at a specific time and can be used as botnets for DDOS attacks. Finally, backdoors are the most common and dangerous malware, as they permit the assailant to access the victim's device remotely, potentially destroying or altering anything they want. Also known as RAT-Remote Access Trojan, malicious software may cause anything from a slight nuisance to severe financial loss and interruption; thus, it is crucial to safeguard computer systems.

©IJRASET: All Rights are Reserved

A. Malware Basic Classification

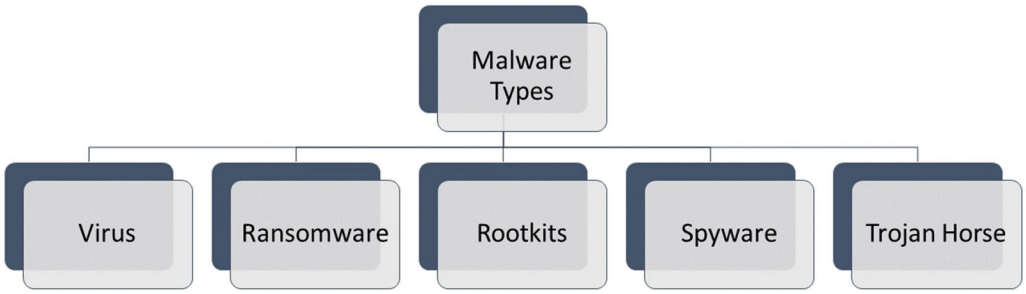


Figure 1.1: Types of malware

A computer user is never permitted to load what is known as malicious software or simply malware. This suggests the user may locate the software on their machine without knowledge. Additionally, malware is computer software that acquires personal data from any machine without the user's knowledge or agreement. As its name says, this software is dangerous and can seriously damage a computer, a mobile device, or other connected devices. This injury can be modest, such as changing the author's name on a piece of paper, or severe, causing equipment to malfunction totally. The good news is that malware can be stopped or managed. In truth, protecting a gadget from malware threats does not cost a fortune. It also requires a sufficient understanding of them. ("12 Types of Malware + Examples You Should Know," n.d.) However, it is crucial to comprehend the varied areas used to characterize harmful software to be able to accomplish this. Malware is a broad term with many classes and their corresponding sub-classes. Here, we will look at some of the malware types that are most popular with attackers. ("Types of Malware and Malware Examples," n.d.)

1) *Virus:* A virus is a harmful software program that contaminates systems and tampers with data. Computer viruses are designed to get into networks, make disastrous errors, lead to security breaches, and erase data. Knowing how computer viruses spread across systems and applications by design is essential. Computer viruses often attach themselves to executable host files, enabling their infectious code to start operating when the file is opened. The linked programs or document spreads the infection across networks, DVDs, file-sharing programs, and email attachments. (Chess, Conference, and 2000 n.d.)
2) *Worms:* A computer worm is a virus that spreads around networks and reproduces. Unlike viruses, worms may proliferate and propagate to other computers by exploiting weaknesses in host systems' security. They proliferate via emails, network connections, or software flaws and present risks, including data theft and service interruption. Because of their rapid and autonomous spread, worms pose a severe threat and need the usage of firewalls, software updates, and user awareness in cybersecurity measures. (Pratama, Computer, and 2012)
3) *Trojan Horse:* A Trojan Horse virus is a kind of malicious software that may sneakily enter a computer system by masquerading as a reliable program. Using social engineering techniques, hackers hide dangerous malware in what seems to be trustworthy software, giving them access to the user's machine without authorization. Trojans are malware that often spread to users' devices via email attachments or free downloads. After installation, the malicious malware carries out its planned tasks, which include monitoring individuals' online activity, stealing sensitive data, and gaining illegal access to corporate networks. A Trojan's existence may often be determined by unusual behavior on the device, such as sudden changes to the computer's settings. (Applied and 2015 n.d.)
4) Spyware: Spyware, categorized as intrusive software, discreetly installs itself on a computer, initiating covert surveillance of an individual's online activities without their knowledge or consent. This form of software clandestinely collects information about a person or organization and transmits it to external entities. Often interchangeably labeled as "adware," spyware may involve marketers or marketing data businesses seeking to gather user information. Its deployment occurs unbeknownst to the user through drive-by downloads, Trojan horses embedded within seemingly legitimate applications or deceptive pop-up windows. The transmitted personal information encompasses names, addresses, browsing patterns, interests, hobbies, and downloaded content, all shared via the Internet. Certain spyware strains can hijack browsers, compelling them to open multiple websites, initiate automated calls or text messages, or display intrusive advertisements even in offline scenarios. Notably, spyware that can steal usernames, passwords, or other credentials is classified as a "key logger," a malicious precursor to cybercrime. (Sipior, Ward, and Roselli, 2005)

5) *Rootkits:* A "rootkit" is a malicious program that provides hackers access to and controls a targeted device without authorization. While rootkits can occasionally affect a computer's firmware and hardware, most applications only affect the operating system and software (Bickford et al., n.d.). Rootkits are still in use, even though they are pretty good at hiding their existence. Rootkits are dangerous because they give hackers unrestricted access to compromised computers. This can lead to various threats, like the unauthorized extraction of personal and financial data, the installation of more malware, or the use of infected computers as part of a botnet for activities like spam distribution and DDoS attacks. Because rootkits are so sneaky, it's critical to have strong cybersecurity measures in place to identify and successfully block their presence.
6) *Ransomware:* A powerful type of cyberattack known as ransomware seriously jeopardizes user privacy by encrypting files or locking screens and preventing users from accessing devices, all while demanding ransom payments. Specific file formats are encrypted by crypto-ransomware, which demands payment for the decryption keys through pre-arranged web channels (O'Gorman and McDonald, 2012). Users must be cautious when using these sources because ransomware can be inadvertently downloaded by users through malicious websites, email attachments, or malware payloads.

B. Malware Analysis Method

The process of looking at malware to understand its purpose and manner of operation is known as malware analysis. This is essential in recognizing the virus's danger and creating defenses against it. Malware analysis analyzes and extracts as much information as possible from a malware sample or binary. "The information we gather assists us in comprehending the malware's capabilities, the method of its system infiltration, and future defense strategies." The goals of malware analysis are to identify its type and functionality, how the system was affected, and how it communicates with the attacker. For example, is it a RAT tool connecting to the C2 (command and control center) or communicating back to the server so we can understand the infrastructure here? Is it a target attack or a commodity attack? What kinds of information can we pull, such as registry, entry point, origin, and file signature, that will be helpful for future detection? Most anti-virus companies work on these incident samples. How can one prevent such types of attacks in the future? What traces do they leave behind? Malware analysis can be done by utilizing static or dynamic methods.

