ISSN: 2522-3429 (Print)
ISSN: 2616-6003 (Online)

International Journal for Electronic Crime Investigation

(IJECI)

VOLUME: 8
ISSUE: 1 Jan-Mar 2024

Email ID: ijeci@lgu.edu.pk

Digital Forensics Research and Service Center
Lahore Garrison University, Lahore, Pakistan.

International Journal for Electronic Crime Investigation

Volume 8(1) Jan-Mar 2024

SCOPE OF THE JOURNAL

The International Journal for Crime Investigation IJECI is an innovative forum for researchers, scientists and engineers in all the domains of computer science, white Collar Crimes, Digital Forensics, Nano Forensics, Toxicology and related technology, Criminology, Criminal Justice and Criminal Behaviour Analysis. Moreover, the scope of the journal includes algorithm, high performance, Criminal Data Communication and Networks, pattern recognition, image processing, artificial intelligence, VHDL along with emerging domains like quantum computing, IoT, Hacking. The journal aims to provide an academic medium for emerging research trends in the general domain of crime investigation.

SUBMISSION OF ARTICLES

We invite articles with high quality research for publication in all areas of engineering, science and technology. All the manuscripts submitted for publication are first peer reviewed to make sure they are original, relevant and readable. Manuscripts should be submitted via email only.

To submit manuscripts by email with attach file is strongly encouraged, provided that the text, tables, and figures are included in a single Microsoft Word/Pdf file.

Contact: For all inquiries, regarding call for papers, submission of research articles and correspondence,kindly contact at this address:

IJECI, Sector C, DHA Phase-VI Lahore, Pakistan
Phone: +92- 042-37181823
Email: ijeci@lgu.edu.pk

International Journal for Electronic Crime Investigation

Volume 8(1) Jan-Mar 2024

CONTENTS

International Journal for Electronic Crime Investigation

Volume 8(1) Jan-Mar 2024

Patron in Chief: Maj Gen Muhammad Khalil Dar, HI(M) (Retd)
Vice Chancellor Lahore Garrison University

Advisory Board

Mr. Kaukab Jamal Zuberi, HOD Department of Criminology and Forensic Sciences, Lahore Garrison University, Lahore.
Dr. Abeo Timothy Apasiba, Temale Technical University, Central African Republic.
Dr. Atta-ur-Rahman. Imam Abdulrahman Bin Faisal University (IAU), Saudi Arabia.
Dr. Natash Ali Mian. Beaconhouse National University, Lahore.
Prof. Dr. M. Pervaiz Khurshid, Govt College Science, Lahore.
Dr. Nadeem Abbas, Linnaeus University, Sweden

Editorial Board

Mr. Kaukab Jamal Zuberi, HOD Department of Criminology and Forensic Sciences, Lahore Garrison University, Lahore.
Dr. Badria Sulaiman Alfurhood, Princess Nourah bint Abdulrahman University,Riyadh, Saudi Arabia.
Dr. Muhammad Adnan Khan, Gachon University, Seongnam, Republic of Korea.
Dr. Faheem Khan, Gachon University, Seongnam, Republic of Korea.
Prof. Dr. Peter John, GC University, Lahore
Prof. Dr. Saqib Shehzad, Higher Education Department, Lahore
Dr. Shabbir Hussain, KFUEIT, Rahim Yar Khan.
Dr. Tahir Alyas, ORIC Director, Lahore Garrison University
Dr. Zahida Perveen, Lahore Garrison University.
Dr. Ahmed Naeem, Lahore Garrison University
Dr. Sumaira Mazhar, Lahore Garrison University.
Dr. Roheela Yasmeen, Lahore Garrison University.

Editor in Chief: Dr. Syeda Mona Hassan, Lahore Garrison University.
Associate Editor: Dr. Syed Ejaz Hussain, Lahore Garrison University.
Ms. Fatima, Lahore Garrison University.
Assistant Editors: Mr. Imran Khalid, Lahore Garrison University.
Mr. Qais Abaid, Lahore Garrison University.

Reviewers Committee:

Dr. Qaisar Abbas, Islamic University of Madinah, Madinah, Saudia Arabia.
Dr. Shehzad Ahmad. King Fahd University of Petroleum & Minerals, Saudia Arabia.
Dr. Haroon Ur Rasheed, University of Lahore.
Dr. Munawar Iqbal, University of Education, Lahore.
Engr. Dr. Shahan Yamin Siddiqui. Minhaj University Lahore.
Dr. Saima Naz, University of Education, Lahore.
Dr. Shagufta Saeed, UVAS, Lahore.
Dr. Shazia Saqib, University of Central Punjab, Lahore.
Dr. Mohsin Javed, UMT, Lahore.
Dr. Ayesha Atta, GC University, Lahore.
Dr. Nida Anwar, Virtual University of Pakistan, Pakistan.
Dr. Faisal Rehman, Lahore Leads University, Pakistan.
Dr. Sagheer Abbas, Bahria University, Lahore.
Dr. Asad Mujtaba, Shanghai Jiao Tong University, Shanghai, China.
Dr. Nadia Tabassum, Virtual University of Pakistan, Pakistan.
Dr. Shahid Naseem, UOE, Lahore
Dr. Gulzar Ahmed, Pak Aims Lahore.
Dr. Muhammad Asif, NCBA&E, Lahore
Dr. Waseem Iqbal, Superior University, Lahore.
Dr. Ayesha Ahmad, Govt Collage for women Multan.
Dr. Muhammad Hamid, UVAS, Lahore
Dr. Khawar Bashir, UVAS, Lahore
Dr. Allah Ditta, University of Education, Pakistan.

Editorial

Enhancing Counterterrorism Measures in Pakistan: Addressing Financing Challenges and Strengthening Legal Frameworks

Kaukab Jamal Zuberi

Terrorist organizations continue to pose a significant threat to global security, relying on diverse methods to finance their activities and evade detection. In Pakistan, combating terrorist financing is of paramount importance, yet inherent weaknesses in the legal framework present formidable obstacles for law enforcement agencies and financial regulators. This article delves into the complexities of terrorist financing, the challenges faced by Pakistani authorities, and recommendations for enhancing the Anti-Terrorism Act (ATA) to bolster counterterrorism efforts.

Terrorist financing encompasses a range of activities, including funding sourced from both legal and illicit means. While legitimate businesses, government allocations, and charitable organizations may unknowingly contribute to terrorist financing, illicit sources such as drug trafficking, kidnapping, and corruption remain significant avenues for funding terrorist activities. Moreover, the use of money laundering techniques further complicates efforts to trace and disrupt the flow of funds, necessitating robust regulatory frameworks and investigative techniques to combat this threat effectively.

In Pakistan, the ATA serves as the primary legal instrument for countering terrorism and terrorist financing. However, several inherent weaknesses within the ATA hinder its effectiveness in addressing the evolving nature of terrorist financing activities. One key challenge lies in the broad definition of "terrorism" outlined in the ATA, which can lead to ambiguity and challenges in prosecuting cases related to terrorist financing. Clarifying and refining this definition to encompass specific criteria related to terrorist financing would provide greater clarity for law enforcement agencies and enhance their ability to identify and prosecute individuals involved in financing terrorist activities.

Additionally, the ATA's definition of "terrorist financing" requires further elaboration to address the complexities of modern financial transactions and money laundering techniques employed by terrorist organizations. By providing a more comprehensive definition that encompasses various forms of financial support, including funds sourced from illicit activities and money laundering, the ATA can better equip authorities to detect and disrupt terrorist financing networks effectively.

Furthermore, addressing procedural challenges within the legal system is essential to expedite the investigation and prosecution of terrorist financing cases. Delays in judicial processes and the burden of proof required for successful prosecutions present significant obstacles for law enforcement agencies, allowing perpetrators to evade accountability and continue their illicit activities. Streamlining legal procedures, enhancing cooperation between investigative agencies and the judiciary, and providing adequate resources for timely investigations and prosecutions are essential steps in overcoming these challenges.

Ensuring the safety and protection of witnesses is another critical aspect of enhancing counterterrorism measures in Pakistan. Witness intimidation and lack of adequate protection measures undermine the integrity of investigations and deter individuals from coming forward with crucial information. Implementing comprehensive witness protection programs and strengthening legal provisions to safeguard witnesses' identities and testimonies would encourage greater cooperation with law enforcement agencies and facilitate the prosecution of terrorist financing cases.

Moreover, addressing the partial implementation of Financial Action Task Force (FATF) standards is imperative to strengthen Pakistan's regulatory framework for combating terrorist financing. Close coordination between government institutions, financial intelligence units, and international partners is essential to ensure full compliance with FATF recommendations and enhance Pakistan's ability to combat money laundering and terrorist financing effectively.

Additionally, enhancing oversight of designated non-financial businesses and professions (DNFBPs) and informal financial channels is crucial to closing existing loopholes exploited by terrorist financiers. Strengthening regulatory supervision, conducting regular audits, and implementing stringent reporting requirements for DNFBPs would help mitigate the risks associated with illicit financial transactions and enhance transparency in the financial system.

Furthermore, addressing emerging threats posed by cryptocurrencies and virtual economies requires proactive measures to regulate and monitor these sectors effectively. Developing comprehensive regulatory frameworks, enhancing technological capabilities for monitoring financial transactions, and fostering collaboration with international partners are essential steps in mitigating the risks associated with these emerging technologies.

In conclusion, enhancing counterterrorism measures in Pakistan requires a comprehensive approach that addresses the inherent weaknesses in the legal framework, strengthens regulatory oversight, and enhances investigative capabilities. By refining the Anti-Terrorism Act to provide clearer definitions and procedural guidelines, addressing challenges in the judicial system, and bolstering regulatory compliance with international standards, Pakistan can strengthen its capacity to combat terrorist financing effectively. Moreover, proactive measures to address emerging threats and vulnerabilities in the financial system are essential to safeguarding national security and promoting global stability.

Zohaib *et al. (IJECI)* 2024

International Journal for
Electronic Crime Investigation

DOI: https://doi.org/10.54692/ijeci.2024.0801185

(IJECI)
ISSN: 2522-3429 (Print)
ISSN: 2616-6003 (Online)

Research Article

Vol. 8 issue 1 Jan-Mar 2024

A Comprehensive Study for Malware Detection through Machine Learning in Executable Files

Zohaib Ahmad[1], Ahsan Wajahat[2] and Muhammad Salman Pathan[3]

[1] Faculty of Electronics and Information Engineering, Beijing University of Technology, Beijing, China.

[2] Faculty of information Technology, Beijing University of Technology, Beijing, China.

[3] School of Computer Science, National University of Ireland, Maynooth: IE.

Correspondence Author: ahmedzohaib03@gmail.com

Received: December 18, 2023; **Accepted:** January 07, 2024; **Published:** March 15, 2024

ABSTRACT

Two methods are frequently used to analyze malware and start specimens: static analysis and dynamic analysis. Following analysis, distinct characteristics are retrieved to distinguish malware from benign samples. The detection capacity of malware is contingent upon the effectiveness with which discriminative malware characteristics are retrieved through analysis methods. While conventional approaches and techniques were used inadvertently, machine learning algorithms are now utilized to classify malware, which can deal with the complexity and velocity of malware creation. However, even though a few research papers have been published, recent classifications of signature, behavioral and hybrid machine learning is not introduced well. Based on this demand, we provide a comprehensive analysis of malware detection using machine learning, as well as address the different difficulties associated with building the malware classifier. Finally, future work is addressed to build an effective malware detection system by addressing different malware detection problems.

Keywords: Machine learning, Static analysis, API calls, Ransomware, obfuscation technique malicious software, Dynamic analysis.

1. INTRODUCTION

Despite major improvements in computer security methods and their continual growth, the malware remains the primary threat in cyberspace [1, 2]. Malware investigators analyze malicious samples using techniques from several domains, including program analysis and network analysis, to acquire a better knowledge of their behavior and evolution [3-6]. The first computer malware, known as the brain, was developed in the 1980s and a lot of computers were infected. At that time the speed of malware creation was not very fast because it was the peak of computerization at the time. However, as time passes, a thousand new types of malware are created each day [7]. Due to widespread development of malware nowadays, the most recent malware is significantly more targeted, covert zero-day, and persistent than classic malware, which was open, broad, and performed just once [8-10]. Furthermore, today's malware is quite sophisticated, with the primary goal of exploiting computer system flaws. To get around malware identification and analysis systems, malware authors utilize a variety of obfuscation procedures [11]. Malware authors also used encryption and encoding techniques to create complicated harmful programs such as metamorphic , polymorphic, and packed malware, is extremely difficult to analyze and identify [12-14].

The spreading vectors, which are mentioned in Table 1, are typically used to propagate malware from one system to another. The battle between malware creators and analysts continues. Both sides are creating new methodologies and techniques for malware detection systems concurrently, while the other is building malicious software to breach the detection system to target computer and network resources. The malware researcher analyses knew malware intending to prevent an assault on the computer system [15, 16]. Malware is spotted using one of two approaches: signature-based and behavior-based identification. While signature-based malware recognition methods are quick and effective, obfuscated software easily evades them [17-19]. Behavior-based approaches, on the other hand, outperform obfuscation. The behavior-based approach takes a long time. Not only have methods for detecting malware based on behavior and signatures been developed, but also many hybrid tactics that incorporate the recompenses of both. Hybrid detection methods are intended to overcome the concerns associated with both signature-based and behavior-based methods for detection. Zero-day malware detection is thought-provoking as such malware makes use of the recent susceptibilities that have not yet been discovered [19, 20]. Crackers aim to find vulnerabilities in new software and exploit them to breach the software's security. Since the first malware assault on a computer system, a defense mechanism has been built [21, 22]. Machine learning offers a potential answer to this problem by allowing developers to create malware classifiers that can detect new virus and it’s variant [23, 24]. Various machine learning-based strategies based on supervised and unsupervised algorithms have been suggested in the literature [16, 24].

Two main aspects arise after the evaluation of the proposed machine-based detection methods. For the testing of malware, the first stage is the development of classification algorithms for the classification device and the second step is the extraction of malware using a dynamic and static approach. These two variables affect the precise classification of malware. Both the NB, DT, SVM, and ensemble classification technology RF and Ada boosting have been utilized and enhanced for classification training. Classifying ensembles usually provide better results [25]. The benefits and limitations of each categorization algorithm include. In addition, the representation of the feature greatly changes the detection rate of the classifier. One needs a far more reliable automated malware detection technology. Some academics have created automated cognitive analytic methods for

addressing the extremely disastrous zero-day malware that can also resist malware assaults. Continuous study of malware is crucial to update techniques for detecting new malware patterns and behavior and variants in existing malware.

Malware authors also used encryption and encoding techniques to create complicated harmful programs such as polymorphic, metamorphic, and packed malware, which is extremely difficult to detect and analyze [12-14]. The spreading vectors, which are mentioned in Table 1, are typically used to propagate malware from one computer system to another. The battle between malware creators and analysts continues. Both sides are creating new methodologies and techniques for malware detection systems concurrently, while the other is building malicious software to breach the detection system to target computer and network resources. The malware researcher analyses knew malware intending to prevent an assault on the computer system [15, 16]. Malware is spotted using one of two methods: signature-based detection or behavior-based detection.

While signature-based malware recognition methods are quick and effective, obfuscated software easily evades them [17-19]. Behavior-based approaches, on the other hand, outperform obfuscation. The behavior-based approach takes a long time. Not only have methods for detecting malware based on behavior and signatures been developed, but also many hybrid tactics that incorporate the advantages of both. Hybrid detection methods are intended to overcome the issues associated with both signature-based and behavior-based methods for detection. Zero-day malware detection is thought-provoking because such malware makes use of the recent susceptibilities that have not yet been discovered [19, 20]. Crackers aim to find vulnerabilities in new software and exploit them to breach the software's security. Since the first malware assault on a computer system, a defense mechanism has been built [21, 22]. Machine learning offers a potential answer to this problem by allowing developers to create malware classifiers that can detect new virus and it’s variant [23, 24]. Various machine learning-based strategies based on supervised and unsupervised algorithms have been proposed in the literature [16, 24]. Two main aspects arise after the evaluation of the proposed machine-based detection methods.

For the testing of malware, the first stage is the development of classification algorithms for the classification device and the second step is the extraction of malware using a dynamic and static approach. These two variables affect the precise classification of malware. Both the NB, DT, SVM, and ensemble classification technology RF and Ada boosting have been utilized and enhanced for classification training. Classifying ensembles usually provide better results [25]. The benefits and limitations of each categorization algorithm include. In addition, the representation of the feature greatly changes the detection rate of the classifier. One needs a far more reliable automated malware detection technology. Some academics have created automated cognitive analytic methods for addressing the extremely disastrous zero-day malware that can also resist malware assaults. Continuous study of malware is crucial to update techniques for detecting new malware patterns and behavior and variants in existing malware.

Table 1 : Methods for Spreading

Methods for Spreading	Features
Drive-by Download	Unintentional drive-by downloads point to a malware infection that causes damage to users in a variety of ways. Cybercriminals steal and gather personal information and get all the account credentials via drive-by downloads. For example, their banking information such as usernames and passwords may also include Trojans or exploit kits that may be used to spread other malicious targets.
Vulnerability	A vulnerability is a security hole in a device's or program's software that allows an invader activity to insert malware into the system. It may be a fault arising due to programming in an application or device software, design fault, or some other form of inbuilt flaw. A very successful WannaCry (2017) ransomware exploited Windows 7's vulnerability to encrypt millions of users' files. This malware exploited an existing weakness in the Windows 7 OS SMB. The user who mended the susceptibility did not affect them, while the remaining lost their data. It is a perilous kind of vector of propagation that is very hard to deal with. An invader discovers a weakness in the OS or application and attempts to create malware to exploit the existing defect to do the most harm. As a result, to handle various types of malwares, the user must update the system periodically.
Backdoor	A backdoor denotes any technique that enables allowed and unauthorized operators to utilize regular security mechanisms to increase high-level user access and to a device, network, or software program (aka root-access). Cyber thieves will exploit a loophole when the information is stolen, further software is installed, and the machine is hijacked.
Removable Drives	Nowadays, removable drives are available on both flash discs and hard discs. These are the most prevalent malware distribution techniques for each device. Despite the existence of anti-malware software on the infected computer, it enables infection propagation and connects the infected device to the mobile drive. Users should always take care while transferring data between computers using flash devices. It can transmit any kind of malware, including viruses, worms, and ransomware.
Homogeneity	The setup of similar OS software connected via the same network becomes the source of worm virus spreading from one machine to another.