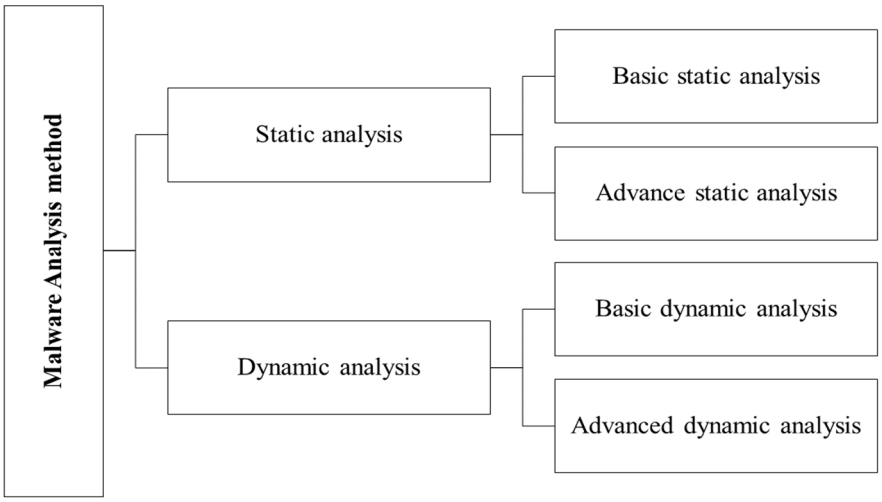


Figure 1.2: Malware Analysis Workflow

1) Static Analysis of Malware

Static malware evaluation is the look at malware that isn't constantly being accomplished. To apprehend the behavior and goals of the virus, this approach looks at its code and shape. The static evaluation aims to comprehend the virus's damage volume without endangering the community or gadget. Once the binary code of the contamination has been deconstructed, readable strings are taken out and compared to acknowledged malware signatures before being utilised for static analysis. The evaluation's findings may be used to increase preventive measures and understand the ailment's behavior and consequences. Static malware analysis, while mixed with dynamic malware analysis, might also offer a whole view of the behavior and effect of the contamination. Without executing the malware, safety professionals may additionally examine its structure and code to examine loads about its behavior and create defenses.

There are several steps involved in static malware evaluation, several of which are indexed below:

a) *Obtain a Malware Sample:* This can be achieved in numerous methods, including obtaining it from an internet site or a public repository or taking the pattern from an infected machine.
b) *Analysis of the Document:* File evaluation delves into the attributes of a document, together with its type, size, date of advent, etc.
c) *Disassembly:* The worm's binary code is damaged to reveal the underlying machine code and meeting language directives. This may forge mild on the malware's competencies.
d) *Strings Extraction:* Interpretable strings can be extracted from the binary code to discover critical facts, record locations, community addresses, and API requests.
e) *Signature Analysis:* One can confirm the malware's family and behavior by evaluating this system's code to regard malware signatures.
f) *Decompilation* is changing machine code returned into a better-level language, which includes Python or C. This aids in clarifying the virus's functionality and coding structure.
g) *Reverse Engineering:* The payload, any doubtlessly risky conduct, and the functionality and moves of the malware can all be ascertained by reading its code.

These are some of the moves in static malware research that employs a static method. Rather than using dynamic evaluation, the aim is to learn as much as possible about the virus without jogging it.

2) Dynamic Analysis of Malware

Malware that is dynamically analyzed has been subjected to controlled environments. Security experts may additionally learn extra about the hazard provided by malware and assist in developing defenses against it by using dynamic analysis to look at how viruses behave as they interact with the device. A malware sample is acquired, remoted in a virtual machine or sandbox to do a dynamic analysis, and determined during operation. The virus's behavior, such as its contacts with the gadget and network, is monitored. Data approximately report machine and registry modifications, method execution, and community site visitors are also logged. Facts for dynamic evaluation are accrued to comprehend the behavior of the virus and its impact on the gadget and to broaden defenses against it. The infection's behavior and impact can be understood by combining static and dynamic analysis. Dynamic evaluation is a valuable method for investigating malware and its behavior. Dynamic malware analysis is essential to comprehending and combating malware threats since it permits protection researchers to execute the virus in safe surroundings, look at its sports, and get the specified expertise. To do a dynamic malware analysis, carry out the following steps:

First, accumulate the malware sample. Obtaining a pattern from an infected PC, an internet site, or a public repository can be used to try this.

a) *Isolation:* The virus is run in isolated surroundings, like a sandbox or digital device, to save it from impacting the device or network.
b) *Monitoring:* The malware's interactions with the device and community are monitored at some stage in execution, as are its actions. With this know-how, one may additionally apprehend how the virus features and affects the system.
c) *Record Network Hobby:* Network traffic generated using the contamination can be monitored and analyzed to learn more about the virus's behavior and its connections to command and manage websites.
d) *Logging:* To provide comprehensive knowledge of viral sports, information concerning the execution of processes, updates to the record gadget and registry, and other recorded actions.
e) *Analysis of Findings:* Information received during dynamic analysis is hired to comprehend the malware's conduct and results at the device. This information can create defenses against the infection and stop its dissemination.

C. Classifying Malware using Statistical Analysis Techniques

1) Static Analysis

Static analysis is the method of examining malware binaries without actually executing them. Information extraction from the functional virus is the aim. As a result, we will understand the kind of malware and its capabilities better. Even while it may not disclose the malware's whole operation, knowing this will help us examine the data more rapidly in the future. It will also help us determine where to concentrate. Static analysis is quicker than dynamic analysis and can go through the whole program code. However, because the malicious code was created by the attacker directly, it may have been updated to make it harder to understand.

Furthermore, the attacker may use binary obfuscation methods such as packing, polymorphism, metamorphism, and similar ones to avoid static analysis methodologies' disassembly and code analysis parts. Some techniques for preparing a sample for analysis are as follows:

a) *Define the Fundamental Structure of the Malware:* As most samples are aimed at Windows computers, it is essential to ascertain the target operating system, portable executable (PE), architecture (32- or 64-bit), and format support (dll, exe).
b) *Determine which Virus it is:* For every sample, generate a hash value that serves as a distinct identification. We can determine whether the item has been examined before by using the hash value; we need to visit websites like Virustotal. Finding file modifications made during the analysis and execution phases might be another factor.
c) *Strings:* This will show us what malware is capable of; we can examine both ASCII and Unicode characters independently and c and c registry entries.
d) *Packing and Obfuscation:* These techniques include packing the file and adding a second layer with a different format to make it harder to analyze, even when some malware is buried. These techniques are often employed to evade detection. This may disclose more information and is sometimes called obfuscation, unpacking, or DE obfuscation.
e) *PE Headers:* Since it provides a wealth of information on its functionality, this is the most crucial section of the study. In addition to the header, a PE (Portable Executable) file has other vital parts. These sections include some helpful information.

2) Static Malware Analysis Techniques

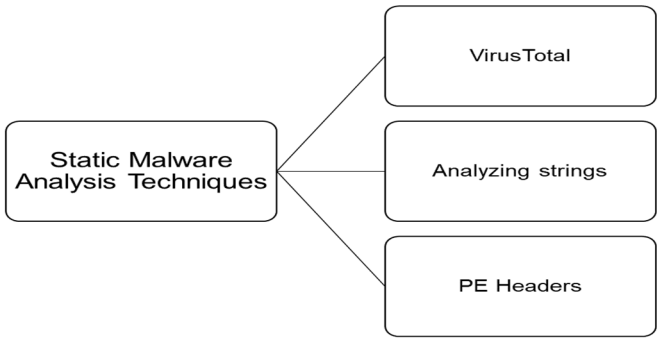


Figure 2.1: Static malware analysis techniques

a) Submitting the Findings to Virus Total

The starting approach in static analysis is to submit a suspicious executable to Virus Total, which is scanned against multiple antivirus programs, and the results are reported. Next, researchers may undertake malware threat hunting, relationship and behavior analysis, and scientific study on billions of malware samples with the help of Virus Total Intelligence, a sophisticated analytics layer over the Virus Total database. We can even use a command-line interface such as PowerShell by simply running the "Get-File Hash" command like this to get the hash result if PowerShell is installed on the machine.

b) Analyzing Strings

Obtaining details regarding software functionality Finding strings is an easy and quick process. When the software visits a URL, for example, it is seen that the URL was saved as a string within the application. Microsoft created a tool called "Strings". Strings can analyze any file and search for strings anywhere in a file. An executable that ignores formatting and context to extract ASCII and Unicode strings. (However, this implies it might mistakenly recognize character bytes as strings.) Strings search for a string consisting of three or more letters, a combination of ASCII and Unicode characters, and a string termination character.

c) Headers and Sections of Portable Executable File Formats

Beyond mere imports, a lot more data may be found in the headers of PE files. The PE file format consists of a header followed by several sections. The header is packed with information because it holds metadata about the file. After the header file, we may access the file's parts, which provide essential information.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

The following are the most typical and exciting portions of a PE file:

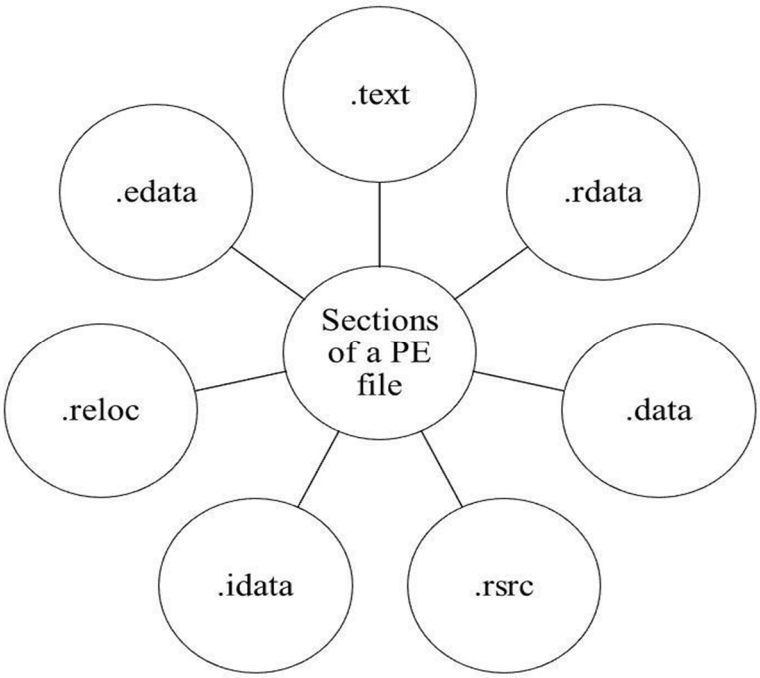


Fig. 2.2: Executable and its functions

Table 2.3: Exe Header Information

Sr. No.	Executable	Function
1	.text	This is where the executable code is stored.
2	.rdata	This section contains globally accessible read-only data.
3	.data	Stores global information accessed by the software.
4	.rsrc	This section comprises the resources that the executable needs.
5	.idata	Stores information about import functions and, if not found in this section, will be found in the .rdata section
6	.reloc	Gathers data for the relocation of library files.
7	.edata	Stores information about export functions and, if not found in this section, will be found in the .rdata section

If an analyst determines through static analysis that the executable will launch a process, and if the following exec and sleep commands are discovered. Still, no information regarding the corresponding DLL, which includes a function to connect with another server, is discovered. In that situation, the executable and the resource are both hidden. To learn more about the malware, use a program like Resource Hacker to open the—src part of the PE file.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

II. COMPREHENSIVE STUDY OF MALWARE ANALYSIS TECHNIQUES

Table 3.1 Comprehensive Study of Malware Analysis Techniques

Sr. No	Year	Topic	Detection Technique	Tools
1.	2017	Investigation of Possibilities to Detect Malware Using Existing Tools[1]	Tools for Static and Dynamic Analysis, Malware Analysis, Malware Detection, Malware Accuracy and Detection Rate	Static analysis tools, Dynamic analysis tools
2.	2017	Malware Detection Based on Multiple PE Headers Identification and Optimization for Specific Types of Files [2]	Multiple PE headers, parasitic viruses, malware detection, and file viruses	Cuckoo Sandbox
3.	2017	Performance Comparison of Static Malware Analysis Tools Versus Antivirus Scanners To Detect Malware[3]	Malware analysis, malware detection, static malware analysis tools, and performance comparison of antivirus scanners and malware static tools for malware detection.	Static analysis tools and Antivirus scanners
4.	2017	An Efficient Approach for Advanced Malware Analysis Using Memory Forensic Technique [4]	Malware, static analysis, memory forensics, malware detection, online malware analysis tools	Volatility tool, Ubuntu

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

5.	2017	Integrated Malware analysis using machine learning [5]	Machine learning, categorization, integrated method, anti-analysis technique, static analysis, and dynamic analysis	URLLIB, Norben sandbox
6.	2017	Bridesmaid: A Hybrid Tool for Accurate Detection of Android Malware[6]	Hybrid analysis, Battery consumption, Benchmarks test,	BRIDESMAID
7.	2017	Static analysis with paragraph vector for malware detection[7]	Support vector machines, malware, static analysis, machine learning, paragraph vectors, and K-nearest neighbor algorithms.	Paragraph Vector
8.	2017	Extracting the Representative API Call Patterns of Malware Families Using Recurrent Neural Network[8]	Classification of Malware, Recurrent Neural Network, and Typical API Call Pattern	RNN
9.	2017	Deep Ground Truth Analysis of Current Android Malware[9]	Malware Apps, Malware Samples, Malware Variants, Malware Behavior	VirusTotal, Anti-virus software
10.	2017	DroidSieve: Fast and Accurate Classification of Obfuscated Android Malware[10]	Analysis, Dynamic Analysis, Machine Learning, obfuscate, evade	DroidSieve

11.	2018	New results on permission-based static analysis for Android malware[11]	Android malware, static analysis, malware classification, mobile security	KNN Algorithm Naïve Bayes
12.	2018	A Study on Malware and Malware Detection Techniques[12]	Malware, malware analysis techniques, malware detection techniques	Hybrid analysis
13.	2018	Evaluating Machine Learning Models for Android Malware Detection: A Comparison Study[13]	Machine Learning, Malware, Classifier, Optimization, DREBIN, Google Play.	Discriminant
14.	2018	Machine Learning Aided Static Malware Analysis: A Survey and Tutorial[14]	Machine learning, Malware, Static analysis, Artificial intelligence	Multiple Machine Learning Algorithms
15.	2018	A Framework for Analysing Ransomware Using Machine Learning[15]	Reverse engineering, assembly instructions, DLL, machine learning, regular binaries, and ransomware detection	Machine Learning Classifiers

16.	2018	Attack and Defence of Dynamic Analysis-Based, Adversarial Neural Malware Detection Models[16]	Adversarial Learning, Dynamic Malware Classification	DNNs
17.	2018	Discovering optimal features using static analysis and a genetic search-based method for Android malware detection[17]	Android; malware; genetic algorithms; static analysis; and machine learning	Machine Learning Classifiers, VirusTotal
18.	2018	Construction and evaluation of the new heuristic malware detection mechanism based on executable files static analysis[18]	Malware, decision trees, neural networks, heuristic analysis, machine learning, and antivirus defense	Heuristic Analyzer Decision Trees
19.	2018	Effective and Explainable Detection of Android Malware Based on Machine Learning Algorithms[19]	Android Malware, SVM, probability statistics, feature extraction, dimensional	Machine Learning Classifiers Dynamic approach