1.1. Contributions

The current state and evolution of malware detection systems are discussed in this research study.

1. Many classifications technique for machine learning is explored and compared.
2. Recent classifications of signature, behavioral and hybrid machine learning are explored. It shares with its advantages and limitations a fraction of the data in the proposed malware detection systems.
3. The present study has covered some important parameters that influence malware classifier performance. A hybrid model for malware detection has been presented utilizing machine learning is also provided. Finally, the paper was completed, and the topic of the future directive was discussed.

1.2. Scope Overview

The emphasis of this study is based on the detection of malware using ML techniques, and to create the executable files system's automated smart malware detection. This article examines the work suggested for the identification of executing files and provides information on current research on malware revealing via numerous characteristics and methods.

1.3. Evolution

There is a complete review of machine learning malware identification methods. Owing to significant variances in the number of data sets used, ML algorithms, and valuation processes, the detection technique provided are very difficult to compare properly. The results of suggested ML-based malware classificatory are nevertheless equated and presented with certain in this research.

1.4. Orgnization

The rest of the paper is systematized as follows. Section 2 introduces systems for malware research. Section 3 debates the machine learning methods employed to categorize malware. In Section 4 static, dynamic, and hybrid, the analysis of different approaches is provided. Section 5 assesses the result of the examined papers, and the many criteria for malware classification are explained in section finally it illustrates the possible breadth and concludes the study.

2. MALWARE ANALYSIS

Different malware data samples are investigated to obtain results that can be utilized to detect them. Static and dynamic studies are two basic malware research methodologies that are illustrated below.

2.1. Static Malware Analysis

In the study, features are gathered without running the malware sample. Following the analysis by the extraction of several static features such as N-grams, hash value, strings, opcodes, and PE header information. The development of malware revealing software (antiviruses, intrusion detection systems etc. is based on these characteristics [26, 27]. Security analysts examine malware samples either by using reverse engineering or not. The malware files are dismantled and converted into assembly language code, which is then used to test the malware sample during the coding stage. Some of the most widely used IDA Pro, Ollydbg, WinDbg, and capstone disassemblers are among the most widely used disassemblers and debuggers [33, 34]. An examination of the assembly code is performed to discover the processing route of a malicious operation file, pattern and structure. This information may be used to detect new or variant malware. To study the assembly language code to identify the execution functions is a time-consuming procedure. The use of code obfuscation practices makes the forecaster's work even more difficult. Malware authors employ a variety of methods to escape malware inspection, including code encryption, code reordering instructions, and dead code insert techniques [28, 29]. Static analysis techniques are defined briefly in Table 2, which is followed by a discussion of the methodologies.

Table 2: Static analysis tools

Tool name	Description
PeView [44]	The 32-bit Portable Execution File (PE) structure and content of files, as well as the Component Object File Format, may be viewed quickly and simply with this tool (COFF). This PE/COFF file reader reads headers, sections, directories, import tables, and tables for exporting, and resource data numerous files (such as EXE, OBJ, DLL, DBG and LIB) as well as information for resource purposes.
PEid [45]	Used to test if the malware is hideous if the packer tool is used (for example NSPACK, UPX, etc.). The creators of malware are increasingly utilizing antivirus methods to disguise the true malicious code. The packaged malware may be classified using PEid.

CFF Explorer [46]	This provides the executable file with complete header information and metadata. In this application, you may view more detailed information about executable files.
PsFile [47]	This useful tool shows details about the system's open files. Analyzing if the computer device is remotely controlled is very beneficial. It can send details about opened files to the remote computer on the local machine.
Accesschk [48]	To evaluate the degree of security of the equipment, Accesschk is utilized. It contains information on access rights, including whether a group or user can write, read, or accomplish, among other things, registry keys and files.
IOC Finder [49]	(IOC) Finder is a free utility that collects host system data and reports IOC presence. IOCs are open-standard XML documents that assist incident responders in capturing a variety of threat information.
Radare [50]	Radare is a comprehensive set of tools for reverse engineering. This utility is available on a wide range of platforms including Windows, Linux, Android, and MacOS. On the file system, Radare can also do forensics.
Yara [51]	The Yara tool is employed in executable files to match a string. To recognize the malware file, such string signatures may be used in malware analysis. The Yara tool has the capability of matching a certain string pattern contained inside a binary file. Other file formats, such as PDFs, Word documents, and other similar documents may be matched using this feature.
SS Deep [52]	Executed fuzzy hash values needed to verify malware variants may be calculated in this utility. The fuzzy hash contains more potential to compact with malware variations, unlike the simple hash value.
Disassemblers [44]	Disassemblers are then employed if a more comprehensive static analysis is needed. IDA Pro is a well-known and widely used disassembler. To effectively carry out the reverse engineering task, a new Ghidra disassembler was constructed. The assembly code that is transformed from the executable file, which is then examined manually to determine the functionality of the malicious software.

2.2. Dynamic Malware Analysis

When using this procedure for investigation malware the files of malware are processed and the resulting malware running time behavior is captured and analyzed. There are many types of runtime behavior including file system, processing execution, modification to registry key, and network activity [30-33]. Dynamic inspection differs from static examination since it relies on the connection amid malware and the windows operating system. To ensure system security, the execution of malware samples is always carried out in a simulated environment, since if the malware program's file to be run direct on the host machine, it would cause damage to the operating system [34-37]. The program Virtual Box or VMware, which allows you to create a virtual environment on a computer, is called virtualization software. There are a variety of behaviors that can be observed when a malware file is executed, including the formation of new program's files, the removal of a system file, modification of a registry key, creation of new log entries, API calls, visiting of URLs, the installation of malware, and the transmission of information to the command-and-control scheme. The following steps determine if the file is benign or dangerous based on its contents. Dynamic analyses may be used to investigate files that were not properly deconstructed or evaluated by static analysis. Table 3 provides a high-level summary of the various dynamic analysis techniques.

Table 3: Dynamic analysis technologies

Name of Tool	Features
Process Explorer [9]	The system's task and control manager is called Process Explorer. It functions similarly to a Windows task manager by offering comprehensive details on the active system processes.
ProcMon [57]	The tasks performed by running processes are recorded by ProcMon (Process Monitor). It collects all operating of file system, registry updates, memory activity, and network operations system calls.
Wireshark, Tshark [58]	Network traffic analysis is used by Wireshark and Tshark. These tools can internment all inward and departing packets from the system to the outer world. Different system characteristics including port addresses, URLs, and data streams are collected and subsequently analyzed for malware file classification.
TCPdump [59]	Network data analysis in real-time may be accomplished via the use of this command-line application. It is a commonly used packet analyzer that shows in real-time the TCP/IP packets are sent and received from a computer system.
TCPview [60]	Windows networking tool that displays system statistics from every endpoint in UDP and TCP.
Regshot [61]	To capturing the registry modifications caused by the sample, the tool Regshot is utilized. The registry state of the Windows operating system will be captured both before and after the malware program sample is implemented. These both states may be equated to identify which changes have been made by the sample file that has been run.
Memoryze [62]	Memoryze is a command-based forensic memory tool that may be used to examine digital memories. There may be a full memory dump. Once the malware has been run, the memory dump is deleted from the computer. The memory dumps of rootkit malware are analyzed to extract different features such as the processes that are currently executing, strings, and the process that is being concealed from view.
Volatility [63]	Volatility is a cutting-edge framework for memory dump analysis that is constantly evolving. It is written in Python and has been built for various operating systems so that it may be used simply (Linux, Windows, and Mac OS). Advanced memory dump features like processes, registry keys, DLL-injected libraries, and strings may all be recovered with this tool, among other things.
Redline [64]	Redline is a portable manager that automatically collects information to examine the IOC. It is designed to be portable (Indicator of Compromise). Safety analysis software is used to examine several Windows components, including memory, the file system, the network, and registry entries.
Inetsim [65]	The network simulator is the component that imitates internet services to spread malware for virtual internet communication, as previously explained. It is Inetsim that supplies the malware with its virtual network environment, allowing the malware to perform its functions properly throughout the dynamic analysis process. Consider the possibility that inetsim may respond to such a query and that the malware will be permitted to continue to execute if the malware tries to connect to the remote device.
Fake DNS [44]	The FakeDNS software responds to DNS queries by generating a response. FakeDNS also resolves the virus's DNS requests, allowing the malware to handle them in real-world situations once the DNS queries have been resolved. Using the inetsim and FakeDNS programs, you may simulate virtual networks to better understand how the virus behaves.
Apate DNS [66]	ApateDNS, improved form of FakeDNS that incorporates a graphical user interface. When compared to the Fake DNS tool, Apate DNS is easier to set up and evaluate DNS responses.
Sandboxes [67]	For automated malware analysis, a variety of sandboxes, such as Cuckoo, Anubis, Panda, Limon, Parsa, and others, are utilized. These tools are composed of a variety of embedded analytical tools, such as those mentioned above. To remove the peculiarities of various groups, sandboxes and other tools are used in conjunction with one another, such as tcpdump for network traffic, the Cuckoo sandbox, volatility of memory dump, and so on.

2.3. Dynamic Analysis Framework

The dynamic analysis framework is configured to record the behavioral features of the executable files using hypervisor Type-1 and Type-2 [38, 39]. Type-1 hypervisors are recognized as bare-metal hypervisors because they handle and monitor guest computers directly on the hardware computer. Nutanix AHV, AntsleOs, VMware ESXi, XEN, Oracle VM Server Microsoft Hyper-V, are the patterns of type-1 hypervisors. Fig. 1 (a) displays the type-1 hypervisor architecture. Type-2 hypervisor run within the guest machine tools including Virtual machine ware, Virtual Box and Microsoft Hyper-V are employed to provide a dynamic analysis environment for type 2. VMware Player, Virtual Box, VMware Workstation, QEM, and so on are examples of type-2 hypervisors. Fig. 1 (b) demonstrates the type-2 architecture. In malware execution, the abstraction level is provided by these figures. Both kinds are based on the benefits and limitations stated in Table 4. The difference between methods for dynamic and static analysis based on their merits and demerits is given in Table 5.

Table 4: Hypervisor comparison for dynamic analysis

	Type-1	Type-2
Architecture	Virtualization of hardware	Most deprived code that works on low-end hardware
Detection methods	CPU Background performance low but not zero overhead	VMs have similar difficulties as emulators, although they may be more transparent.
Benefit	Close to hardware negligible overhead	Easy to use introspection and state control
Drawback	Lower introspection ability but measurable	Conceived for transparency compatibility
Protected	It is protected	little, if a host issue that affects the whole OS, like hypervisor,
Scalability	The scalability may be improved.	Less, reliant on host OS

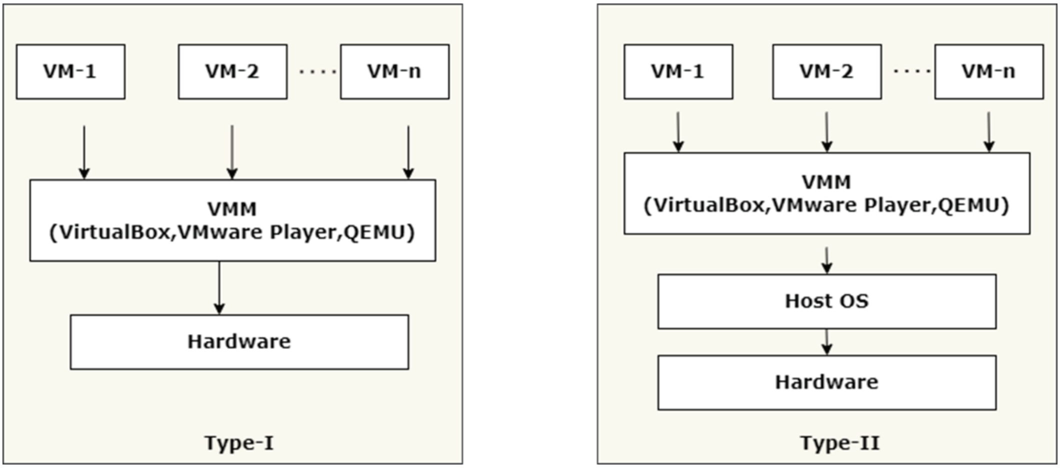

Fig. 1. (a) Hypervisor Type-I architecture

(b) Hypervisor Type-II architecture

Table 5: Brief comparison of static and dynamic analytical approaches

	Static	Dynamic
Methodology	The study is performed without the need to run the files in question.	Analysis takes place with running the files
Benefits	Quick and little time-consuming	Robust for handling obscure methods
Drawbacks	Unable to determine new malware, obscure techniques may simply be circumvented.	Time-consuming and complicated

3. MACHINE LEARNING FOR MALWARE DETECTION

In detecting and grouping malware, machine learning is becoming extremely helpful. There has been a great deal of effort in the literature to categorize benign and malicious files. ML technology offers additional options and flexibility to construct a more precise model by allowing for more properties of a malware and a benign file [26, 40]. In the realm of computer security, ML delivers a variety of options for the revealing of malware, infiltration, and harmful URL identification. Malware samples are examined, and the data collected for the training of the classifier is utilized. Fig. 2 displays the machine learning framework for the detection of benign and malignant files. This figure illuminates the simple building of ML classifiers, as well as the construction of classifiers in other problem areas. First step is feature extraction. After that feature representation and selection are completed. Last step is classification methods are employed to train malware classification models. Another advantage of employing ML in malware revealing is that it may be used to develop a model for identifying previously undiscovered malware. There is reasoning that it consists of many methods that may be used to create several malware algorithms to better identify malware. Furthermore, the following perks of employing ML algorithms to detect malware are some of its many uses.

- Current anti-viruses and sandboxing technique can be ousted.
- Automatically extracts samples of malware.
- The detection of unknown variants can be further generalized.
- It has the potential to decrease human efforts and time spent studying malware.

Many researchers have already carried out extensive research, demonstrating their high level of reliability regarding detecting malware.

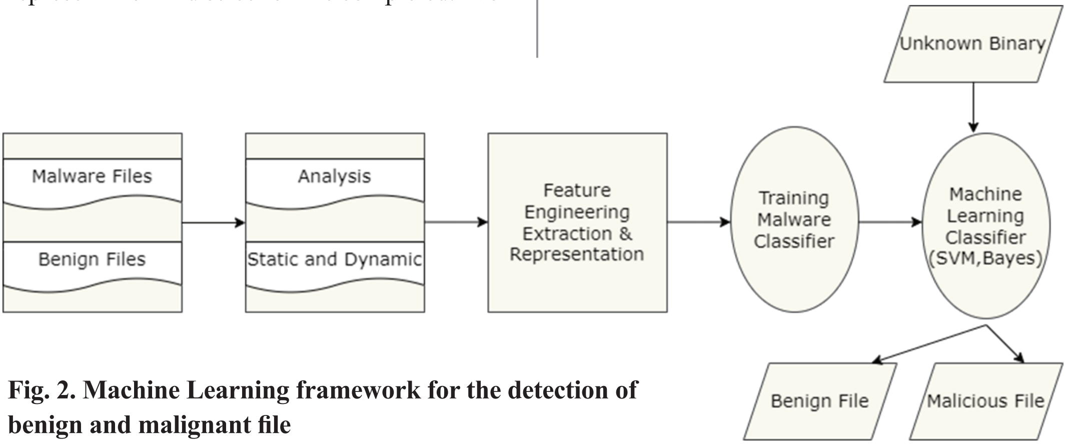
Unknown Binary
Malware Files
Benign Files
Analysis
Static and Dynamic
Feature Engineering Extraction & Representation
Training Malware Classifier
Machine Learning Classifier (SVM,Bayes)
Benign File
Malicious File

Fig. 2. Machine Learning framework for the detection of benign and malignant file

3.1. Implementing ML Challenges in Malware Identification

There are two main hurdles to implement ML in malware revealing, discussed in the subsequent sections.

3.1.1. High Computation Cost

Machine learning must be trained and upgraded to overcome the first barrier to using it in malware classifiers. Malware indicators must be restructured frequently to be effective. Computer views, where computer education has successfully been used, unlike other NLP domains, however, the classifiers often need to be re-trained to identify new and mutated viruses in this domain. According to the finding of the research everyday thousands of new viruses are created, and malware changes its behavior in a matter of hours or minutes. In contrast to other approaches, ML is thus more expensive and complex. As a result, while using machine learning for malware revealing, we must think in a different way.

3.1.2. Adversarial Machine Learning

In the field of computer-based malware categorization, adversarial machine learning is a major concern. The use of machine learning technologies by malware writers to escape malware detectors is critical for the development of unfavorable machine learning. In addition, Kolosnjaji et al. (2016) [41] pointed out that it was feasible to circumvent the machine-learning revealing method provided by the authors [42] with the aid of an intelligent escape assault. It is a fact that, apart from machine learning, there is no other technique available for detecting the most recent and greatly complicated malware. The development of malware is also moving at a breakneck speed. Now the Question is how we can hold these tests to employ machine learning in cyber security Realm. We may choose to minimize the dimensionality of the dataset to reduce training costs since machine learning algorithms take longer to learn from a dataset that includes more data characteristics than necessary. Consequently, only the most useful and discriminating malware features may make use of function selection and size reduction techniques to accomplish this. The second problem of an adversary's machine learning may be fixed by creating fusion malware classifiers. Fusion classifiers may utilize both dynamic and static information in their classification. It is possible to train some classification algorithms once the attributes have been extracted. A single machine learning method may be used to bypass the malware detector; however, the malware detector ensemble may be more resistant to unfavorable machinery learning than a single ML algorithm. Machine learning algorithms such as the vector support machines (SVM), Naive Bayes (NB), Random Forest (RF), and Decision Tree (DT), as well as set algorithms such as Random Forest (RF), and others in the classification literature have been used to train classification models. Machine learning techniques such as k-Nearest Neighbors (KNN) and others are also used to train classification models (ADA). ML classification methods are briefly discussed in Table 6, which includes a brief contrast of the different techniques.

Table 6: Description of classification methods for machine learning

ML Methods	Depiction	Benefits	Drawbacks
NB [43]	The likelihood to each class and the dependent possibility of every database within each class are computed using the NB method. To predict the probability of a class occurring in each data instance collection, the cumulative likelihood of the class is estimated by measuring the class chance and conditionally likelihood of every backing instance of data. NB can be utilized for both multi-class and binary classification.	It is quite easy to use and comprehend the NB classification method. It can operate well with non-relevant data. Furthermore, a tiny dataset may be used in the classifier.	The major disadvantage of the NB classification is that when the data features in training data are correlated, it is poorly performed. According to NB, data components should be autonomous.
K-NN [79]	The K-Nearest Neighbor (K-NN) classification technique divides the instance input into classes based on the class labels of the k-instances that are closest to the input instance. It is anticipated that the class of the input instance will correspond to the class of the majority. If it is necessary to catch the class label of the instance of input from the adjacent K occurrences, distance measures such as Euclidean, Manhattan, Hamming, and Minkowski will be employed.	KNN is simple to construct as novel instances with well-defined class labels, and can be restructured at a low rate. There are no assumptions on the data in the KNN algorithm. Search space is more resilient; thus, the data set is not linearly separable.	The major fault of the k-NN method is that, does not work when a data set is spread randomly. It is moreover uncertain to choose a suitable value for k.
SVM [89]	A hyper plane is used to split data instances into various classes in the data set entry by the SVM algorithm. A point vector in a two-dimensional space input may be seen to divide the instance of input data into two benign and the binary class. For correctly categorizing classes, usage of kernel functions in SVM classification training is essential. SVM classifiers utilize linear, radial, and poly kernel features often.	SVM is the most promising technique for classification since it provides high accuracy while yet being simple to use. SVM is capable of handling large datasets with multiple dimensions. It may also categorize separable non-linear data. Each issue has a regularization parameter and kernel function.	When the penalty parameter is set to a high value, the training time for SVM becomes very lengthy (C). Furthermore, the selection of the C value involves a balancing act between a test error and a training miscalculation.
LR [81]	As a parametrical binary classification technique, Logistic Regression is used to categorize data and divide it into groups. To build a logistic regression classifier, LR acquires the quantities from the training samples. In a qualitative response model, the likelihood ratio (LR) is employed to estimate the empirical parameter value.	It is less difficult to analyze and less complicated to use. The independent variables do not need to have identical variances, nor must a normal distribution with equal variances be used. In addition, since there are no linear connections between the independent and dependent variables, it may be used to cope with non-linear effects.	On average, the accuracy of LR's predictions is low. It has a lot of undesirable qualities that must be dealt with.

DT [90]	A DT is built by calculating the information achievement of each attribute in a dataset and applying that information gain to the decision tree classification. The root is the most important feature for obtaining knowledge. After then, the other transforms into a leaf of the plant's root. DT that has been created is then used to forecast the classes. Each node within the decision tree (not leaf) verifies the function, the function value correlates to the branch of the decision tree and a leaf-node is a class mark. The model has two division functions: information gain and Gini index. The model is trained.	High dimensional datasets and bright data may be handled using DT classifiers. In contrast to KNN and SVM classifiers, it operates in the white case. It is feasible to do a trained interpretation. This enables the trained model to be analyzed in depth. DT classification also has a high workout pace.	A minor modification of the dataset may lead to a big change in the decision tree structure, which will make the model unstable. It does not work properly because of the limited amount of data characteristics.
ANN [73]	Artificial Neural Networks (ANNs) represent standard techniques that help identify decision limits while reducing error rates in the same manner that human brains do.	ANN may be used to model a non-linear dataset with a high number of input characteristics. ANN may be utilized to solve virtually any issue, especially the optimum problem.	Over-fitting may be a problem for ANNs. The weights for training data may not be computationally costly for additional data sets, even though they are of the same demographics as the training data.
RF [91]	Random Forest is a machine learning method that makes use of bagging to improve performance. DT produces a single decision tree, while RF creates multiple decision trees based on separate sub-sets of the dataset with replacement, each of which is different from the previous one. The result of the RF is determined by the votes of each tree.	Since the RF learning machine algorithm randomly chooses various subsets, it is immune from data set variations and therefore minimizes the danger of overfitting. This provides the best results in categorization. It may provide excellent results, but the dataset varies, as opposed to the decision tree.	The training pace is sluggish. The number of classifiers that must be trained to create a powerful classifier is directly proportional to the strength of the classifier. Unlike decision-making book since many decision-making bodies are designed to produce a strong modal outcome. Because of this, it is challenging to assess the consequences.
Boosting Algorithms [67]	Alternative approach to ML is to boost processes that are used to train many weak classifiers sequentially. Without a replacement for sub-sets of data, weak classification is learned linear. The average of all weak graduates is then utilized to build a strong grading system. Another kind of algorithm is boosting machines. Boosting varies in many ways from bagging. Bagging subsets are chosen at random from the entire data samples, while the training subsets are designated from previously unselected data. Adaptive and Gradient boosting (GB) are two boosting approaches that are often utilized (AdaBoost).	Boosting techniques with any dataset works well. Boost techniques, like the RDF, address the variation in data characteristics. To train the week classifiers, several fundamental classification methods for machine research (SVM, KNN, DT) may be employed in conjunction with boost approaches before edifice the ultimate classifier.	Enhancing systems may be time-consuming and computer consuming. Increased algorithms on a real-time platform are tough to implement. A long and time-consuming computation is required for malware analysis since the classification method must be used to classify millions of malware samples.

4. MALWARE DETECTION TECHNIQUES

The objective is to detect and defend from harmful programs that could harm the computer systems or network properties. The input is evaluated in various instances so that malware samples are detected and classified into an appropriate family. It is necessary the use of specialized harmful file knowledge or expertise that accurately reflects the actual malware file behavior. Consequently, numerous malware program samples are evaluated utilizing dynamic, static and hybrid systems, depending on their complexity. It is then displayed properly to help with the continued training of the revealing system. According to authors [43], many distinct characteristics are used by several malware detection researchers, including Opcodes, strings, PE header information , API calls, ,Windows registry, files system accessing AV/Sandbox submissions, network Activities, and generated exceptions. Malware programmed files are assessed, and their characteristics are extracted and presented in an intermediate format prior to initiating malware detection. This intermediate form shows an important function in revealing. The false-positive rate will only be decreased if the collected data is optimal for malware revealing. The numerous malware revealing approaches presented are categorized into three primary classes, as mentioned below.

4.1. Signature-Based Malware Detection

This technique used to identify malware that is characterized by a certain file pattern and signature. It is a standard way of detecting known harmful files quickly, Compared to other techniques. Signature-based methods are often used in the development of antivirus software. One easy technique to construct the signature of malware program files by utilizing a hash algorithm such as message-digest algorithm (MD5), Secure hash Algorithm (SHA1) and others. Malware file signatures are generated and kept in the detection system database to check the unknown program file signature. It is assumed that if the signatures of the harmful file match, formerly it will be avowed as virus else the benign file. There is a problem with this because, if just one byte of a file code is modified, the signature of the program file will also be altered. It follows that a new signature must be created for each new malware variant and each upgraded malware variant. Only after that will a malware detector be able to identify this malware. In addition, new malware detection techniques based on signatures have been proposed, which make use of a variety of models, including program, graphic flow control, and mnemonic sequences. The malware data samples are investigated by means of a range of IDA Pro, Preview, PeStudio, and functional tools, among other tools, to determine if they are malicious. Previously mentioned standard and master learning techniques are utilized to train the malware detection system, which in turn is taught using the characteristics that were extracted. All the various methods to creating signature malware classifiers that are based on static characteristics is shown in Fig. 3. When it comes to developing malware detectors, there are many tools, static features, and methods to choose from. All the techniques discussed in this section based on signatures were proposed using this architecture.

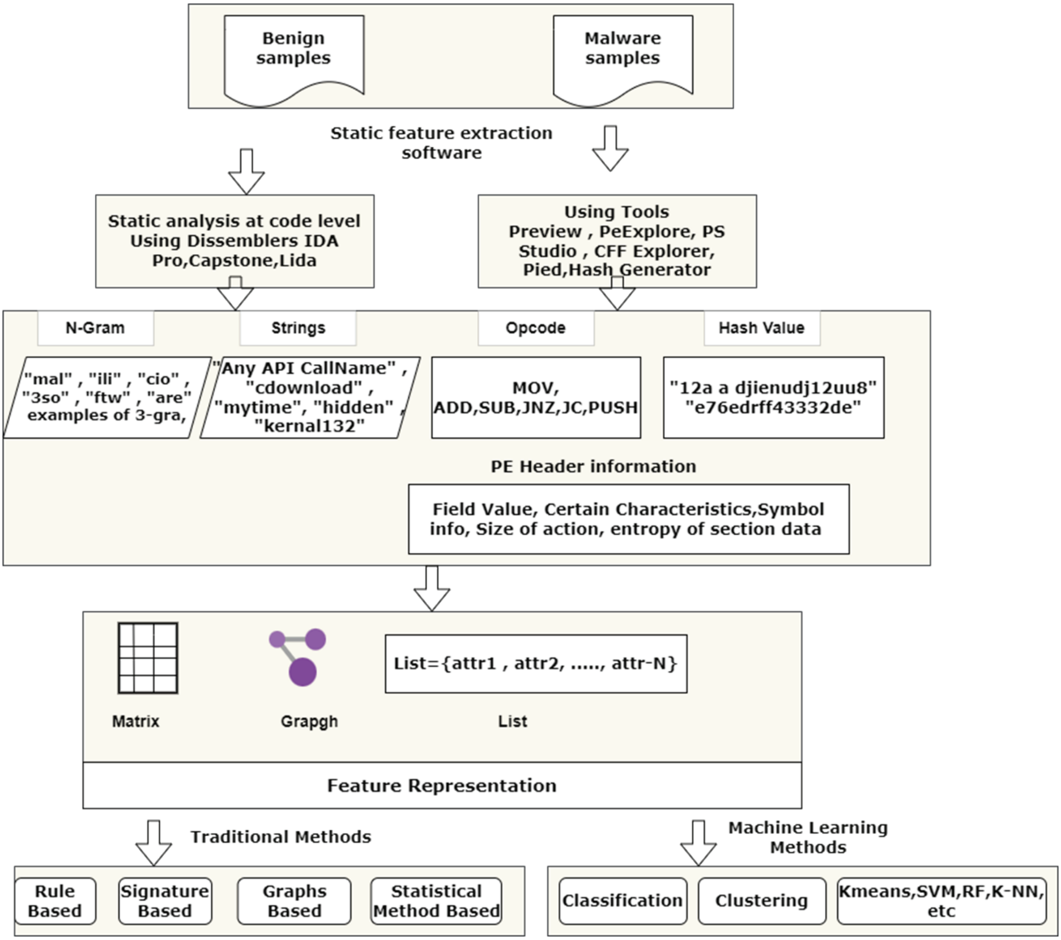
Benign samples
Malware samples
Static feature extraction software
Static analysis at code level Using Dissemblers IDA Pro,Capstone,Lida
Using Tools Preview , PeExplore, PS Studio , CFF Explorer, Pied,Hash Generator
N-Gram
Strings
Opcode
Hash Value
"mal" , "ili" , "cio" , "3so" , "ftw" , "are" examples of 3-gra,
"Any API CallName" , "cdownload" , "mytime", "hidden" , "kernal132"
MOV, ADD,SUB,JNZ,JC,PUSH
"12a a djienudj12uu8" "e76edrff43332de"
PE Header information
Field Value, Certain Characteristics,Symbol info, Size of action, entropy of section data
List={attr1 , attr2, ….., attr-N}
Matrix
Grapgh
List
Feature Representation
Traditional Methods
Machine Learning Methods
Rule Based
Signature Based
Graphs Based
Statistical Method Based
Classification
Clustering
Kmeans,SVM,RF,K-NN, etc

Fig. 3. The malware detection system architecture based on static extracted features

The assessment of numerous potential signature-based techniques is presented in this sector. Karnik et al. presented a malware revealing method where he uses the sequences of the function[53]. The sequence element represented the opcode group, and the function sequence was the hallmark of the malicious program file to distinguish the malware versions. The measure of cosine similitude has been calculated to deal with the obfuscation systems. Nevertheless, in this case, advanced obfuscation methods (equivalent instruction substitution) and packaged malware were unable to defeat the attack. In 2007, Bruschi et al [54] devised a method for categorizing malware using graphical representations [80]. According to the author, this method can control several fundamental obfuscation techniques. Binary files were constructed to match control flow charts to graphs of previously identified hazardous files and then executed. Two algorithms were utilized in this malware detecting technique. The first method searches for similarities between the two graphs of the binary program file B which is underneath assessment and, M. that previously recognized file malware. The B-file charts that had been reduced were then compared to the known

M-file charts in the second approach. Concerning the first and second methods, the author calculated the false positive rate as 4.4 % and 4.5 % correspondingly for 78 malware data samples that were established against these procedures. Nonetheless, this technique is incapable of dealing with zero-day malware. Based on the characteristics of an n-graph byte sequence, Zhang et al. suggested a technology for detecting and classifying malware using n-gram byte sequence properties [55]. A selection strategy removed pre-eminent bytes of n-gram that can indicate the malware program files. Subsequently, a classifier was constructed using the method of a stochastic neural network. For each classification, entailed of a series of malware detection decisions. Three malware courses have been occupied from the virtual VX Heavens database for training. The authors [56] presented an opcode-based machine learning methodology for detecting unknown malware program files. The Opcode have been arranged into the following sizes: 1 byte, 2 bytes, 3 bytes, 4 bytes, 5 bytes, and 6 bytes features.

In the n-byte function sets, four classification techniques were employed, including Decision Tree (DT), Naive Bayes, and Random Forest (NB) and Adaboost. According to the author, this method can considerably predict file maliciousness. Griffin et al. provided a technique for the identification of heuristic malware[68]. This proposed method generates a 48-byte sequence that was used as a string signature for identifying the malware strains. The authors employed the several module signatures for training the classifier instead of utilizing a single component signature. When compared to a single component signature, it has a higher probability of achieving high accuracy. On the other hand, it was unable to comprehend the effect of many component signatures during runtime. The one-sided perceptron method employed by Gavrilut et al. employed a single-sided perception of several machine learning algorithms to distinguish between benign and malicious [69]. This algorithm was intended to minimize the number of false positive rate. Firstly, they employed a basic partial perception cascade, now the unilateral partial perception system cascading demonstrated greater accuracy (88.79 %) compare to simple partial perceived cascades. The authors suggested a technique for malware recognition using the recovery of malicious file execution [70]. Malware flow control charts have been created using execution flow function calls. Graph matching has been performed to match the malware program files with the stowed CFG malware patterns. Malware patterns have been saved. The CFG was labeled harmful by a malware detector when it was included inside the CFG template. The issue is that obfuscation methods such as code rearrangement may escape malware detection by changing the real execution of the path of the malware. The authors created a text-based pattern matching technique for the development for the revealing of a malware system [71]. In this study the feed-forwards bloom filter was used to scan the whole collection of sample malware files.

Two kinds of outputs were generated: (i) malicious file matched and (ii) a subset of signatures. The Signature subsets were mined from the signature data base that was required to recognize the malware program files. A check reduces bloom filter-induced false-positive outcomes. It deals with the two problems. Firstly, the enormous database of signatures is easy to manage by decreasing those using subsets of signatures. The issue is that obfuscation methods such as code rearrangement may escape malware detection by changing the real execution of the path of the malware. The author created a text-based form matching technique for the development of a malware system identifications [71]. In this study the feed-forwards bloom filter was used to scan the whole collection of sample malware files. Two kinds of outputs were generated: (i) malicious file matched and (ii) a subset of signatures. The Signature subsets were mined from the signature data base that was needed to recognize the malware files. After that, a check is carried out to reduce the number of false-positive outcomes caused by the bloom filters. It deals with the two difficulties. Firstly, the enormous database of signatures is easy to manage by decreasing those using subsets of signatures. Secondly, a single-bit vector is used to manage the difficulty of memory scaling. Text-based detection matching method has not produced useful results that this text-based revealing model does not offer the high false positive outcomes. The authors have suggested a method for packaging detection framework to tackle obfuscation malware [45]. The primary goal of this technique is to detect malware that has been packed. Malicious sample have been evaluated utilizing static analysis for collecting the more refined executable attributes. Then, using a two-class vector machine support approach, a malware detector was trained to identify malicious code. Veeramani and Rai have built a system for malware detection utilizing the appropriate API calls[26]. The IDA Disassembler was used in this method to extract the malicious API calls. Unpacking tools were employed before the virus was disassembled to investigate packed malware. The malware program files and Windows system 32 are evaluated statically to excerpt the appropriate API for these both classes. SVM classifier training followed API call extraction. Even though various methods exist to deconstruct packaged malware, every infection is hard to detonate. Obfuscated malware makes API call extraction difficult. Consequently, there is high chances of the false positive results.