20.	2018	Analytics: A Malware Detection Scheme[20]	Static analysis, binary level n-grams, word frequency shattering, malware detection, and the extreme learning machine	Simhashing ELM
`	2019	Feature Optimization for Run-Time Analysis of Malware in Windows Operating System using Machine Learning Approach[21]	Cuckoo sandbox, genetic algorithms, dynamic malware analysis, feature extraction, feature selection, and machine learning	Genetic Algorithm Vector Machine Naive Bayes
22.	2019	Static Malware Analysis to Identify Ransomware Properties[22]	Ransomware, malware detection, static analysis, dynamic analysis.	Static analysis Dynamic analysis
23.	2019	What Static Analysis Can Utmost Offer for Android Malware Detection[23]	Android malware, Android malware detection, static analysis, machine learning, Android	DroidAPI manner KNN SVM
24.	2019	Analysis of ResNet and GoogleNet models for malware detection[24]	Malware detection, Malware classification, Opcode, ResNet, GoogleNet	Hybrid Analysis, Deep learning-based malware detection ResNet
25.	2019	Review of Machine Learning Methods for Windows Malware Detection[25]	Windows, Portable Executable (PE), Features, Static Analysis, Malware, Benign, Classification	Static analysis tools ML classifiers

26.	2019	A Survey on malware analysis and mitigation techniques[26]	Evasion Malware, Malware analysis, Packers, Sandboxes, Advanced persistent threats	(Book) tools related to prevention
27.	2019	A Close Look at a Daily Dataset of Malware Samples[27]	Malware and its mitigation, Software reverse engineering, Economics of security and privacy, Malware, measurement, prioritization, classification	VirusTotal Sandbox
28.	2019	Malware Classification Using Early-Stage Behavioral Analysis[28]	dynamic approaches, classifiers, hybrid analysis, feature selection	Hybrid tools(OPEM) Dynamic Approaches(RNN, CNN)
29.	2019	Analysis and Comparison of Disassemblers for OpCode-Based Malware Analysis[29]	Disassembler, IDA Pro, Capstone, Udis, Zydis, Radare2, Distorm, Malware Detection, static malware analysis	Distorm Udis Zydis Radare2
30.	2019	AIMED: Evolving Malware with Genetic Programming to Evade Detection[30]	AIMED, Genetic Programming, Malware, Bytelevel perturbations, Adversarial learning	AIMED

31.	2020	Platform-Independent Malware Analysis Applicable to Windows and Linux Environments [31]	malware analysis, binary analysis, strings analysis, deep neural network, feature importance	BinText DNN
32.	2020	Advanced Windows Methods on Malware Detection and Classification [32]	API calls, feature development, malicious behavior analysis, dynamic analysis, anomaly-based detectors, malware detection, malware classification, and machine learning learning	API-based dynamic feature extraction using ML
33.	2020	Detecting Cryptomining Malware: a Deep Learning Approach for Static and Dynamic Analysis[33]	spyware called CryptoMining, Static analysis, dynamic analysis, and deep learning	Static analysis Dynamic analysis CNN LSTM
34.	2020	A Survey on Malware Detection and Analysis Tools[34]	Malware, Detection, Analysis, Tools, Machine Learning.	IDA-PRO OllyDbg LordPE
35.	2020	Static, Dynamic, and Intrinsic Features-Based Android Malware Detection Using Machine Learning[35]	Cybersecurity, Malware detection, Malware image, Convolutional neural network	--

36.	2020	Deep learning based Sequential model for malware analysis using Windows exe API Calls[36]	LSTM model, Cuckoo Sandbox, Virus Total Service, Windows API Calls	Sandbox, LSMT
37.	2020	Malware Elimination Impact on Dynamic Analysis: An Experimental Machine Learning Approach[37]	Static Analysis, Dynamic Analysis, Hybrid Analysis, Cuckoo Sandbox	--
38.	2020	DeepDetectNet vs. RLAttackNet: An adversarial method to improve deep learning-based static malware detection model[38]	RLAttackNet, deep learning, a static PE malware detection, DeepDetectNet, adversarial samples,	DeepDetectNet. UPX Reinforcement Learning
39.	2020	A dynamic Windows malware detection and prediction method based on contextual understanding of API call sequence[39]	API call sequence, Malware detection, Malware prediction, Word embedding, Chain Sequence	Dynamic analysis
40.	2020	Malware Analysis and Detection using Memory Forensics[40]	Linux and Parrot OS, RAT Trojan Malware (Linux Malware), latest malware, live and dead forensics of memory	Static analysis Dynamic analysis Live Forensics

41.	2021	A Method for Windows Malware Detection Based on Deep Learning[41]	Cybersecurity, Malware detection, Malware image, Convolution of a neural network	Hybrid Visualisation
42.	2021	Fuzzy-import hashing: A static analysis technique for malware Detection[42]	Malware analysis, Fuzzy-import hashing, Fuzzy hashing, Import hashing, YARA Rules, Fuzzy C-Means clustering Ransomware	YARA rules SSDEEP hashing
43.	2021	Malware Analysis by Combining Multiple Detectors and Observation Windows[43]	Malware detection, malware evasion, feature extraction, diversity of detection algorithms, API sequence, memory dump, correlation, deep learning, ensemble detector, malware activation, training time, observation windows,mean-time-to-detect	combined-ensemble detectors Alpha count ensemble-detectors
44.	2021	Graph-Based Malware Detection Using Opcode Sequences[44]	malware detection, static analysis, opcode analysis, graph-based detection, packed malware	Disassembler tools(Distorm3) Static analysis tools
45.	2021	Malware detection on Windows audit logs using LSTMs[45]	Malware, LSTM, Embeddings, Windows audit logs	LSTM

46.	2021	Malware Makeover: Breaking ML-based Static Analysis by Modifying Executable Bytes[46]	adversarial machine learning, malware, neural networks, security	DNNs Machine Learning Algorithm
47.	2021	Sisyfos: A Modular and Extendable Open Malware Analysis Platform[47]	Malware detection, malware evasion, feature extraction, diversity of detection algorithms, API sequence, memory dump, correlation, deep learning, ensemble detector, malware activation, training time, observation windows, mean-time-to-detect.	ANY.RUN Sandbox Hybrid analysis YARA rules
48.	2021	VIRUS-MNIST: A BENCHMARK MALWARE DATASET[48]	Neural Networks, Computer Vision, Image Classification, Malware Detection, MNIST Benchmark	--
49.	2021	Combining Static and Dynamic Analysis to Improve Machine Learning-based Malware Classification [49]	Static features, dynamic features, sandbox, Machine learning, Deep learning	Random Forest Machine Learning(CNN for feature extraction)
50.	2021	A Methodological Study on Malware Analysis[50]	Malware, static analysis, dynamic analysis	--