The authors in [72] Extended Moskovitch et al. [73] work, which was built employing opcode pattern set of features. Four sizes of 50, 100, 150, and 200 opcode n-gram sequences were created from malware program samples. Eight different classifiers were trained. Logistic regression, Artificial Neural Networks (ANNs), Vector Support, Naive Bays, Random Forestry, DT, and Boosted Naive Bays are some of the methods that are used to train the opcode patterns. In the study, it was discovered that the accuracy was above 96 %, and the false-positive rate was just 0.1 %. The API graph dependent malware revealing solution was proposed by Elhadi et al. [74]. In this method for each malware file to a data-based API calls graph were design. Apparently, a malware database consisting of data type API graphs was created. The analysis used the Longest Common Subsequence (LCS) technique to equal the unknown file's resemblance to the saved malware graph. However, in the trial, just 85 malware samples were utilized, which led to a detection rate of 98 %. In particular, The authors suggested the malware system for the portable executable program file of windows type PE header information [43]. This approach was utilized to train the model based on file metadata. The results of the experiments suggest that the executable metadata may be used to discriminate between benign and malicious. On the created attribute of the portable program executable header three important machine learning algorithms were used. Decision Tree classification beat NB and logistic regression classification. The authors joined the n-gram structure with the statistical examination with the malware-detection ML method [75]. The goal was to produce the co-operative architecture to identify the emerging malware i.e., metamorphic, those are difficult to identify by employing the statistical method of the n-gram model. The Markov blanket approach has been employed to choose the features. It has been employed because it minimizes feature size. In the next step, the Hidden Markov Model (HMM), which achieved an accuracy of 90 %, was trained in the resultant function sequences.

Srndic et al. submitted a study on ML algorithms for static analysis for the classification of malware samples [76]. The virus developer now uses portable document format (PDF) and shock wave flash (SWF) files to incorporate the executable scripts to harm system resources. This research has evaluated 40,000 SWF and 440,000 PDF files. To identify possibly dangerous programs included inside PDF and SWF files, this approach was employed in conjunction with other methods. When it comes to malware detection, there are many factors to consider. Kim et al. present a technique for PE malware header detection using the ML algorithm [77]. The main goal of this approach is to advance the revealing ratio as equated to preceding methods of malware revealing that focused on the PE header data of executable files. The portable executable header features of malicious and benign program files were observed in both cases (PE header information). From there on three ML procedures were applied to the retrieved feature: Gradient Descent (GD), SVM and Classification and Regression Tree (CART). To put the technique through its paces, almost 27,000 malwares and 11,000 benign files were used. This system has an accuracy rate of 98 % and a false positive rate of 0.2 %. This method, on the other hand, will not function if the actual PE header was obfuscated. Narra et al. developed a malware detection clustering approach that is competitive in terms of efficiency with SVM [3]. In their prior research, they have only applied the clustering technique like k-means, the maximizing of expectations with HMM. In the research, clustering techniques have been trained with 7800 malware program samples without HMM and effects were equated with the SVM classification, those have also been trained in the same data samples. It's problematic to regulate that many clusters are in the malware dataset, leading to a hit and a testing solution. Raff et al. developed byte n-gram-type approaches and examined the shortcomings of earlier n-gram strategies, those relied on the n-gram function [42]. The characteristics for Elastic Net Regularized logistic regression model training have been selected and the multi-byte identification has been analyzed. This approach led to the discovery of three significant faults in the prior n-gram methods. First challenge is how the earlier corpora was created the overestimate the detection precision, the second challenge was most n-grams have only been retrieved from string features and the third challenge was, n-gram feature has overpowered the classifier.

Searles et al. enhanced the previous work the usual malware detection graph program using control flow technique [93]. In this approach, similarities were found between CFGs retrieved from binary files using the Shortest Path Graph kernel (SPGK). In conjunction with a similarity matrix, the SVM method was then employed to enhance the precise classification. Various parallelization methods were assessed to lessen computational costs or to boost up the classification. It intimates a better accurateness on 22000 binary files in contrast to the 2 gram and 3-gram model. But it is quite difficult to cope with the enormous sizes of CFG. No such solutions have been familiarized to deal with this method. The authors utilized malware detection characteristics such as CPU utilization, network traffic, and swap usage [2]. This way, spyware known as APTs (Advanced Persistent Threats) detected. The findings further showed that the obfuscated malware can also be detected. The classifiers are trained on the self-organization feature map to minimize the problem of over fitting and produced great outcomes 7% to 25% higher than older methodologies. The authors have presented a technique for malware detection that analyses execution files using static analysis tools and extracts elements such as DLL import, hex dumping, and assembly code [34]. These structures have been applied to paragraph vectors via the training of SVM and k-NN algorithms, which have been used to analyze the data. Approximately 3600 malware samples were used in the trial, which resulted in a 99 % accuracy in detection. But simple obfuscated method can avoid the proposed method. The graph-based malware revealing method was enlarged by authors [94]. In the current methods each node represents a single system call freely however in the methodology every single node in the graph signifies a group of systems calls for related kinds for the compilation of the system dependence graph (ScD-graph). It is also possible to detect mutated malware using this technique, which involves categorizing a weighted directed network, also known as a group relationship graph (such as oligomorphic and metamorphic). The main issue with graphics detection techniques is the difficulty in matching graphs for similarity. Same-similarity and NP-similarity metrics have been proposed as potential solutions to this problem.

There have been 2631 malware samples utilized in the development of the anticipated model that comprises 48 malware families. The detection rate for the proposed model was 83.42 % when using the data from this dataset. The suggested system was assessed on 10568 binary program files, with an accuracy rate of 98.7 %, and it was shown to be effective. The authors [96] created SVM type classifiers, which are now widely used. Making use of the many features provided by the executable PE header. The static PE header features were split into 54 categories. A total of 500,000 malware data samples collected from the Vxheaven and Virusshare repositories were utilized to train the SVM classification system. There was a significant flaw in this method in that the proposed model was not explored in depth before it was implemented. In addition, there is no clear description of how to extract the static PE features of hidden malware from its source code.

Table 7 confirmations the complete analysis of the methods evaluated based on the signature. It also compares the methods suggested after a study of the signature-based malware classifiers. Further, this also summarizes the malware features, classification methods, and performance metrics utilized to develop the suggested strategies. In Real-time Detection Scenarios, signature-based methods are advantageous for reduced overhead and runtime. Some suggested methods have claimed greater than 99 % accuracy in malware detection. These methods used several malware features and showed their efficiency in the classification of dangerous and benign files. When combined with malware detection features, the representation methodologies show an important part, as demonstrated by Burn AP et al. (2017) [2] and the authors [78], both of whom are using API calling to improve detection rates but have used various machine learning and representation methodologies to do so. However, methods relying on signatures have some drawbacks, such as the inability to identify new and disguised malware. While we think that these methods may identify unknown malware, especially those which rely on heuristic signatures, they cannot detect malware that has been disguised. This is because the malware detector, which is based only on the static features of known malware, is rendered ineffective. Compared to traditional malware detection approaches, the presence of machine learning in signing-based methods has resulted in significant improvements in malware prediction (Fig. 2.). Using machine learning techniques, the authors in [42], [79], and [73] have obtained more accuracy than the research by authors in [80] and [75].

4.2. Behavior-Based Malware Detection

In the behavioral approach, malware is detected based on malicious activities carried out during execution. APIs, browser events, and system events, as well as network events, are all examples of feature kinds of behavior that have been specified [37, 81, 82]. These qualities are classified into three groups in the context of behavior methods: File activities, registry activities, and network activities. The entire process description to construct behavioral malware detection systems is illustrated in Fig. 4. New malicious Files can be identified by employing this malware detection because malware files somehow share harmful behaviors. Consequently, the malware detection system is trained to recognize similar behavior to identify new malware or variants of current malware that has been introduced. Similar activity, thus, is utilized to educate the system for malware detection to detect new malware or recognized malware variation. The behavioral method also provides a strategy for controlling the obfuscated malware. The obfuscation tactics utilized by makers of malware can dodge signature-based techniques. Behavior-based technology is trained in two ways. Anomaly denotes a malfunction that malicious files perform. When a file displays an aberrant behavior other than usual file stored behavior, the file is declared as a malicious file. The logic behind the malware is the abnormality (abnormal operation performed by malware). Malware detectors are being trained in the system to identify abnormalities. Benign files are examined using static or dynamic analysis. For the categorization of benign files, the usual activities of the benign files are used as a guide. Anomaly and benign analysis are the process in which malware data are analyzed alongside innocuous ones. Because both normal and dangerous activities are performed, it is desirable to distinguish between benign and malware behavior rather than using an anomaly-based approach. The detector must be trained for a longer time as compared to when using an abnormal technique. Behavioral malware detection techniques that use heuristics perform much better. Machine learning is much more important than traditional malware detection methods when it comes to detecting highly complex malware.

Table 7: A complete analysis of the signature- based malware detection techniques

Author(s)	Input	Data Source/ Number of malwares samples	Outcomes
Wang and Wu [45]	Portable Executable (PE) Header	1056-Packed Malware and3789-Unpacked Malware and Benign Files /Vxheaven and PCHome Malware Repositories	TPR-94.54%
Veeramani [26]	Application Programming Interface calls	214- Malware and 300-Benign Files Vxheaven Malware Repository	Accuracy-97%
Gavrilut [69]	Application Programming Interface calls	16437-Benign Files, 12817-Malware	Accuracy-88.78%
Shabtai [73]	N-gram Opcode sequence	7688-Malware and 22735-Benign Files / Vxheaven Malware Repository	Accuracy-96% TPR-95%, FPR-0.1%
Elhadi [74]	Data Dependent Application Programming Interface Graph	85-Malware Files / Vxheaven Malware Repository	Accuracy-98%
Markel [43]	Portable Executable Header	42003 benign files, 122799 malware files.	Accuracy-97% (Tree CART) 94.5% (LR)
Pechaz [75]	N-gram	1207-Malware and 194-Benign Files / Vxheaven Malware Repository	Accuracy-90%
Srndic [76]	Strings, API Calls	440000-PDF and 40000-SWF files / Virus total Malware Repository	Accuracy-99%(PDF)95%(SWF)
Wang [80]	Opcode Sequence	11665-Malware, 1000-Benign Files / Vxheaven Malware Repository	Accuracy-88.75%
Huda [78]	Application Programming Interface calls	Vxheaven Malware Repository and using Honeypot	Accuracy-96.84%
Kim [93]	Portable Executable Header	271095-Malware, 9773-Benign Files / Vxheaven Malware Repository	Accuracy-99%
Narra [3]	Opcode Sequence	7800-Malware Files / VirusShare dataset	Accuracy-98%
Raff [42]	Byte N-gram	400000-Malware and Benign Files / /VirusShare Malware Repository and Open Malware, MS Window	Accuracy-97.4%
Liu [19]	N-gram opcode, Image representation	20000-Malware Files	Accuracy-95.10%
Searles [117]	Control Flow Graph	22000-Malware Files	19% more accuracy than n-gram model
Nagano and Uda [79]	DLL import, hexdump and assembly code	3600-Malware Files / MWS 2016 Malware Dataset	Accuracy-99%
Nikolopoulos and Polenakis [96]	System Call Dependency graph	2631-Malware Files	Accuracy-83.42%
Le [97]	Greyscale images	10568-Malware Files	Accuracy-98.8%

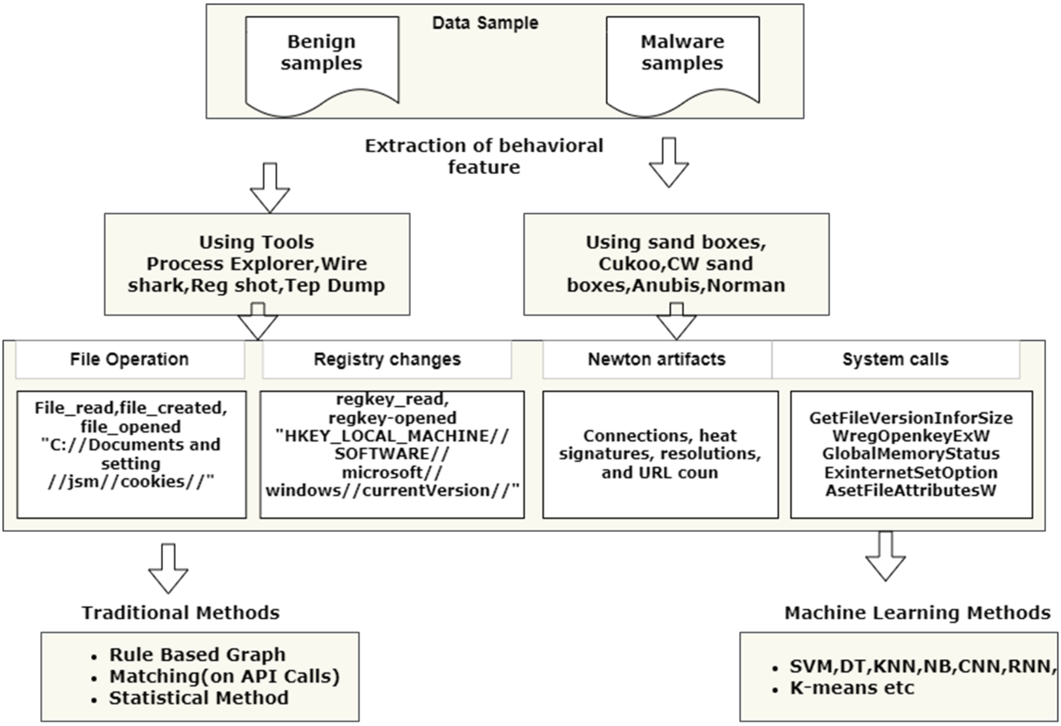
Data Sample
Benign samples
Malware samples
Extraction of behavioral feature
Using Tools Process Explorer,Wire shark,Reg shot,Tep Dump
Using sand boxes, Cukoo,CW sand boxes,Anubis,Norman
File Operation
Registry changes
Newton artifacts
System calls
File_read,file_created, file_opened "C://Documents and setting //jsm//cookies//"
regkey_read, regkey-opened "HKEY_LOCAL_MACHINE// SOFTWARE// microsoft// windows//currentVersion//"
Connections, heat signatures, resolutions, and URL coun
GetFileVersionInforSize WregOpenkeyExW GlobalMemoryStatus ExinternetSetOption AsetFileAttributesW
Traditional Methods
Rule Based Graph
Matching(on API Calls)
Statistical Method
Machine Learning Methods
SVM,DT,KNN,NB,CNN,RNN,
K-means etc

Fig. 4. Scheme of behavioral malware detection system architecture

The proposed behavior-based malware detection different algorithms are explained in this section. Bailey et al. provided an interaction malware detection technique with system services that included System API calls, special addresses, and functions [83]. Malware files behave in a variety of ways, which may be classified into many categories. A malware file, for example, may include a virus, a Trojan horse, spyware, or worms. Moreover, malware is also categorized according to the information retrieved in various classes and subclasses. The Malware Instructions Set (MIST) was suggested by Trinius et al. have described a new Malware Instruction Set (MIST) representing malware behaviors. CW sandbox used to analyze the malware samples[84] . The XML report is created by Sandbox and after that it is converted into a MIST format. In addition, machine learning and data mining have helped to improve the efficiency of malware analysis while simultaneously reducing the number of reports. Rieck et al. proposed an autonomous malware detection framework to detect malware class variants [84]. The framework divided into two steps. The first step uses the clustering method to make malware types comparable. The second step classifies or assigns the unknown malware file to the discovered classes. MIST representation was utilized to speed up the process of clustering and classification. Vasilescu et al. have made use of the Cuckoo sandbox to get dynamic analysis [85]. The generated report includes API calls to the system, log entries, and portable binary running information, among other things. The malware detector is trained with these extracted characteristics to identify zero-day malware. A data-dependent API graph-based malware detection method was developed by Elhadi et al. [76]. In the first stage, the clustering technique is used to create malware classes that have comparable characteristics to one another. In the second step, the unknown malware file is categorized or allocated to one of the recognized malware classes, depending on the properties of accelerate the clustering and classification process. Hegedus et al. utilized behavioral characteristics for malware identification to two classifiers k-nearest neighbor and random forest [86]. The technique proposed operates in two phases. In the first phase, the random projection is used to reduce the dimensionality and duration of the variable space. The index of Jaccard was used to measure the similarity of malware traits. Then the second phase is to utilize, detect and categories malware samples with Virus Total's K-nearest neighbor classifier. According to Mohaisen et al. they have used the malicious filename in a virtual environment to remove behavioral devices [81]. After that, malware samples are identified based on file transactions, network activity, registry key changes, and memory operations to determine their origin. SVM was used to categorize the data, and it was quite effective. Logic regression and hierarchical clustering methods were used to split Characteristics. A special version of this format, known as the MIST representation, was created to Malware into families that had similar characteristics, with each family having its own set of attributes. The experiment was conducted out on both medium and big datasets (each with 400 samples), and the results were compared (115,000 malware samples). The article implies 98% accuracy in the classification of malware. According to Ghiasi et al. proposed a new malware recognition approach using CPU registry entries [87]. In the monitored environment, this approach extracts API calls with dynamic analysis by running binary files. A similarity between two binary files was then calculated based on the content of their register. Four Machine learning methods are utilized include Random Forest, Bayesian Logistic Regression, and Bayesian Logistic Regression with Bayesian Logistic Regression. To train the detection model, regression, SMO, and J48 have all been used. The article declares well-regulated time to match existing patterns efficiently. In the Cuckoo sandbox method, Pirscoveanu et al collected the contextual features of binary files. A random forest algorithm was developed for the categorization system [65]. Ki et al. presented a technique of malware identification that relied on the API call sequences [108]. The study shows that dynamic analysis has been proven to be more effective in obtaining malware behavioral features. it is observed that all the malwares contain Sequence alignment technique has been used to address redundant or non-relevant code insertion in malware. In the testing step, if unknown file calls are matched to stored API patterns for the extracted APIs, the file is declared malicious else the file is benign. Pan et al. proposed a malware classification system based on the BPNN model [88].

The HABO system is designed to collect runtime functions. The essential elements of the report were extracted, for example reading foreign memory, creating mortexes, creating the process, or modifying registry entries. The model was developed utilizing the method of the Back Propagation neural network. Narayanan et al. developed a supervised machine learning techniques to build the classification of malware [92]. Polymorphic malware was managed using the suggested method as pictures capable of capturing small changes while maintaining the general structure. Three classifiers have been trained on the provided dataset using the KNN, ANN, and SVM methods. Cho et al. presented a technique employing the API call sequence [93]. API calls have been extracted by running malware samples in cuckoo sandbox for the construction of the API sequences. 150 samples from 10 malware families were trained and 87 % accuracy was obtained. Mira et al. presented a research in which they built an API-based malware detection model that was trained in two algorithms: the Longest Common Subsequent (LCSS) and the Longest Common Substring (LCSS) (LCS) [90]. The multi-process execution behavior of malicious files was presented by Bidoki et al. [57] . Malware may disseminate its activities over a wide range of legitimate operations, but the overall effect is always negative. It is necessary to use the improved learning method during the training phase, and every API request must be gathered during the detection phase. The execution rules of all processes have been merged to determine whether a binary file is harmful. A graphical call system technique was proposed by Ming et al. (2017) [94] that generates an invisible malware problem system call-based dependency diagram using a graphical call system. According to the author, all variants of the same malware have the same semantics; the only difference is in the malware's syntax. A technique was trained on 5200 malware files and then tested on 960 malware files before being released. The model that was proposed was 97.30 % accurate. According to Wagner et al. (2017) [95], malware may be visually identified by its activities. This system was created to monitor the visual pattern of malware activities using knowledge-assisted visual analysis. Mao et al. (2017) [113] established a methodology wherein they assess, based on their usefulness, the importance of system subjects. This allows us to create a network for security dependencies that gives us an insight into the value of system object security. Enhanced DT It was decided to utilize Amazon Web Services to host the new cloud-based design, which provides for greater scalability. To train the classifier, 150,000 malware samples and 87,000 benign samples were used. Using the improved DT algorithm, the detecting system obtained a 99 % accuracy rate in its detection.

Ding et al. proposed the idea of graph-based malware detection [31]. Instead, then creating a behaviour graph for each malware, the author proposes a standard graph for every malware family. Dynamic taint analysis technique was used for building the behavior graph. The highest weight parameter subgraph was used to compare the graph of an unknown file to the graphs of each malware family that had previously been created. Stiborek et al. have presented a technique for capturing malware behavior by running malware in a sandbox environment [99]. The sandbox has a series of names and resources for each malware sample. The term for this approach to the issue framing is multiple instance learning. Machine learning is used to samples of varied sizes of malware; a report on many instances of learning explains several ways to deal with the issue of sample size. To address the issue of sample size fluctuation, a vocabulary approach was employed. The method suggested was developed in a big 11,2115 binary and obtained a precision rate of 95.4 %. Ghafir et al. presented an advanced persistent threat detection approach. [100]. To train the classifier, the author utilized SVM, KNN, and group methods. This method successfully generated the accuracy of 84.8% in APT prediction.

Run-time features were used by Alaeiyan et al. to construct a malware detection algorithm [101]. Using the Parsa sandbox, this method has also discovered evasive malware. The given method was tested on 1100 malware samples and was shown to be accurate to 97.9% of the samples. For the Windows platform, Xiaofeng et al. created an API that uses sequence-based malware classifiers.[102]. The cuckoo sandbox was used to eliminate dynamic API calls from the application. Following the training of two classifiers. A model for categorizing malicious traffic was proposed by Arivudainambi et al. in their study of network traffic analysis [103]. They integrated PCA to handle sophisticated anti-network traffic analyses. The approach presented was evaluated via the Noriben, Cuckoo, and Limon sandboxes by running 1000 malware samples. This technique has a 99% accuracy rate in terms of malware identification. Yucel et al. presented a techniques for creating executable memory file images [104]. A total of 123 malware samples from various families were collected. It was discovered that malware samples ran similarly when run in virtual machines, and 3D memory snapshots were created for comparison. It showed that various families of malware had varying rates of similarity, such as 0.99 for Marina Botnet, 0.99 for Rex Virus, and 0.886 on average. Rabbani et al. presented a model for the detection of malicious behavior conduct in network traffic using a Probabilistic Neural Network (PNN) [105]. The vector featured IP, TCP, UDP, CON, jitters, and other network capabilities. A modified version of this technique was developed by combining the PSO (Particle Swarm Optimization) algorithm with the PNN algorithms, which resulted in a malicious traffic detection rate of 96.5 %.

Table 8 compares all the behavioral techniques that were examined. There is now more research being conducted on behaviorally based techniques of malware identification. This is because signature-based techniques are incapable of dealing with emerging and zero-day malware. Using runtime features, these approaches have outperformed signature-based accuracy solutions in terms of accuracy. Many researchers, including Elhadi et al. [106], Pan et al. [88], Ali et al. [67], and others, have made use of dynamic API calls. Alaeiyan et al. [101] have used file, registry, and network activities in the training of malware classifiers using supervised classification techniques, as have Stiborek et al. [99], Paketas et al. and Stiborek et al. [99]. For instance, Ghafir et al. [101] have been developing their models with several runtime characteristics. The detection rate is considerably greater for behavior methods as compared to signature based. The methods also suggested to claim that new and obscured malware may be predicted. Historically, it has taken a long time to extract the runtime function from conventional behavioral methods; however, the use of machine-learning algorithms has sped up the process, allowing the proposed model to make use of more data and larger malware samples to train and test the malware classification system. However, the implementation of these suggested methods presents certain problems and difficulties. The proposed methods are tested and verified using a range of malware samples. Furthermore, classifiers vary in techniques for training. Before prediction, high processing time and the running duration of malware samples are obstacles to applying the behavioral technology to a real system. The advantages of signature and Behavior methods, described in the next section. In addition, hybrid approaches have been put out and will be examined in the section.

Table 8: A Complete analysis of the behaviors-based malware detection techniques

Author(s)	Input	Data Sources/ Number of malwares samples	Outcomes
Elhadi [106]	Data Dependent Application Programming Interface Graph	416-Malware and 98-Benign Files / Vxheaven Malware Repository	Precision 98 %
Mohaisen [81]	Application Programming Interface calls and Network Activities	115000-Malware Files / Antivirus companies	Precision 99.5 % Recall 99.6 %
Ghiasi [87]	Contents of Registers, Application Programming Interface Calls	850-Malware and 300-Benign Files	Precision 95.9 %
Pirscoveanu [65]	Window Application Programming Interface calls	42000-Malware Files / VirusShare and Virustotal Malware Repositories	Precision 98 %
Ki [118]	Application Programming Interface Call Sequence	23080-Malware Files / VirusTotal Malware Repository and Malica Project	Recall 98.8 % F1-Score 99.9 %
Pan [88]	Application Programming Interface calls	13600-Malware Files / Kafan Forum	Precision 98 %
Nayaranan [92]	Representation of Malware in form of images	10868-Malware Files / Kaggle Microsoft Malware Dataset	Accuracy 96.6 % (Linear KNN)
Cho [93]	Application Programming Interface call sequence	150-Malware Files / Vxheaven Malware Repository	Precision 87 %
Mira [90]	Application Programming Interface call sequence	13600-Malware / VirusSign, SAMI, CSDMC Malware datasets /	Precision 99 %
Mao [119]	System objects	7257-Malware Files / Vxheaven, MALICA and Virustotal Malware Repository	Precision 93.92 % FPR 0.1 %
Bidoki [57]	Application Programming Interface Calls	378-Malware and 500-Benign Files / Vxheaven Malware Repository	Precision 91.66 %
Ming [94]	Dependency Graph	5200-Malware Files	Precision 97.30 %
Wagner [95]	Sequence of Application Programming Interface calls	8847-Malware and 1460-Benign Files / Vxheaven Malware Repository	Precision 95.25 %
Pektas and Acarman [98]	Registry, Network, File System, Application Programming Interface call sequence	17900-Malware / Virusshare Malware Repository	Precision 92.5 %
Ali [115]	Run-time features	150000-Malware and 87000-Benign Files / Malware Repository of Nettitude	Precision 99 %

Stiborek [99]	Behaviour artefacts	112115-Malware Files	Precision 95.4 %
Alaeiyan [101]	Behavioral features using Parsa sandbox	1700-Malware and 1700-Benign Files / Virusshare Malware Repository	Precision 97.9 %
Xiaofeng [102]	Application Programming Interface call Sequence	1430 Malware and 1352 benign Files / Virusshare and Virus Total Repositories	Precision 96.7 %
Arivudainambi [103]	Network Artefacts	1000-Malware Files	Precision 99 %
Rabbani [105]	Network features	677789-Benign Files and 22211-Malware	Precision 96.5 %
Namavar [4]	Behavioral features	18831-Malware Files / Vxheaven and Microsoft Kaggle	Precision 99.65 %
		Datasets	
Yucel [104]	Memory Images	123-Malware Files / Virusign Malware Dataset	Precision 99.5 %

4.3. Hybrid Malware Detection

Behavior-based and signature-based techniques have some advantages and disadvantages. Consider that several researchers have suggested ways to hybrid malware detection that incorporate the advantages of both static and dynamic malware techniques. In this part, we discuss and compare the research of different methods for hybrid malware detection using various criteria. Rabek et al. suggested a malware detection technique for detecting harmful files that were obfuscated [107]. In addition to the dynamics, the information gathered includes all the function names, addresses, and return addresses of system calls. Malware files have been executed to record runtime activity in a controlled dynamic environment. The program is classed as malicious when calling the same previously saved system calls (which represents the malware file). If, however, any unnecessary system calls are included in the code, the malware creator fails. The graph detector for network worms was suggested by Collins et al. [108]. Hosts in a network were represented as nodes and connections as edges. The approach mimicked a network to learn about the behavior of worms. It addresses worms exclusively, not Trojan horses, viruses, or other malware. The following are the different methods used for detecting hybrid malware. To minimize false-positive reactions, Mangialardo and others [109] suggested that the FAMA framework address faults in both static and dynamic analysis methods. For extracting static characteristics, IDA pro was utilized and Cuckoo's sandbox for capturing behavioral functions. The features gathered were then used to train the random forest and C5.0 algorithms, which were both developed by the researchers. The tests showed that the unknown file can be classified as benign or dangerous with 95.75 % accuracy. Shijo et al. presented an integrated malware detection approach that was based on machine learning [110]. Following the deconstruction of the binary files, information in the form of readable string information was obtained. It also takes care of any superfluous printed strings that were used to conceal the code. Once the binary files had been run, they were subjected to dynamic analysis in the Cuckoo Sandbox. This included file system-relevant API calls, change registries, and the extraction of a specific process and memory addresses. The Random Forest and Support Vector Machine were used to train the malware classifier. Edem et al. suggested a malware detection methodology that is automated [111].

Several data-mining techniques have been used in integrated with the author's malware investigation. A clustering technique called k-means clustering was used to group malware samples that had similar behavior together to make the analysis easier and more useful. The malware samples were analyzed both statically and dynamically from the outset. The IDA Pro and OllyDbg tools, as well as the CW Sandbox, were utilized to extract the static characteristics from the code during the static analysis. An XML report on the malware sample activity is generated by the CWSandbox. The data mining method was then utilized to process both static and behavior characteristics. The enhanced malware detection method for malware categorization using the SVM algorithm was proposed by Okane et al. [112]. Unlike others, this method utilizes the runtime trace as the tracking feature of the application to train the detection system. The Support Vector Classifier was trained after the extracted functions were reduced to smaller feature sets utilizing opcode filtering methods, which resulted in the reduction of the retrieved functions to smaller feature sets. Nauman et al. introduced the concept of tridimensional decision-making in the context of malware detection systems [113]. All the previously proposed techniques were accompanied by a binary file, which might be malware or benign. Practically, the detector is unable to correctly classify all the examined files. Some complicated malware is wrongly categorized, such as Stuxnet or extremely unique files. In this instance, detectors are more likely to produce false positive or negative findings. Thus, the strategies of fered address such viruses with three types of accepted, rejected, and delayed decisions. Two approaches to malware are proposed: (i) Rough theoretical rough game sets (ii) rough theoretical information test. The disadvantage of this method is that it does not offer a solution for the handling of malware that has not been detected yet. By combining various static and dynamic analytic methods with component-based frameworks, Kaur et al. have created a hybrid methodology for identifying malware that may be used to detect a wide range of threats [20]. Initially, the Hybrid Framework was developed to automatically identify zero-day malware that mimicked the destructive behaviors of existing malware. The malware detector is trained by extracting static and dynamic characteristics from malware samples, which are then used to identify malicious code. This technique captures a broad variety of static properties, such as the hash value, PE header information, string values, and dynamic actions, such as process activities, file operations, storage operations, and network activities. This method makes it possible to identify new malware and classify it according to its static characteristics.

Kolosnjaji et al. proposed an improved semi-supervised malware detection approach that incorporates both dynamic and static malware analysis results to improve the performance of categorization and classification

[41]. During the processing of extract characteristics of dynamic and static analysis, various methods were employed, which are distinct from prior malware detection approaches. To categorize static findings, the semi-supervised propagation method has been utilized and the dynamic reports, which found hidden semanticized characteristics in malware files, have been statistically modeled. Above all, it provides an online dynamic malware classification system for non-parametric techniques. Damodaran et al. proposed a hybrid approach for training a classifier, in which an Opcode sequence or an API request was utilized to train the classifier in parallel [114]. Like earlier hybrid techniques, binary data is used to extract both static and dynamic features from it. The Hidden Markov technique was then used in the classification process, both dynamically and statically. This approach only produced good results for a few malware types. Current methods of obscuring may also avoid the static analytical procedure of malware identification using malware tactics. Pfeffer et al. suggested MAAGI for malware detection (Malware Analysis and Attributed using Genetic Information) [115]. In this context, the genetic algorithm has been used to the comparability features of malware. Malicious samples are processed in static and dynamic sandboxes to collect malware features. Static analysis was carried out using PEid and IDA pro tools while dynamic analysis tools were utilized on Symon and Introvert. The foundation of the MAAGI is founded on the notion that biological behavior and malware behavior share a great deal. The outcome was the creation of the malware detection framework using the artificial intelligence algorithm, which showed promising results. It is expected that it will improve collaboration between cyber defense and artificial intelligence groups in the long run. Huda et al. suggested a semi-supervised machine learning method that would automate information regarding unknown malware in the sensing system using previously tagged and unlisted data, which would be implemented in a sensor network [89]. Other techniques do not have the advantage of automatically updating the database of the detection system, which makes this one stands out since it does not need external help. This technology makes use of k-means clustering with reverse document frequency as the distance metric, word frequency as the distance metric, and the Support vector machine technique to categorize binaries to extract cluster information, all of which are used to extract cluster information. Huda et al. [116] proposed a hybrid method that combines the inclusion of wrapper filters with the selection of characteristics to get the best possible results. The author has selected the maximum and minimum characteristics in this research After that, several MR+SVM, MRED+SVM and Fisher+SVM machine learning methods were employed to train the model that provided 99,49 % precision utilizing. Table 9 offers a comprehensive summary of the updated techniques for hybrid malware detection, as well as their advantages and disadvantages. It also presents the results of the investigation into hybrid malware ratings. Huda et al [116] achieved the highest accuracy in malware detection with a 99.49 % detection rate. These methods were proposed as a means of bridging the gap between static and dynamic malware detection by integrating the benefits of both approaches.