51.	2022	PROUD-MAL: a static analysis-based progressive framework for deep unsupervised malware classification of Windows portable executable[51]	PROUD-MAL framework, neural networks, Feature Attention-based Neural Network (FANN)	LordPE Ollydbg Static analysis tools
52.	2022	Memory Forensics-Based Malware Detection Using Computer Vision and Machine Learning[52]	malware detection, machine learning, security, static analysis, dynamic analysis, memory forensics, computer vision	x64 debugger Pestudio ProcDt Process Hacker XGBoost Decision Trees
53.	2022	Investigation of Possibilities to Detect Malware Using Existing Tools[53]	Malware Analysis, Malware Detection, Static and Dynamic Analysis Tools, Malware Accuracy and Detection Rate	static analysis tools (PEiD, MD5deep, and BinText) and dynamic analysis tools (Process monitor, Regshot, and Wireshark).
54.	2022	MLMD—A Malware-Detecting Antivirus Tool Based on the XGBoost Machine Learning Algorithm[54]	malware, classification, static analysis, dynamic analysis, supervised machine learning, cybersecurity	MLMD(Machine Learning Malware Detector)

55.	2022	Malware Analysis in IoT & Android Systems with Defensive Mechanism[55]	IoT, android system, malware, kernel-based attack, application attack, application hardening technique	NLP LSTM TinyDroid DroidMoss (For Android detection)
56.	2022	Features Engineering for Malware Family Classification Based API Call[56]	Malware classification, Jaccard similarity, API call sequence	Static analysis(tools) Dynamic analysis(tools) API calls
57.	2022	Identification of malware families using stacking of textural features and machine learning[57]	API calls, Image conversion, classification algorithm	KNN classifier Machine Learning algoritm(SVM, RS, NB)
58.	2022	Fileless malware threats: Recent advances, analysis approach through memory forensics and research challenges[58]	Windows PowerShell, Dynamic analysis, memory-based approach	Static analysis Dynamic analysis (Tools)
59.	2022	HAMLET: Hunt Malware Using Wavelet Transform on Cross-Platform[59]	Wavelet transforms, Ground penetrating radar, Linux, Cyberspace, Geophysical measurement techniques, Machine learning, Drives	Machine learning control flow graph, natural language processing, graph neural networks

60.	2022	Tree-Based Classifier Ensembles for PE Malware Analysis: A Performance Revisit[60]	Portable executable malware, tree-based ensemble, performance comparison, statistical significance test	Tree-based ensemble (random forest, XGBoost, CatBoost, GBM, LightGBM)

Table 3.2 Tools Used for Static Analysis of Malware

Sr. No.	Tool's name	Functionality of the tools
	Exeinfo PE	Retrieves the information from the Windows PE headers. The file signature also determines whether the executable has been packed and indicates how to unpack it.
	HashMyFiles / HashCalc	Generate various hashes such as MD5, SHA-1, SHA-256, RIPEMD 560, CRC32, TIGER, SHA-256, PANAMA etc.
3.	Strings	PowerShell or cmd strings can extract all the strings in ASCII and UNICODE.
4.	UPC	UPX tool can pack and unpack an exe file using CFF Explorer or Pestudio. One can identify if the malware is packed with UPX or not.
5.	Pestudio	Comprehensive amount of information extraction tools, file type, arch, PE headers, strings, hashes
6.	DIE	Detect It Easy is a packer identifier to aid in the definition of file type.
7.	Resource Hacker	It functions as a resource compiler and decompiler, allowing for the viewing and modifying of resources in executables and compiled resource libraries.
8.	Wireshark	Detailed packet examination of numerous protocols at various layers is possible.
9.	PEiD	An application that is used to find such malware that is packed or encrypted
10.	PEview	Provides details on portable executable (PE) file headers and their sections
11.	PE E	A tool to display the PE's content and organizational structure. It can also be used as a file unpacker for packed files.
10.	CFF Explorer	It was developed to make PE editing as simple as feasible while maintaining awareness of the internal organization of the portable executable.
11.	Yara rules	Records malware issue categories based only on patterns
12.	Dependency walker	Detects missing files, invalid files, mismatched CPU types of modules, circular dependency error
13.	HxD Hex Editor	Designed to display both the ASCII interpretation and the file's raw hexadecimal format
14.	BinText	A tool that can search through and display character strings in a binary file.

15.	IDA Pro	The professional interactive disassembler is popular among malware analysts, reverse engineers, and vulnerability analysts.
16.	Hex-Rays Decompiler	Plug-in for IDA Pro that transforms assembly code into pseudocode that resembles C and is readable by humans.
17.	Cuckoo Sandbox	Utilized to execute and analyze files automatically, gather thorough analysis data, and show what the virus performs inside an isolated operating system.
18.	Antivirus scanners	It detects viruses and then stops and gets rid of them.
19.	URLLIB	Used module in Python to fetch URL
20.	Noriben sandbox	For extracting API calls that have been snooped on while an executable file is running, use the NORIBEN sandboxing tool.
21.	Ubuntu	Ubuntu is a Debian-based Linux distribution. It is appropriate for Internet of Things (IoT) gadgets, servers, workstations, and cloud computing.
22.	Volatility tool	A command-line program, Volatility, enables DFIR teams to collect and examine volatile data momentarily held in random access memory (RAM). Such information frequently provides investigators with crucial hints.
23.	BRIDESMAID	A comprehensive and accurate framework for on-device analysis of Android applications that uses a combination of static and dynamic approaches to distinguish between legal and malicious Android apps.
24.	Paragraph vector	One of these techniques, called Paragraph Vector, expands the word2vec process by treating the paragraph as an extra word.
25.	RNN (Recurrent Neural Network)	Recurrent neural networks are a form of artificial neural network frequently utilized in speech recognition and natural language processing.
26.	DroidSieve	Static analysis-based malware classifier for Android that is quick, precise, and resistant to obfuscation
27.	KNN Algorithm	A non-parametric, supervised learning classifier called KNN or k-NN employs closeness to classify or anticipate how a particular data point will be grouped.
28.	Naïve Bayes	A classification algorithm that works well for binary and multiclass classification is labeled Naive Bayes.
29.	Hybrid Analysis	Hybrid Analysis is a file analysis technique that combines runtime data with memory dump analysis to uncover every potential malware execution channel.
30.	Machine Learning Classifiers	An algorithm known as a classifier in machine learning automatically arranges or categorizes data into one or more of a set of classes.
31.	Deep Neural Networks	Models based on deep neural networks (DNNs) can overcome these drawbacks of matrix factorization. Due to the network's input layer's versatility, DNNs may readily add query features and item characteristics, which can assist in identifying a user's interests and increase the relevancy of suggestions.
32.	Smashing	SimHash is a method for quickly determining the degree of similarity between two collections.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

33.	Distort	Since diStorm3 is a decomposer rather than a static text generator, it accepts an instruction and produces a binary structure that explains it.
34.	Radare2	A framework for static and dynamic analysis that uses reverse engineering
35.	AIMED	A genetic programming (GP) method is used in Automatic Intelligent Malware Alterations to Evade Detection to automatically discover modifications that, when injected into malware that has already been detected, can cause malware scanners to misclassify the object.
36.	OllyDbg	When source code is unavailable, OllyDbg, an x86 debugger, can be used to analyze binary code.
37.	LordPE	LordPE is a tool for system programmers and reverse engineers to modify and inspect various PE (Portable Executable) file components and dump such files from memory.
38.	Hybrid Visualisation	Dynamic and static techniques are combined in hybrid visualization.
39.	ANY.RUN	wherein a user may upload a suspicious sample file and get a comprehensive analysis with data gathered from both static and dynamic sources analysis
40.	RegShot	When making system modifications or installing new software, a registry snapshot can be taken and compared to a second one using the registry comparison program.
41.	Process Hacker	Allows the viewing of currently active processes on a device, locates applications that consume CPU power, and locates network connections connected to processes.
42.	ProcDOT	A handy tool for visualizing system operations is called ProcDOT, a malware expert who seeks to understand the behavior of harmful software.
43.	Process Monitor	Process Monitor is a sophisticated Windows monitoring program that displays process/thread activity, file system activity, and registry activity in real time.
44.	MLMD	The underlying principles of the records are found and formalized using a machine studying machine. With these facts, the set of rules can also "assume" approximately the characteristics of samples it has in no way visible earlier than. A previously unknown pattern is probably a brand-new document in malware detection.