Table 9: A complete analysis of the hybrid malware detection approaches

Author(s)	Input	Data sources / No. of samples	Results
Mangialardo [109]	Application programing interface call sequences	131073-Malware Files / Virus Share Malware Dataset	Precision-95.75%
Shijo [110]	Printable strings information (PSI) and Application programing interface calls	997-Malware and 490-Benign Files	Precision -98.7% (SVM), 97.68% (RF)
Edem [111]	Signature and Window Application programing interface calls	1143-Malware Files / 1143-Malware / Malware Sample taken from MWanalysis.org	
Okane [120]	System Application programing interface calls	350-Malware, 300-Benign / Vx heaven Malware repository	Precision -86.3%
Nauman [113]	System Calls	UNM Application Dataset	Precision -92.51%
Kolosnjaji [41]	Portable Executable Header Information and	2000-Labelled Malware and 15000-Unlabelled Malware Files / Virus total Dataset	Precision-90%
	Application programing interface Calls		
Damodaran [121]	Portable Executable information, Application programing interface Calls	745-Malware and 40-Benign Files / Vx heaven Dataset	Precision -98%
Pfeffer [115]	Application programing interface calls	8336-Malware and 128-Benign Files / MIT Lincoln Lab Malware Dataset	Precision -86%
Huda [89]	Printable Strings, Imports, Procedure Call Graph (PCG)	967-Malware Files / CA Technologies VET Zoo Malware Dataset	Precision -93.83%, FPR-0.144%
Huda [116]	Runtime activities	2000-Malware and 1500-Benign Files / Malware sample from CA Technologies VET, ZOO, Offensivecomputing.net and Vx heaven Repositories	Precision 99.49%

5. Discussion and analysis of proposed malware detection techniques

Authors	Type		ML Algorithm								Features						
	SA	DA	DT	SVM	KNN	NB	LR	ANN	RF	ADA	MN	PEH	AC	PSI	RC	IM	RTF
Ali [67]		✓	✓	✓						✓			✓				✓
Gavrilut [69]	✓							✓					✓				
Ghafir [100]		✓		✓	✓				✓	✓							✓
Ghiasi [87]		✓											✓		✓		
Huda [89]	✓	✓	✓	✓		✓			✓				✓	✓			
Huda [116]	✓	✓		✓													✓
Huda [78]	✓			✓									✓				
Ki [118]		✓											✓				
Kim [77]	✓		✓	✓								✓					
Kolosnjaji [41]	✓	✓							✓	✓		✓	✓				
Le [97]	✓							✓								✓	
Liu [19]	✓								✓		✓		✓			✓	
Mangialardo [109]	✓	✓	✓						✓				✓				

Mao [119]		✓															✓
Markel [43]	✓		✓			✓	✓					✓					
Mohaisen [81]		✓	✓	✓	✓		✓						✓				✓
Nagano and Uda [79]	✓			✓	✓						✓		✓				
Narra [3]	✓			✓							✓						
Nauman [113]	✓	✓											✓				
Nayaranan [92]		✓		✓	✓			✓								✓	
Okane [120]	✓	✓		✓									✓				
Pan [88]		✓						✓					✓				
Pfeffer [115]	✓	✓											✓				
Pirscoveanu [65]		✓											✓				
Raff [42]	✓						✓				✓						
Searles [117]	✓			✓													
Shabtai [72]	✓		✓	✓	✓	✓	✓	✓	✓		✓						
Shijo [110]	✓	✓		✓					✓				✓	✓			
Srndic [76]	✓			✓									✓	✓			
Stiborek [99]		✓		✓					✓								✓
Veeramani [26]	✓			✓									✓				
Wagner [95]		✓			✓								✓				
Wang [80]	✓										✓						

Table 10: Analyzing three kinds of malware revealing methods: ML algorithms and malware characteristics

The statistical examination of ML revealing techniques is covered in this part of the paper. Because there are three different types of malware detection techniques. These methods may be classified into a variety of categories, including analytical approaches, conventional or machine-based learning systems, deep learning systems, computer technologies, and mobile malware detection technique. We have examined the strategies of malware detection designed for Computer malware. Table 10 summarizes research into methods of malware detection based upon algorithms and the features of input. It shows which algorithms have been greatly utilized and employed in past and present and which characteristics are static and dynamic.

- Static Analysis- (SA)
- Dynamic Analysis- (DA)
- API Calls, Imports, DLL Import- (AC)
- Image Representation of binary file- (IM)
- Register Content- (RC)
- Portable Executable Header- (PEH)
- Runtime features (FILE, Network)- (RTF)

In addition to the analytical category and the machine learning systems, malware types are the important thing in malware detection. For example, hash values, specific string data, opcodes, n-bytes, and registry changes, practice activities, system operations, and other system data. As exposed in the Table 10, most of authors used single-feature sets while many others used multiple different features for the classification of malware. Moreover, it affects malware detection performance how the characteristics are handled and displayed. Many techniques, such as API calls, runtime functions, opcodes, and n-grams, among others, have been tried; nevertheless, the processing and application of different methods have formed superior outcomes overall. Fig. 5 expressions the accuracy contrast to various static malware revealing techniques. The greatest accuracy of 99 % was obtained by [117] the authors in [77], and [27] using the SVM classification methods. Fig. 6 exhibits dynamic malware classifier's accuracy. The authors in [4], [104],and [17] some researchers have reached more than 99 % accuracy. The authors [116] generated the highest precision of 99. 49%, as shown in Fig. 7, utilizing MR+SVM and MRED+SVM systems. From the previous work we can elaborate that static and dynamic technology can produce a more precise malware revealing system with machine learning. Likewise, for constructing a model employing different malware traits a single classification technique is not suitable. However, SVM did better in static analysis than other systems. The algorithms of the dynamic examination collective also worked effectively. Some significant research problems arise after the analysis of different malware detection methods. There are several advantages and disadvantages to each method. For instance, signature-based malware detection methods may identify only known malware that has previously recorded its signature in the detector database. Malware detection systems based on a signature can be readily avoided by obscure methods. Malware detection technology that is based on behaviour may be used to remedy flaws of signature-based detection technique. However, behavioral methods require far more time and have a larger false positive rate than signature-based techniques. The identification of malware is an infinite process. By the day, it's becoming harder. More attackers create advanced malware which is rigid to identify though the number of computer users is growing. In addition to the intricacy of malware, malware is also a huge difficulty in stopping malware attacks. Therefore, as improvement increases, the fight between malware developers and security analyzers never ends. In the following part, we proposed a framework for addressing these malware detection challenges. It would not be a strengthen, but the maximal virus detection dimensions are included in this technique.

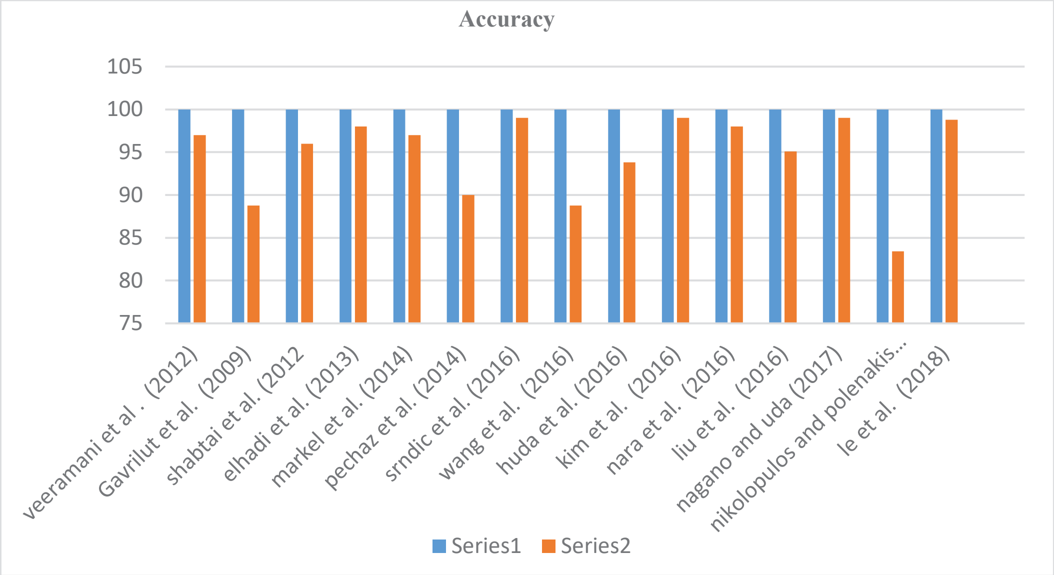

Fig. 5. Chart showing the accuracy contrast of static malware revealing methods

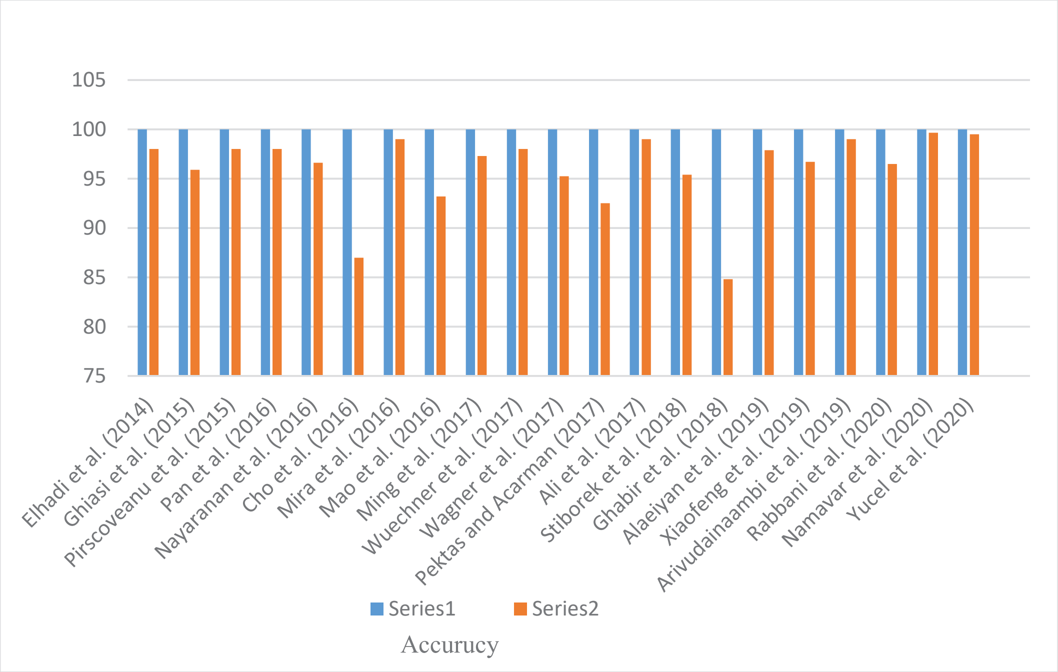

Fig. 6. Comparison Graph of Dynamic Malware Detection Techniques for Accuracy

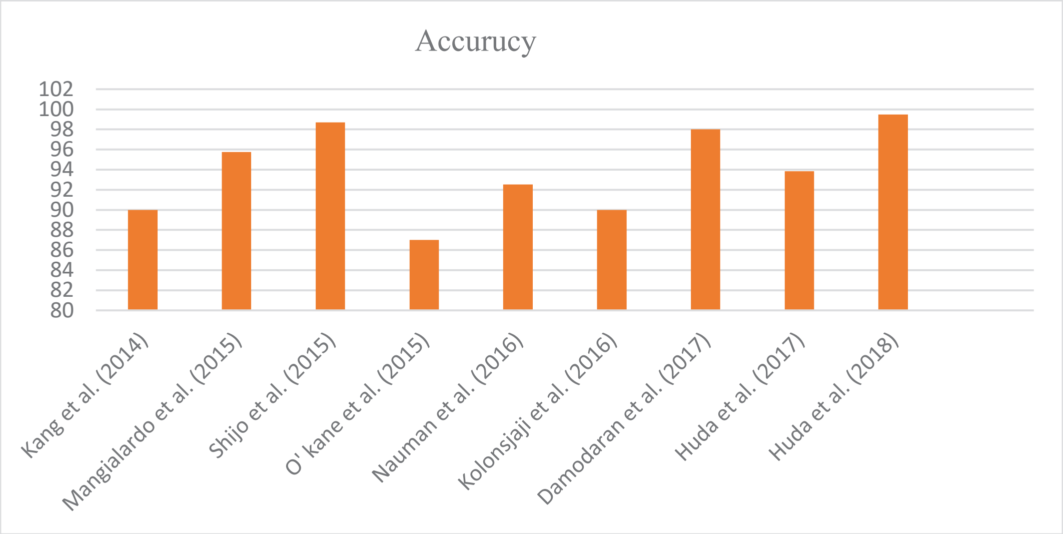

Fig. 7. Comparison Graph of Hybrid Malware Detection Techniques for Accuracy

6. IMPORTANT CRITERIA FOR MALWARE DETECTION SYSTEMS DEVELOPMENT

This section deals with several important elements of malware creation, providing the scientist insights into how to investigate this issue.

6.1. Handling Anti-Analysis Technique

The issue arises when malware is produced employing the method of antistatic or anti-dynamic analysis. In dynamic analysis, malware detects the analytical environment and hides or inhibits actual activity from occurring in this environment. The issue occurs when malware is produced using antistatic or dynamic analysis methods. The malware identifies the analytical environment and conceals or discontinues to run in that environment in dynamic analysis. As a result, throughout the development of the malware analysis system, analysts will be tasked with addressing this issue. In the case of static analysis, malware analysts must develop unpacking software and build a dynamic environment by addressing all the patterns of virtual control devices used by malware to control the analysis environment to do the static analysis. Here are a few indicators of how dynamic malware analysis may be configured for malware samples.

- The Default media access control address of virtual machine may be altered so that malware does not detect virtual machine with virtual machine standard and known media access control access address. Make a note of the names of virtual machines that seem to be the host systems. For example, Ahsan-pc, Sunny, and so on.
- Install all the essential program applications, including Microsoft Office, Adobe Reader, VLC, and others, to give your machine the appearance of a personal computer.
- Create subdirectories for documents in various folders, such as Documents, Desktop, Downloads, and Temp etc. Also, utilize the VM machine for many days to ensure that malware does not identify it as a new computer suitable for malware analysis and mark it as a target. Several documents are generated, as well as cache and other temporary files when accessing the internet after utilizing a virtual machine for personal work.

6.2. Analysis Tools and Environment Setup

Apart from the anti-analysis methodologies, the tools and kind of environment setup especially in dynamic analysis have a big impact on the accuracy of malware classifiers. In static analysis, several static extractor tools were utilized, including IDA Pro disassembler, capstone, Peid, PsStudio, and others. Statistical analyses are conducted on deconstructed files and disassembled malware files. Disassemblers extract more comprehensive characteristics which cannot be obtained with basic tools such as CFF Explorer, Psfile, IOC Finder, etc. The dynamic analytical analysis is influenced by the kind of architecture of the analytics environment. The hypervisor Type 1 is more robust than the hypervisor type 2 when it comes to anti-analytic methods. However, both have advantages and disadvantages that are addressed in Section 2. Thereby, the developer must select techniques for extracting the malware feature required while building an anti-malware system.

6.3. Data Samples for Malware Classifier Training

Samples of malware are gathered in the

academic literature from different sources. Certain sources like Virus Share enable scientists to download millions of malwares, including the latest. Vxheaven has been used to gather the most frequently used samples of malware and give labeled to the malware, but this malware repository has not been updated since 2010. In this manner, malware detectors need new malware samples. As we have found some research, repositories used for the formation of malware are Kaggle, Microsoft, Malware, contagio, the Zoo. The major problem is that the Malware Samples are not labeled with a particular database. Although certain internet systems can mark samples of malware such as VirusTotal and Hybrid Malware Analysis are often utilized. Labeling the malware samples one at a time is a time-intensive procedure, on the other hand. Malware samples from different categories may also be hard to include on a level playing field. A honeypot may also be used to acquire livemalware. Some of the suggested methods utilized honeypots to acquire payloads for malware in real-time, although most research documents do not disclose this. In addition, we may acquire malware samples of our suggested model from antivirus firms. This would be an excellent method of testing malware classifiers with actual malware samples.

6.4. Malware Characteristics Selection

The scalability of the suggested approach is a major issue. It mainly depends on the set of features in which benign and malicious files are classified. With the addition of features, the precision is improved, but the time to scan is increased, restricting malware detection systems in real-time applications. Malware selection is mainly used to distinguish between malware and benign files. Many studies have used API calls, PE headers, and file network activity.

6.5. Machine-Learning Algorithmic Frameworks Selection

Following a comprehensive examination of numerous methodologies, all kinds of machine learning algorithms were applied. Ensemble produces better accuracy than simple classification algorithms (SVM, DT, KNN), but these algorithms take more training time than simple algorithms. Classification methods for Machine Learning may be selected depending on malware size and variety. Another important factor in the accuracy of malware categorization is the parameterization of the algorithms that are used. This issue has not been addressed in depth by the approach that has been proposed. it is very important factor for the accuracy of the malware classifier.

6.6. Future Directives

The challenge now is how one can build a classification of malware that can deal with these problems. We understand now that malware analysts evaluate malware samples and continue to update the malware detection system to block attacks malware. Currently it's almost impossible for signature-based technique to detect new malware. However, the behavioral malware detection technique gives hope for detecting new malware. Therefore, we shall design a hybrid framework, not like hybrid methodologies previously offered. For implementing the suggested method, a two-layered architecture will be utilized. At first, signature-based malware detection is conducted, which, if it fails, will be utilized in second-level Behavior based analytical methods. In the first layer known and basic unbuffered malware may be readily detected, while in the dynamic analysis the malware can be predicted using runtime. When each new virus occurs, the database will be updated and utilized to anticipate future malware. Fig. 8 shows the framework of the malware detection system. This approach collects runtime characteristics by using both sandboxing automation (e.g., Cuckoo sandbox) and a variety of dynamical analysis tools, including Ollydbg, Regshot, Wireshark, and ProcMon, for collecting run time data. Furthermore, tools such as PExplorer, Peview, Peid, and IDA pro are used to extract static features such as strings, imports, and exports from a program. The algorithms for learning machines are then used for malware classification training, which is a kind of machine learning. Using runtime behavior, this approach has the potential to inherit the benefits of both signature-based and behavioral methods. It can successfully identify existing malware while also detecting new malware. We are going to use anti-obfuscation techniques to correctly analyses malware samples. It can be done only after the use of static and dynamic malware analyses to check for malware and benign samples. The approach that has been described is intended to bridge the gap that exists between signature and behavioral methods. Real-time malware detection will be made feasible by the development of a two-layer hybrid technique, which will be implemented in real-time. In addition, various algorithms are used to enhance the resistance of the hybrid model against the adversarial machine learning. The various types of malwares and their description is included in Table 11. A chronological examination of several well-known malwares is presented in Table 12.

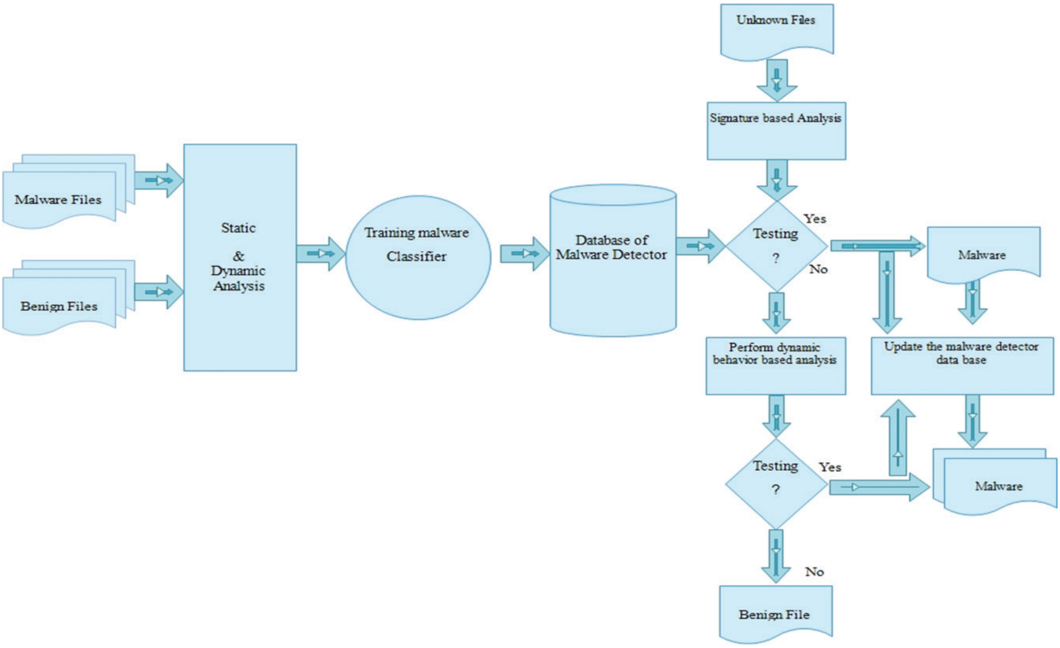

Fig. 8. The Proposed schematics of Hybrid Malware Detection Technique.

Table 11: Types of Malwares

Types of malwares	Description	Examples
Virus	More technologically speaking, a computer virus is a harmful code or software intended to alter the way a computer function and to propagate it from one computer to the next. A virus inserts or attaches to a legitimate application or document supporting macros to run its code. A virus may have unintentional or negative consequences in this process by corrupting or deleting information, such as damaging system software.	Common warrior, Creeper, Eliza, Elk cloner, and the Chernobyl Virus.
Worm	A computer worm is a kind of malware, which can be copied and extended across computers. Without human involvement, a worm may reproduce itself and does not need an association with a software application to do damage.	The Storm worm, SQL slammer, The Morris worm, Jerusalem worm, Dabber, and the Code Red II.
Trojan Horse	A Trojan horse is a kind of malware that masquerades as legitimate software, and it is also known as a trojan horse. Hackers and cyber thieves may enter a computer system from Trojans. Social engineering is sometimes used to convince people on their computers to download and execute Trojans.	I love you, Code red, Melissa, Sasser, Zeus, and Conficker.
Spyware	Spyware refers to a kind of software that is designed to steal personal or business information. It is done by performing a sequence of actions without the required user rights, and in certain cases, even in plain view of the user. Advertising, data collection, and changing the computer's user configuration settings are all common actions performed by spyware.	Coolwebsearch (CWS), Gator, Transponder, BlazeFind, Hot as Hell, and ISTbar.
Rootkit	A rootkit is hidden computer software that retains privileged computer access while shielding the presence of a machine. The phrase 'rootkit' consists of the combination of the term's 'root' and 'kit.' A rootkit was originally a collection of tools that enabled administrators to manage a computer or network.	Soni BMG Copy protection Rootkit, NT Rootkit.
Adware	Adware is a program that shows unwanted ads or advertising-supported applications. When pop-up advertisements are shown, your browser's homepage is altered, Spyware is installed, and pop-up advertisements are blasted, Adware programs will bombard your device with advertising. Adware is a term used to describe potentially harmful software.	1080 Solution Assistant, Altnet, Cool web search, Ads by Game Vance,
Bot	A malicious is a kind of harmful malware that infects the host system and establishes a connection with a central server. The server is used as a command and the control center is a botnet or network of infected computers and other devices.	Earth link spammer, cut wail, Storm, Grum, Kraken.
Ransomware	Ransomware is a malware kind that encrypts victims' data. In return for restoring access to data, the attacker then demands a ransom from the victim.	Wannacry, Bad Rabbit, Ryuk, Troldesh, Jigsaw.

Table 12: A chronological examination of several well-known malwares

Year	Malware Attacks
1986	First IBM-PC brain Sector Virus W
1987	The Jerusalem virus was found in Jerusalem and all executables on computers were infected and destroyed after it had just begun Friday the 13th.
1988	A virus of Ping Pong Boot sector was founded at Turin University in Italy.
1989	There is a Trojan AIDS. It requested urgent payment to be dropped.
1990	The chameleon virus was the first polymorphic virus to be created.

1999	An e-mail worm known as Happy99 emerges, hides modifications, and wants the computer user a good New Year. A new version is available. Outlook Express and Internet Explorer were impacted by Windows 95 and 98.
2000	More than a million PCs were infected with "I love you" dubbed as a love bug.
2001	Bad Trans was created to steal credit card information as well as passwords.
2003	Microsoft defects make spreading simple for Agobot and Bolgimo computer worms.
2004	Hacks may access the hard drive of the infected machines using the MyDoom (also Novang), the fastest mail and file-sharing computer worm.
2005	Cellular phone virus, Commwarrior-A propagated via text messages from mobile phones to mobile phones.
2006	The first malware to attack Mac OS X arrived as the low-threat Trojan called OSX/Leap-A.
2007	A Trojan horse called Zeus used a method called keystroke logging to steal bank sensitive information.
2008	The Koobface virus targets people who use MySpace and Facebook.
2009	In the United States and South Korea W32.Dozer Follows a serious cyber-attack.
2010	A Kenzero virus spreads the history of the browser online.
2011	Zeus and SpyEye have joined together to create a new method of attacking mobile phones to collect financial information.
2013	Cryptolocker one of the early ransomware programs crypto lockers had a large global impact and contributed to the rise of ransomware.
2014	Backoff malware infiltrates POS systems to steal information from credit cards.
2016	Cerber One of the most powerful ransomware threats. It's also one of the most common forms of crypto malware. Cerber infected more enterprise PCs than any other ransomware family at one point, according to Microsoft.
2017	WannaCry ransomware almost affected 150 countries including hospitals, banks, warehouses, telecommunication companies, and many other industries.
2018-2020	During that time, many crypto miners and ransomware, such as the COVID19 RAT, the Samsam ransomware, the cyborg ransomware, and the clop ransomware, were developed.

7. CONCLUSION

This study helps to identify malwares using runtime characteristics. It provides the knowledge fraction for harmful software analysis and detection procedures through machine learning algorithms. Malware is now well-known to be highly sophisticated and rapidly changing. Today, it is not only used to disturb users, steal information, and destroy user data but also to enforce its objectives for businesses and nations. The protection of data and resources is a key component in information technology. This is essential because computers or embedded digital devices in every area are utilized to execute activities more quickly and precisely without human involvement. The computer system is more susceptible to hacking because of its wide range of applications. This study covers the development of malware, present status, and techniques of detection. Two malware analysis techniques exist signature based, and behavior based. Signature-based technology has two major flaws that must be addressed. For the time being, signature-based techniques will not be able to identify new or previously unknown malware. Second, different types of malwares may readily outwit the system's detection mechanisms. Behavior-based technique can identify new variants of malware, dynamic techniques in malware-based methods are more robust than signature-based methods. The actual application of dynamic methods nevertheless remains rigid and time-intensive, whereas signature technology is quicker and more efficacious than dynamic techniques in identifying known malware. We spoke about malware methods that were suggested for machine learning to train the classification of malware in this research. This is because it includes a huge number of algorithms that may be used for a variety of malware features. In addition to its accessibility, machine learning algorithms provide many benefits over conventional malware classifications, such as the capacity to get information from file samples, rapidly detect, unexpected changes, and minimize the human work and time spent analyzing malware. In this article, malware detection methods are classified according to analytical methodologies such as static, dynamic, and hybrid techniques. Each method has advantages and disadvantages. These methods, however, produced encouraging results in the categorization of malware in a certain scenario. For example, static methods are quicker, with a reduced false-positive rate, but it is difficult to cope with the obfuscation technique while collecting static characteristics. In this respect, on the contrary, dynamic methods offer a beneficial although implementation in real-time is inconvenient. Finally, we addressed the two-layer malware detection framework with static and dynamic functions that can effectively and reliably detect the new and known malware.

REFERENCES

[1] H. Weijie, X. Jingfeng, W. Yong, H. Lu, K. Zixiao and M. Limin, “MalDAE: Detecting and explaining malware based on correlation and fusion of static and dynamic characteristics”, Computers and Security, vol. 83, pp. 208-233, 2019.

[2] B. Pete, F. Richard, T. Frederick and J. Kevin, “Malware classification using self organising feature maps and machine activity data”, Computers and Security, vol. 73, pp. 399-410, 2018.

[3] N. Usha, T. F. Di, C. V. Aaron, A. Thomas and S. Mark, “Clustering versus SVM for malware detection”, Journal of Computer Virology and Hacking Techniques, vol. 12, pp. 213-224, 2016.

[4] J. A. Namavar, H. Sattar, D. Ali and C. R. Kim kwang, “An improved two-hidden-layer extreme learning machine for malware hunting”, Computers and Security, vol. 89, pp. 10-16, 2020.

[5] F. Massimo and P. F. Leaf, “An open-source cybersecurity training platform for realistic edge-IoT scenarios”, Journal of Systems Architecture, vol. 97, pp. 107-129, 2020.

[6] K. Afreen, Z. Swaleha and Al. S. Muaadhabdo, “An improved pre-processing machine learning approach for cross-sectional imaging of demented older adults”, International Conference of Intelligent Computing and Engineering (ICOICE), IEEE, pp. 1-7, 2019.

[7] P. Jakub, N. Q. Anh, B. Adrian, G. Jonathan and L. Y. Kubo, “a framework for automated efficacy testing of anti-virus behavioral detection with procedure-based malware emulation”, In Proceedings of the 13th International Workshop on Automating Test Case Design, Selection and Evaluation, pp. 37-44, 2022.

[8] B. Andrew, “How deception can change cyber security defenses”, Computer Fraud & Security, vol.5, no.1, pp. 12-14, 2019.

[9] G. Ekta, B. Divya and S. Sanjeev, “Malware analysis and classification: A

survey", Journal of Information Security, vol. 6, 2014.

[10] M. S. Mariam, S. M. Ali, Z. Si-Jing and Y. Hong-Ji, "Conceivable security risks and authentication techniques for smart devices: A comparative evaluation of security practices", International journal of Automation and Computing, vol. 13, pp. 350-363, 2016.

[11] B. Kang, F. Liu, Z. Yun and Y. Liang, "Design of an Internet of Things-based smart home system", International Conference on Intelligent Control and Information Processing, IEEE, vol. 2, pp. 921-924, 2011.

[12] A. Shahid, H, R. Nigel, T. Issa and S. Ibrahim, "A framework for metamorphic malware analysis and real-time detection", Computers and Security, vol. 48, pp. 212-233, 2015.

[13] H. Xin, "Large-scale malware analysis, detection, and signature generation", Doctoral dissertation, University of Michigan, 2011.

[14] R. Kalpika, A. R. Vasudevan, "Detection of Zeus Bot Based on Host and Network Activities", Communications in Computer and Information Science, vol. 746, pp. 978-981, 2017.

[15] E. Nabeil, E. Rashad, H. Alzubair and L. Fagen, "A blockchain-based attribute-based signcryption scheme to secure data sharing in the cloud", Journal of Systems Architecture, vol. 102, pp. 10-16, 2020.

[16] J. Daehee, J. Yunjong , L Sungman , P. Minjoon, K. Kuenhwan , K. Donguk and K. B. Byunghoon, "Rethinking anti-emulation techniques for large-scale software deployment", Computers and Security, vol. 83, pp. 182-200, 2019.

[17] M. Samaneh and G. Ali A, "Application of deep learning to cybersecurity: A survey", Neurocomputing, vol. 347, pp. 149-176, 2019.

[18] Z. Weizhe, W. Huanran, H. Hui and L. Peng," DAMBA: detecting android malware by ORGB analysis", IEEE Transactions on Reliability, vol.69, no.1, pp. 55-69, 2020.

[19] L. Liu, W. Bao-sheng, Y. Bo and Z. Qiu-xi, "Automatic malware classification and new malware detection using machine learning", Frontiers of Information Technology & Electronic Engineering, vol. 18, no. 9, pp. 1336-1347, 2017.

[20] K. Ratinder and M. Singh, "Hybrid real-time zero-day malware analysis and reporting system", International Journal of Information Technology and Computer Sciences, vol. 8, pp. 63-73, 2016.

[21] M. Jelena, M. Miroslaw and F. Alberto, "Time, accuracy and power consumption tradeoff in mobile malware detection systems", Computers and Security, vol. 82, pp. 314-328, 2019.

[22] S. Hudan, S. Ferdous and P. Christian, "A survey on forensic investigation of operating system logs", Digital Investigation, vol. 29, pp. 1-20, 2019.

[23] N. Bruce, K. K. Hwan, K. Y. Jin, K.H. Ho, K. T. Yong and L. H. Jae, "Cross-method-based analysis and classification of malicious behavior by api calls extraction", Applied Sciences, vol. 9, no.2, pp. 239, 2019.

[24] Z. Hanqi, X. Xi, M. Francesco, N. Shiguang, M. Fabio and S. A. Kumar, "Classification of ransomware families with machine learning based on N-gram of opcodes", Future Generation Com-

puter Systems, vol. 90, pp. 211-221, 2019.

[25] N. Jose, P. Araujo, P. Donald M and R. C. Ghedini, "MULTS: A multi-cloud fault-tolerant architecture to manage transient servers in cloud computing", Journal of Systems Architecture, vol. 101, pp. 101-108, 2019.

[26] R. Veeramani and R. Nitin, "Windows api based malware detection and framework analysis", In International Conference on Networks and Cyber Security, vol. 25, 2012.

[27] C. Mihai, J. Somesh, K. Johannes, K. Stefan and V. Helmut, "Software transformations to improve malware detection", Journal in Computer Virology, vol. 3, pp. 253-265, 2007.

[28] O. Yoshihiro, "Trends of anti-analysis operations of malwares observed in API call logs", Journal of Computer Virology and Hacking Techniques, vol. 14, no. 1, pp. 69-85, 2018.

[29] C. Sibi S and V. V. Sangeetha, "A survey on malware analysis and mitigation techniques", Computer Science Review, vol. 32, pp. 1-23, 2019.

[30] D. Jaime, S. Igor, C. Xabier, P. Yoseba K and B. Pablo G, "Automatic behaviour-based analysis and classification system for malware detection", International Conference on Enterprise Information Systems, vol. 2, pp. 395-399, 2010.

[31] D. Yuxin, X. Xiaoling, C. Sheng and L. Ye, "A malware detection method based on family behavior graph", Computers and Security, vol. 73, pp. 73-86, 2018.

[32] S. Jagsir and S. Jaswinder, "A survey on machine learning-based malware detection in executable files", Journal of Systems Architecture, vol. 112, pp. 10-18, 2021.

[33] R. Anusmita and N. Asoke, "Introduction to Malware and Malware Analysis: A brief overview", International Journal, vol. 4, no.10, 2016.

[34] W. Huanran, H. Hui and Z. Weizhe, "Demadroid: Object reference graph-based malware detection in Android", Security and Communication Networks, vol. 2018, 2018.

[35] B. Tao, T. Takeshi, G. Shanqing, I. Daisuke and N. Koji, "Integration of multi-modal features for android malware detection using linear SVM", In 2016 11th Asia Joint Conference on Information Security (Asia JCIS), IEEE, pp. 141-146, 2016.

[36] B. Ulrich, K. Engin and K. Christopher, "Improving the efficiency of dynamic malware analysis", In Proceedings of the 2010 ACM Symposium on Applied Computing, pp. 1871-1878, 2010.

[37] W. Gerard, S. Radu and D. Alexandre, "Malware behaviour analysis", Journal in Computer Virology, vol. 4, pp. 279-287, 2008.

[38] M. Asit and T. Shashikala, "Virtual machine introspection: towards bridging the semantic gap", Journal of Cloud Computing, vol. 3, no.1, pp. 1-14, 2014.

[39] V. Jiri and P. Martin, "Virtualization of operating system using type-2 Hypervisor", In Software Engineering Perspectives and Application in Intelligent Systems: Proceedings of the 5th Computer Science On-line Conference 2016 (CSOC2016), Springer International Publishing, vol. 25, pp. 239-247, 2016.

[40] M. Andreas, K. Christopher and K. Engin, “Exploring multiple execution paths for malware analysis”, In 2007 IEEE Symposium on Security and Privacy (SP'07), IEEE, pp.231-245, 2007.

[41] K. Bojan, Z. Apostolis, L. Tamas, W. George and E. Claudia, “Adaptive semantics-aware malware classification”, In Detection of Intrusions and Malware, and Vulnerability Assessment: 13th International Conference, DIMVA 2016, San Sebastián, Spain, Springer International Publishing, vol. 13, pp. 419-439, 2016.

[42] R. Edward, Z. Richard, C. Russell, S. Jared, Y. Paul, W. Rebecca, T. Anna, M. Mark and N. Charles, “An investigation of byte n-gram features for malware classification”, Journal of Computer Virology and Hacking Techniques, vol. 14, pp. 1-20, 2018.

[43] M. Zane and B. Michael, “Building a machine learning classifier for malware detection”, In 2014 second workshop on anti-malware testing research (Water), IEEE, pp. 1-4, 2014.