IV. DISCUSSION

Security experts can look at numerous malware samples to the speed and performance of static analysis. Static evaluation is a helpful technique for preventing malware infections because it can be used to locate malicious code in software even before it is released. However, static analysis has its boundaries. For instance, the virus cannot provide comprehensive records on its conduct because its whole pastime can best be visible when executed. The static analysis effects may also be misguided if the contamination is designed to evade the exam.

In conclusion, malware static evaluation is a precious approach for figuring out the damage that malware poses and creating defenses against it. However, it ought to be used in conjunction with other analytical techniques, together with dynamic evaluation, to offer a complete image of the behavior and effects of the virus. Because it allows for fast and green analysis, early malware detection, and beneficial insight into the conduct and outcomes of malware, static malware evaluation is a crucial device in cyber security.

Static malware evaluation is one powerful technique that individuals and agencies can use to manage malware. Static evaluation offers danger detection, more robust safety, early warning, price savings, and multiplied productivity, which could help lessen the chance of malware infections and guard vital statistics and structure.

V. PROSPECTS AND TRENDS FOR THE FUTURE

Evasive malware detection remains a hassle despite the advances in research methodologies for sincere malware detection and class models. Diverse execution contexts have been employed to pick out evasion behaviors, famous evasion strategies, and the advent of API-based evasive malware signatures, all of which have been used to locate evasive malware. To our information, every approach for detecting evasive malware has drawbacks. For example, it's challenging to differentiate between evasion techniques used in malevolent activity and people utilized in legal action. In addition, they're detecting any new malware that AI-primarily based software programs haven't picked up on remains hard, which might cause a zero-day assault. Therefore, fast-update studying tactics are required to allow the created models to learn new behaviors adaptably. Deep getting-to-know techniques can be used with unsupervised device mastering strategies to replace getting-to-know and construct models that can adaptively study new dangerous behaviors.

VI. CONCLUSION

Combining safe computer usage, anti-malware software, and careful and responsible online behavior is the most effective defense against malware. Users and systems may protect themselves from the Internet's many threats by being cautious while surfing, avoiding strange links, refusing to accept emails from senders they don't recognize, and frequently updating and running anti-malware software. By reading this essay, anybody may take the first steps toward learning malware analysis. Many recommendations for tools and detecting techniques have been addressed to get more insights. Everybody may start with different research modalities using our review as a reference more efficiently. Technologies for detecting malware must adapt continuously to the ever-evolving nature of malicious software. Artificial intelligence and machine learning developments help solve the malware detection issue as malware writers become more skilled at obfuscating their files and creating scripts that can infiltrate any system and circumvent security measures already in place in many organizational sectors. This article offers several surveillance detection techniques and strategies to exploit how malware may be utilized.

REFERENCES

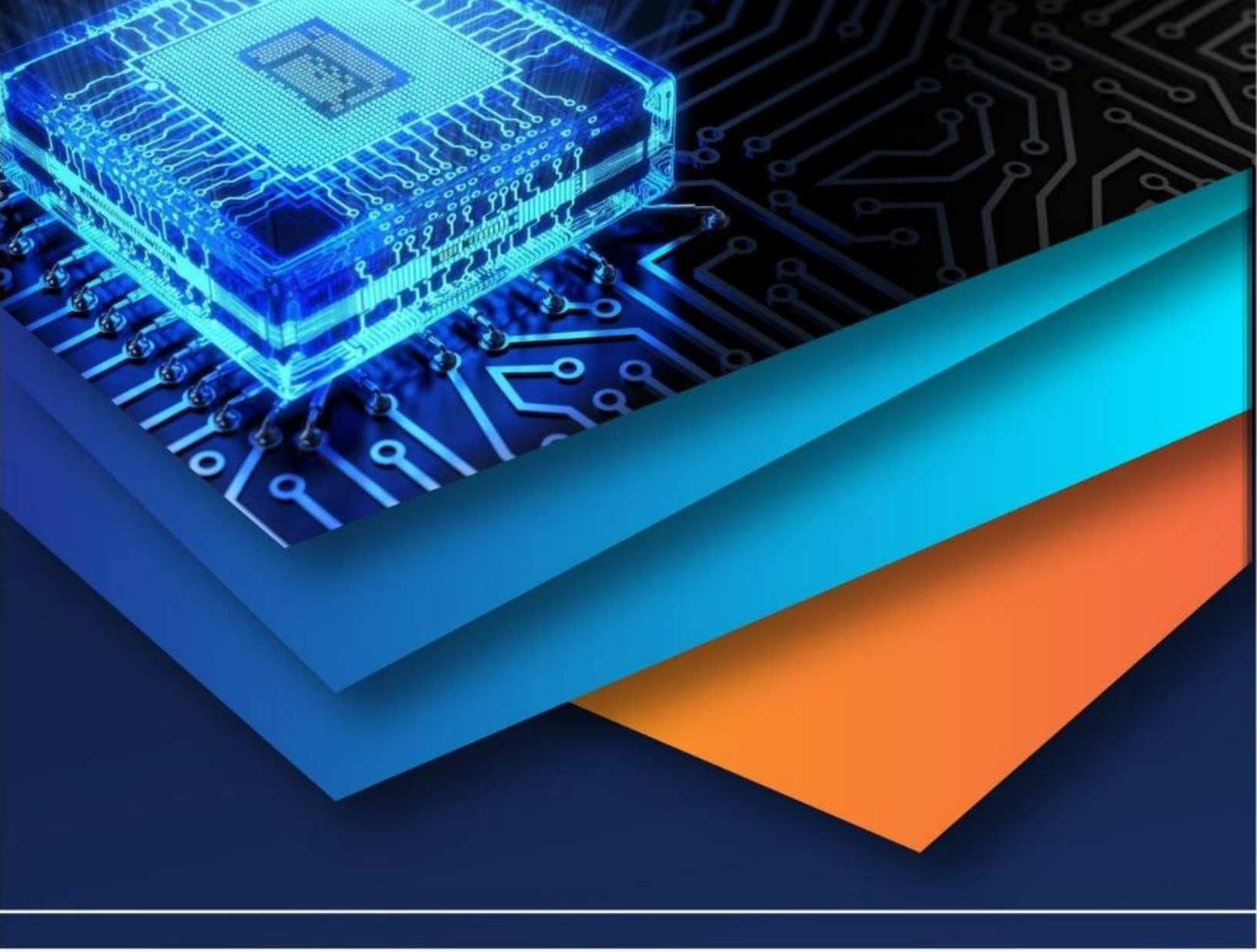
[1] Aslan Ö, … RSI 14th IC on, 2017 undefined. Investigation of possibilities to detect malware using existing tools. ieeexplore.ieee.org. Accessed December 23, 2022. https://ieeexplore.ieee.org/abstract/document/8308437/
[2] Zatloukal F, Znoj J. Malware Detection Based on Multiple PE Headers Identification and Optimization for Specific Types of Files. J Adv Eng Comput. 2017;1(2):153-161. doi:10.25073/JAEC.201712.64
[3] Aslan Ö, Onyedi B, Üniversitesi E. Performance Comparison of Static Malware Analysis Tools Versus Antivirus Scanners To Detect Malware. Published online 2017:25-26. Accessed December 24, 2022. https://www.researchgate.net/publication/321759536
[4] An Efficient Approach for Advanced Malware Analysis Using Memory Forensic Technique | IEEE Conference Publication | IEEE Xplore. Accessed December 24, 2022. https://ieeexplore.ieee.org/abstract/document/8029568
[5] Jain A, Singh AK. Integrated Malware analysis using machine learning. 2nd Int Conf Telecommun Networks, TEL-NET 2017. 2018;2018-January:1-8. doi:10.1109/TEL-NET.2017.8343554
[6] Martinelli F, Mercaldo F, Saracino A. BRIDEMAID: A hybrid tool for accurately detecting android malware. ASIA CCS 2017 - Proc 2017 ACM Asia Conf Comput Commun Secur. Published online April 2, 2017:899-901. doi:10.1145/3052973.3055156
[7] Nagano Y, Uda R. Static analysis with paragraph vector for malware detection. Proc 11th Int Conf Ubiquitous Inf Manag Commun IMCOM 2017. Published online January 5, 2017. doi:10.1145/3022227.3022306
[8] Kwon I, I EG. Extracting the representative API call patterns of malware families using recurrent neural networks. Proc 2017 Res Adapt Converg Syst RACS 2017. 2017;2017-January:202-207. doi:10.1145/3129676.3129712
[9] Wei F, Li Y, Roy S, Ou X, Zhou W. Deep ground truth analysis of current android malware. Lect Notes Comput Sci (including Subser Lect Notes Artif Intell Lect Notes Bioinformatics). 2017;10327 LNCS:252-276. doi:10.1007/978-3-319-60876-1_12/COVER
[10] Suarez-Tangil G, Dash SK, Ahmadi M, Kinder J, Giacinto G, Cavallaro L. DroidSieve: Fast and accurate classification of obfuscated android malware. CODASPY 2017 - Proc 7th ACM Conf Data Appl Secur Priv. Published online March 22, 2017:309-320. doi:10.1145/3029806.3029825
[11] Şahin DÖ, Kural OE, Akleylek S, Kiliç E. New results on permission-based static analysis for Android malware. 6th Int Symp Digit Forensic Secur ISDFS 2018 - Proceeding. 2018;2018-January:1-4. doi:10.1109/ISDFS.2018.8355377
[12] Management RTIJ of E and, 2018, undefined. A study on malware and malware detection techniques. mecs-press.net. Accessed December 24, 2022. http://www.mecs-press.net/ijeme/ijeme-v8-n2/IJEME-V8-N2-3.pdf
[13] Shohel Rana M, Gudla C, Sung AH. Evaluating machine learning models for android malware detection - A comparison study. ACM Int Conf Proceeding Ser. Published online December 14, 2018:17-21. doi:10.1145/3301326.3301390
[14] Shalaginov A, Banin S, Dehghantanha A, Franke K. Machine learning aided static malware analysis: A survey and tutorial. Adv Inf Secur. 2018;70:7-45. doi:10.1007/978-3-319-73951-9_2/COVER
[15] Poudyal S, Subedi KP, Dasgupta D. A Framework for Analyzing Ransomware using Machine Learning. Proc 2018 IEEE Symp Ser Comput Intell SSCI 2018. Published online January 28, 2019:1692-1699. doi:10.1109/SSCI.2018.8628743
[16] Stokes JW, Wang D, Marinescu M, Marino M, Bussone B. Attack and Defense of Dynamic Analysis-Based, Adversarial Neural Malware Detection Models. Proc - IEEE Mil Commun Conf MILCOM. 2019;2019-October:102-109. doi:10.1109/MILCOM.2018.8599855

[17] Firdaus A, Anuar NB, Karim A, Razak MFA. Discovering optimal features using static analysis and a genetic search-based method for Android malware detection. Front Inf Technol Electron Eng. 2018;19(6):712-736. doi:10.1631/FITEE.1601491/METRICS
[18] Kozachok A V., Kozachok VI. Construction and evaluation of the new heuristic malware detection mechanism based on executable files static analysis. J Comput Virol Hacking Tech. 2018;14(3):225-231. doi:10.1007/S11416-017-0309-3/METRICS
[19] Kumar R, Xiaosong Z, Ullah Khan R, Kumar J, Ahad I. Effective and explainable detection of android malware based on machine learning algorithms. ACM Int Conf Proceeding Ser. Published online March 12, 2018:35-40. doi:10.1145/3194452.3194465
[20] Yousefi-Azar M, Hamey LGC, Varadharajan V, Chen S. Malytics: A malware detection scheme. IEEE Access. 2018;6:49418-49431. doi:10.1109/ACCESS.2018.2864871
[21] Irshad A, Maurya R, Dutta MK, Burget R, Uher V. Feature optimization for run time analysis of malware in Windows operating system using machine learning approach. 2019 42nd Int Conf Telecommun Signal Process TSP 2019. Published online July 1, 2019:255-260. doi:10.1109/TSP.2019.8768808
[22] Vidyarthi D, Kumar C, Rakshit S, Chansarkar S. Static Malware Analysis to Identify Ransomware Properties. doi:10.5281/zenodo.3252963
[23] Kabakus AT. What Static Analysis Can Utmost Offer for Android Malware Detection? Inf Technol Control. 2019;48(2):235-249. doi:10.5755/J01.ITC.48.2.21457
[24] Khan RU, Zhang X, Kumar R. Analysis of ResNet and GoogleNet models for malware detection. J Comput Virol Hacking Tech. 2019;15(1):29-37. doi:10.1007/S11416-018-0324-Z/METRICS
[25] Naz S, Singh DK. Review of Machine Learning Methods for Windows Malware Detection. 2019 10th Int Conf Comput Commun Netw Technol ICCCNT 2019. Published online July 1, 2019. doi:10.1109/ICCCNT45670.2019.8944796
[26] Sibi Chakkaravarthy S, Sangeetha D, Vaidehi V. A Survey on malware analysis and mitigation techniques. Comput Sci Rev. 2019;32:1-23. doi:10.1016/J.COSREV.2019.01.002
[27] Ugarte-Pedrero X, Graziano M, Balzarotti D. A Close Look at a Daily Dataset of Malware Samples. ACM Trans Priv Secur. 2019;22(1). doi:10.1145/3291061
[28] Kumar N, Mukhopadhyay S, Gupta M, Handa A, Shukla SK. Malware classification using early-stage behavioral analysis. Proc - 2019 14th Asia Jt Conf Inf Secur AsiaJCIS 2019. Published online August 1, 2019:16-23. doi:10.1109/ASIAJCIS.2019.00-10
[29] Nar M, Kakisim AG, Yavuz MN, Sogukpinar I. Analysis and Comparison of Disassemblers for OpCode Based Malware Analysis. UBMK 2019 - Proceedings, 4th Int Conf Comput Sci Eng. They were published online September 1, 2019:17-22. doi:10.1109/UBMK.2019.8907153
[30] Castro RL, Schmitt C, Dreo G. AIMED: Evolving malware with genetic programming to evade detection. Proc - 2019 18th IEEE Int Conf Trust Secur Priv Comput Commun IEEE Int Conf Big Data Sci Eng Trust 2019. Published online on August 1, 2019:240-247. doi:10.1109/TRUSTCOM/BIGDATASE.2019.00040
[31] Hwang C, Hwang J, Kwak J, Lee T. Platform-Independent Malware Analysis Applicable to Windows and Linux Environments. Electron 2020, Vol 9, Page 793. 2020;9(5):793. doi:10.3390/ELECTRONICS9050793
[32] Rabadi D, Teo SG. Advanced Windows Methods on Malware Detection and Classification. ACM Int Conf Proceeding Ser. Published online December 7, 2020:54-68. doi:10.1145/3427228.3427242
[33] Darabian H, Homayounoot S, Dehghantanha A, et al. Detecting Cryptomining Malware: a Deep Learning Approach for Static and Dynamic Analysis. J Grid Comput. 2020;18(2):293-303. doi:10.1007/S10723-020-09510-6/METRICS
[34] A Survey on Malware Detection and Analysis Tools by Sajedul Talukder, Zahidur Talukder : SSRN. Accessed December 25, 2022. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3901568
[35] Mantoo BA, Khurana SS. Static, dynamic, and intrinsic features-based Android malware detection using machine learning. Lect Notes Electr Eng. 2020;597:31-45. doi:10.1007/978-3-030-29407-6_4/COVER
[36] Catak FO, Yazi AF, Elezaj O, Ahmed J. Deep learning based Sequential model for malware analysis using Windows exe API Calls. PeerJ Comput Sci. 2020;6:e285. doi:10.7717/PEERJ-CS.285
[37] Nassiri M, HaddadPajouh H, Dehghantanha A, Karimipour H, Parizi RM, Srivastava G. Malware elimination impact on dynamic analysis: An experimental machine learning approach. Handb Big Data Priv. Published online March 18, 2020:359-370. doi:10.1007/978-3-030-38557-6_17/COVER
[38] Fang Y, Zeng Y, Li B, Liu L, Zhang L. DeepDetectNet vs RLAttackNet: An adversarial method to improve deep learning-based static malware detection model. PLoS One. 2020;15(4):e0231626. doi:10.1371/JOURNAL.PONE.0231626
[39] Amer E, Zelinka I. A dynamic Windows malware detection and prediction method based on contextual understanding of API call sequence. Comput Secur. 2020;92:101760. doi:10.1016/J.COSE.2020.101760
[40] Malware Analysis Using Memory Forensics. Accessed December 25, 2022. https://www.secjuice.com/malware-analysis-memory-forensics/
[41] Huang X, Ma L, Yang W, Zhong Y. A Method for Windows Malware Detection Based on Deep Learning. J Signal Process Syst. 2021;93(2-3):265-273. doi:10.1007/S11265-020-01588-1/METRICS
[42] Naik N, Jenkins P, Savage N, Yang L, Boongoen T, Iam-On N. Fuzzy-import hashing: A static analysis technique for malware detection. Forensic Sci Int Digit Investig. 2021;37:301139. doi:10.1016/J.FSIDI.2021.301139
[43] Ficco M. Malware Analysis by Combining Multiple Detectors and Observation Windows. IEEE Trans Comput. 2022;71(6):1276-1290. doi:10.1109/TC.2021.3082002
[44] Gulmez S, Sogukpinar I. Graph-Based Malware Detection Using Opcode Sequences. 9th Int Symp Digit Forensics Secur ISDFS 2021. Published online June 28, 2021. doi:10.1109/ISDFS52919.2021.9486386
[45] Ring M, Schlör D, Wunderlich S, Landes D, Hotho A. Malware detection on Windows audit logs using LSTMs. Comput Secur. 2021;109:102389. doi:10.1016/J.COSE.2021.102389
[46] Lucas K, Sharif M, Bauer L, Reiter MK, Shintre S. Malware Makeover: Breaking ML-based Static Analysis by Modifying Executable Bytes. ASIA CCS 2021 - Proc 2021 ACM Asia Conf Comput Commun Secur. Published online May 24, 2021:744-758. doi:10.1145/3433210.3453086
[47] Serpanos D, Michalopoulos P, Xenos G, Ieronymakis V. Sisyfos: A Modular and Extendable Open Malware Analysis Platform. Appl Sci 2021, Vol 11, Page 2980. 2021;11(7):2980. doi:10.3390/APP11072980
[48] Noever D, Noever SEM. Virus-MNIST: A Benchmark Malware Dataset. Published online February 28, 2021. doi:10.48550/arxiv.2103.00602

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

[49] Chanajitt R, Pfahringer B, Gomes HM. I am combining Static and Dynamic Analysis to Improve Machine Learning-based Malware Classification—2021 IEEE 8th Int Conf Data Sci Adv Anal DSAA 2021. I published it online in 2021. doi:10.1109/DSAA53316.2021.9564144
[50] Panwala HR. A Methodological Study on Malware Analysis. academia.edu. 2021;9. doi:10.22214/ijraset.2021.38416
[51] Rizvi SKJ, Aslam W, Shahzad M, Saleem S, Fraz MM. PROUD-MAL: a static analysis-based progressive framework for deep unsupervised Windows portable executable malware classification. Complex Intell Syst. 2022;8(1):673-685. doi:10.1007/S40747-021-00560-1/FIGURES/8
[52] Shah SSH, Ahmad AR, Jamil N, Khan A your R. Memory Forensics-Based Malware Detection Using Computer Vision and Machine Learning. Electron 2022, Vol 11, Page 2579. 2022;11(16):2579. doi:10.3390/ELECTRONICS11162579
[53] Aslan O, Samet R. Investigation of possibilities to detect malware using existing tools. Proc IEEE/ACS Int Conf Comput Syst Appl AICCSA. 2018;2017-October:1277-1284. doi:10.1109/AICCSA.2017.24
[54] Palša J, Ádám N, Hurtuk J, et al. MLMD— A Malware-Detecting Antivirus Tool Based on the XGBoost Machine Learning Algorithm. Appl Sci 2022, Vol 12, Page 6672. 2022;12(13):6672. doi:10.3390/APP12136672
[55] Yadav CS, Singh J, Yadav A, et al. Malware Analysis in IoT & Android Systems with Defensive Mechanism. Electron 2022, Vol 11, Page 2354. 2022;11(15):2354. doi:10.3390/ELECTRONICS11152354
[56] Daeef AY, Al-Naji A, Chahl J. Features Engineering for Malware Family Classification Based API Call. Comput 2022, Vol 11, Page 160. 2022;11(11):160. doi:10.3390/COMPUTERS11110160
[57] Kumar S, Janet B, Neelakantan S. Identification of malware families using stacking of textural features and machine learning. Expert Syst Appl. 2022;208:118073. doi:10.1016/J.ESWA.2022.118073
[58] Kara I. Fileless malware threats: Recent advances, analysis approach through memory forensics and research challenges. Expert Syst Appl. 2023;214:119133. doi:10.1016/J.ESWA.2022.119133
[59] Park S, Jeon S, Kim HK. HMLET: Hunt Malware using Wavelet Transform on Cross-Platform. IEEE Access. Published online 2022. doi:10.1109/ACCESS.2022.3225223
[60] Bergadano F, Giacinto G, Hilda M, Louk L, Adhi Tama B. Tree-Based Classifier Ensembles for PE Malware Analysis: A Performance Revisit. Algorithms 2022, Vol 15, Page 332. 2022;15(9):332. doi:10.3390/A15090332

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)