[44] S. Michael and H. Andrew, “A Practical malware analysis: the hands-on guide to dissecting malicious software”. Starch Press, 2012.

[45] W. Tzu-Yen and W. Chin-Hsiung, “Detection of packed executables using support vector machines”, International Conference on Machine Learning and Cybernetics, IEEE, vol. 2, pp. 717-722, 2011.

[46] A. Satheesh and R. Kumaravelu, “A mathematical model of HMST model on malware static analysis”, International Journal of Information Security and Privacy, vol. 13, no. 2, pp. 86-103, 2019.

[47] A. Imad and L. Saiida, “A new classification based model for malicious PE files detection”, International Journal of Computer Network and Information Security, vol. 9, no.6, pp.1-7, 2019.

[48] L. Michael, A. Steven, H. Blake, R. Matthew, “Malware Analyst's Cookbook and DVD”, Wiley Publishing, 2011.

[49] L. Xiaojing, Y. Kan, W. XiaoFeng, L. Zhou, X. Luyi and B. Raheem, “Acing the ioc game: Toward automatic discovery and analysis of open-source cyber threat intelligence”, ACM SIGSAC conference on computer and communications security, pp. 755-766, 2016.

[50] S. Sebastian and K. Stefan, “Code obfuscation against static and dynamic reverse engineering”, In Information Hiding: 13th International Conference, IH 2011, Prague, Czech Republic, Revised Selected Papers, Springer Berlin Heidelberg, vol. 13, pp. 270-284, 2011.

[51] C. Michael , “Scanning memory with Yara”, Digital Investigation, vol. 20, pp. 34-38, 2017.

[52] S. Nikolaos, B. Chafika, A. Omar and A. Ameer, “Forensic malware analysis: The value of fuzzy hashing algorithms in identifying similarities”, IEEE Trustcom/ Big Data SE/ ISPA, IEEE, pp. 1782-1787, 2016.

[53] K. Abhishek, G. Suchandra and G. Ratan, “Detecting obfuscated viruses using cosine similarity analysis”, Asia International Conference on Modelling and Simulation (AMS'07), IEEE, pp. 165-170, 2007.

[54] B. Danilo, M. Lorenzo, M. Mattia,

"Code normalization for self-mutating malware", IEEE Security and Privacy, vol. 5, no.2, pp. 46-54, 2007.

[55] Z. Boyun, Y. Jianping, H. Jingbo, Z. Dingxing and W. Shulin, "Malicious codes detection based on ensemble learning", In Autonomic and Trusted Computing: 4th International Conference, ATC 2007, Hong Kong, China, July 11-13, 2007. Proceedings 4, Springer Berlin Heidelberg, pp. 468-477, 2007.

[56] M. Robert, F. Clint, T. Nir, B. Eugene, G. Marina, D. Shlomi and E. Yuval, "Unknown malcode detection using opcode representation", In Intelligence and Security Informatics: First European Conference, Euro ISI 2008, Esbjerg, Denmark, Proceedings, Springer Berlin Heidelberg, pp. 204-215, 2008.

[57] B. S. Mojtaba, J. Saeed and T. Asghar, "PbMMD: A novel policy based multi-process malware detection", Engineering Applications of Artificial Intelligence, vol. 60, pp. 57-70, 2017.

[58] N. Vivens, X. Zhifeng, M. V. Rao, M. Ke and X. Yang, "Network forensics analysis using Wireshark", International Journal of Security and Networks, vol. 10, no.2, pp. 91-106, 2015.

[59] H. Nazrul, B. Monowar H, B. Ram Charan, B. Dhruba K and K. Jugal K, "Network attacks: Taxonomy, tools and systems", Journal of Network and Computer Applications, vol. 40, pp. 307-324, 2014.

[60] E. Eldad, "Reversing: secrets of reverse engineering", John Wiley & Sons, 2011.

[61] G. Daniel, M. Carles and P. Jordi, "The rise of machine learning for detection and classification of malware: Research developments, trends and challenges", Journal of Network and Computer Applications, vol. 153, pp. 102526, 2020.

[62] R. Chathuranga and J. Aruna, "An efficient approach for advanced malware analysis using memory forensic technique", In 2017 IEEE Trustcom/ Big Data SE/ ICESS, IEEE, pp. 1145-1150, 2017.

[63] K. Ilker, "A basic malware analysis method", Computer Fraud and Security, vol. 2019, no. 6, pp. 11-19, 2019.

[64] K. Joakim, "Fundamentals of Digital Forensics", Springer International Publishing, 2020.

[65] P. Radu, H. Steven, L. Thor, S. Matija, P. Jens and C. Alexandre, "Analysis of malware behavior: Type classification using machine learning", International conference on cyber situational awareness, data analytics and assessment (Cyber SA), IEEE, pp. 1-7, 2015.

[66] A. Omer and S. Refik, "Investigation of possibilities to detect malware using existing tools", 14th International Conference on Computer Systems and Applications. IEEE, pp. 1277-1284, 2017.

[67] M. Q. Ali, A. Irfan and Y. Muhammad, "Cloud Intell: An intelligent malware detection system", Future Generation Computer Systems, vol. 86, pp. 1042-1053, 2018.

[68] G. Kent, S. Scott, H. Xin and C. Tzi-cker, "Automatic generation of string signatures for malware detection", Recent Advances in Intrusion Detection: 12th International Symposium, RAID 2009, Saint-Malo, France, Proceedings, Springer Berlin Heidelberg, vol. 12, pp.

101-120, 2009.

[69] G. Dragoş, C. Mihai, A. Dan and C. Liviu, "Malware detection using machine learning", International multi-conference on computer science and information technology, IEEE, pp. 735-741, 2009.

[70] B. Philippe, G. Isabelle and M. Jean-Yves, "Behavior abstraction in malware analysis", In Runtime Verification: First International Conference, RV 2010, Proceedings, Springer Berlin Heidelberg, vol. 1, pp. 168-182, 2010.

[71] C. Sang Kil, M. Iulian, J. Jiyong, T. John, B. David and A. David G, "Split Screen: Enabling efficient, distributed malware detection", Journal of Communications and Networks, vol. 13, no.2, pp. 187-200, 2011.

[72] S. Asaf, M, Robert, F. Clint, D. Shlomi and E. Yuval, "Detecting unknown malicious code by applying classification techniques on opcode patterns", Security Informatics, vol. 1, no.1, pp. 1-22, 2012.

[73] S. Asaf, M. Robert, E. Yuval and G. Chanan, "Detection of malicious code by applying machine learning classifiers on static features: A state-of-the-art survey", Information Security Technical Report, vol. 14, no.1, pp. 16-29, 2009.

[74] E. A. Ahmed E, M. M. Aizaini and B. Bazara, "Improving the detection of malware behaviour using simplified data dependent API call graph", International Journal of Security and its Applications, vol. 7, no. 5, pp. 29-42, 2013.

[75] P. Bassir, J. M. Vafaie and J. Mehrdad, "Malware detection using hidden Markov model based on Markov blanket feature selection method". International congress on technology, communication and knowledge, IEEE, pp. 558-563, 2015.

[76] S. Nedim and L. Pavel, "Hidost: a static machine-learning-based detector of malicious files". EURASIP Journal on Information Security, vol. 2016, pp. 1-20, 2016.

[77] K. Dong Hee, W. Sang Uk, L. Dong Kyu and C. Tai Myoung, "Static detection of malware and benign executable using machine learning algorithm". Eighth International Conference on Evolving Internet, pp. 14-19, 2016.

[78] H. Shamsul, A. Jemal, A. Mamoun, A. Mali, I. Rafiqul and Y. John, "Hybrids of support vector machine wrapper and filter based framework for malware detection". Future Generation Computer Systems, vol. 55, pp. 376-390, 2016.

[79] N. Yuta and U. Ryuya, "Static analysis with paragraph vector for malware detection". In Proceedings of the 11th International Conference on Ubiquitous Information Management and Communication, pp. 1-7, 2017.

[80] W. Cheng, Q. Zheng, Z. Jixin and Y. Hui, "A malware variants detection methodology with an opcode based feature method and a fast density based clustering algorithm". 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), IEEE, pp. 481-487, 2016.

[81] M. Aziz, A. Omar and M. Manar, "AMAL: high-fidelity, behavior-based automated malware analysis and classification". Computers and Security, vol. 52, pp. 251-266, 2015.

[82] Y. Yanfang, W. Dingding, L. Tao, Y.

Dongyi and J. Qingshan, "An intelligent PE-malware detection system based on association mining", Journal in computer virology, vol. 4, pp. 323-334, 2008.

[83] B. Michael, O. Jon, A. Jon, M. Z. Morley, J. Farnam and N. Jose, "Automated classification and analysis of internet malware", Recent Advances in Intrusion Detection: 10th International Symposium, RAID, vol. 10, pp. 178-197, 2007.

[84] R. Konrad, T. Philipp, W. Carsten and H. Thorsten, "Automatic analysis of malware behavior using machine learning", Journal of Computer Security, vol. 19, no.4, pp. 639-668, 2011.

[85] V. Mihai, G. Laura and T. Nicolae, "Practical malware analysis based on sandboxing", Ro Edu Net Conference 13th Edition: Networking in Education and Research Joint Event RENAM 8th Conference, IEEE, pp. 1-6, 2014.

[86] H. Jozsef, M. Yoan, I. Alexander and L. Amaury, "Methodology for behavioral-based malware analysis and detection using random projections and k-nearest neighbors classifiers", International Conference on Computational Intelligence and Security, IEEE, pp. 1016-1023, 2011.

[87] G. Mahboobe, S. Ashkan and S. Zahra, "Dynamic VSA: a framework for malware detection based on register contents", Engineering Applications of Artificial Intelligence, vol. 44, pp. 111-122, 2015.

[88] P. Zhi-Peng, F. Chao and T. Chao-Jing, "Malware classification based on the behavior analysis and back propagation neural network". In ITM Web of Conferences, EDP Sciences, vol.7, pp. 20-28, Nov. 2016.

[89] H. Shamsul, M. Suruz, H. M. Mehedi, I. Rafiqul, Y. John, A. Majed and A. Ahmad, "Defending unknown attacks on cyber-physical systems by semi-supervised approach and available unlabeled data", Information Sciences, vol. 379, pp. 211-228, 2017.

[90] M. Fahad, H. Wei and B. Antony, "Novel malware detection methods by using LCS and LCSS", 22nd International Conference on Automation and Computing (ICAC), IEEE, pp. 554-559, 2016.

[91] H. Eduardo, M. Rubén S and L. Ignacio M, "Evaluating the reliability of computational grids from the end user's point of view", Journal of Systems Architecture, vol. 52, no. 12, pp. 727-736, 2006.

[92] N. B. Narayanan, D. Ouboti and K. Temesguen, "Performance analysis of machine learning and pattern recognition algorithms for malware classification", IEEE national aerospace and electronics conference (NAECON) and ohio innovation summit (OIS), IEEE, pp. 338-342, 2016.

[93] C. In Kyeom, K. T. Guen, S. Y. Jin, R. Minsoo and I. E. Gyu, "Malware analysis and classification using sequence alignments", Intelligent Automation and Soft Computing, vol. 22, no.3, pp. 371-377, 2016.

[94] M. Jiang, X. Zhi, L. Pengwei, W. Dinghao, L. Peng and M. Bing, "Impeding behavior-based malware analysis via replacement attacks to malware specifications", Journal of Computer Virology and Hacking Techniques, vol. 13, pp.193-207, 2017.

[95] W. Markus, R. Alexander, T. Niklas and

A. Wolfgang, "A knowledge-assisted visual malware analysis system: Design, validation, and reflection of KAMAS", Computers and Security, vol. 67, pp. 1-15, 2017.

[96] N. Stavros D and P, Iosif, "A graph-based model for malware detection and classification using system-call groups", Journal of Computer Virology and Hacking Techniques, vol. 13, no.1, pp. 29-46, 2017.

[97] L. Quan, B. Oisin, M. N. Brian and S. Mark, "Deep learning at the shallow end: Malware classification for non-domain experts", Digital Investigation, vol. 26, pp. S118-S126, 2018.

[98] P. Abdurrahman and A. Tankut, "Classification of malware families based on runtime behaviors", Journal of information security and applications, vol. 37, pp. 91-100, 2017.

[99] S. Jan, P. Tomás and R. Martin, "Multiple instance learning for malware classification", Expert Systems with Applications, vol. 93, pp. 346-357, 2018.

[100] G. Ibrahim, H. Mohammad, P. Vaclav, H. Liangxiu, H. Robert, R. Khaled, and A. Francisco J, "Detection of advanced persistent threat using machine-learning correlation analysis", Future Generation Computer Systems, vol. 89, pp. 349-359, 2018.

[101] G. Ibrahim, H. Mohammad, P. Vaclav, H. Liangxiu, H. Robert, R. Khaled, and A. Francisco J, "Detection of advanced persistent threat using machine-learning correlation analysis", Future Generation Computer Systems, vol. 89, pp. 349-359, 2018.

[102] X. Lu, F. Jiang, X. Zhou, S. Yi and J. Sha and L. Pietro, "ASSCA: API sequence and statistics features combined architecture for malware detection", Computer Networks, vol. 157, pp. 99-111, 2019.

[103] D. Arivudainambi, K.A. Varun Kumar and P. Visu, "Malware traffic classification using principal component analysis and artificial neural network for extreme surveillance", Computer Communications, vol. 147, pp. 50-57, 2019.

[104] Y. Cagatay and K. Ahmet, "Imaging and evaluating the memory access for malware", Forensic Science International: Digital Investigation, vol. 32, pp. 20-27, 2020.

[105] R. Mahdi, W. Yong Li, K. Reza, J. Hamed, Z. Ruxin and H. Peng, "A hybrid machine learning approach for malicious behaviour detection and recognition in cloud computing", Journal of Network and Computer Applications, vol. 151, pp. 12-19, 2020.

[106] E. A. Ahmed E, M. M. Aizaini, B. Bazara- IA and H. Hentabli, "Enhancing the detection of metamorphic malware using call graphs", Computers and Security, vol. 46, pp. 62-78, 2014.

[107] R. Jesse C, K. Roger I, L. Scott M and C. Robert K, "Detection of injected, dynamically generated, and obfuscated malicious code", ACM workshop on Rapid malcode, pp. 76-82, 2003.

[108] C. Michael, "A protocol graph based anomaly detection system", Doctoral dissertation, Carnegie Mellon University, 2008.

[109] M. R. Jose and D. J. Cesar, "Integrating static and dynamic malware analysis using machine learning", IEEE Latin America Transactions, vol. 13, no. 9, pp. 3080-3087, 2015.

[110] S. PV and S. AJPCS, “Integrated static and dynamic analysis for malware detection”, Procedia Computer Science, vol. 46, pp. 804-811, 2015.

[111] E. E. Inang, B. Chafika, A. Ameer and W. Paul, “Analysis of malware behaviour: Using data mining clustering techniques to support forensics investigation”, Fifth Cybercrime and Trustworthy Computing Conference, IEEE, pp. 54-63, 2014.

[112] W. Ahsan, I. Azhar, L. Jahanzaib, N. Ahsan and B. Anas, “A novel approach of unprivileged keylogger detection”, 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), IEEE, pp. 1-6, 2019.

[113] N. Mohammad, A. Nouman and Y. Jing Tao, “A three-way decision making approach to malware analysis using probabilistic rough sets”, Information Sciences, vol. 37, no. 4, pp. 193-209, 2016.

[114] N. Mohammad, A. Nouman and Y. Jing Tao, “Detecting malware evolution using support vector machines”, Expert Systems with Applications, vol. 143, pp. 113022, 2020.

[115] P. Avi, R. Brian, K. Lee, H. Michael, C. Catherine, O. Alison, T. Glenn, R. S. Neal, P. Terry, and T. Jason, “Artificial intelligence based malware analysis”, arXiv preprint, pp.16-23, 2017.

[116] H. Shamsul, I. Rafiqul, A. Jemal, Y. John, H. M. Mehedi and F. Giancarlo, “A hybrid-multi filter-wrapper framework to identify run-time behaviour for fast malware detection”, Future Generation Computer Systems, vol. 1, no. 83, pp. 193-207, 2018.

[117] S. Robert, X. Lifan, K. William, V. Tristan, F. Teague, H. John, P. Zachary, S. Corey, S. Joshua and C. John, “Parallelization of machine learning applied to call graphs of binaries for malware detection”, 25th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP), IEEE, pp. 69-77, 2017.

[118] K. Youngjoon, K. Eunjin and K. Huy Kang, “A novel approach to detect malware based on API call sequence analysis”, International Journal of Distributed Sensor Networks, vol. 11, no.6, pp. 659101, 2015.

[119] M. Weixuan, C. Zhongmin, T. Don, F. Qian and G. Xiaohong, “Security importance assessment for system objects and malware detection”, Computers and Security, vol. 68, pp. 47-68, 2017.

[120] O. Philip, S. Sezer, and K. McLaughlin, “Detecting obfuscated malware using reduced opcode set and optimised runtime trace”, Security Informatics, vol. 5, pp. 1-12, 2016.

[121] D. Anusha, T. Fabio Di, V. C. Aaron, A. Thomas H and S. Mark, “A comparison of static, dynamic, and hybrid analysis for malware detection”, Journal of Computer Virology and Hacking Techniques, vol. 13, pp. 1-12, 2017.

Aftab *et al. (IJECI)* 2024

International Journal for
Electronic Crime Investigation

DOI: https://doi.org/10.54692/ijeci.2024.0801186

(IJECI)
ISSN: 2522-3429 (Print)
ISSN: 2616-6003 (Online)

Research Article

Vol. 8 issue 1 Jan-Mar 2024

Online shopping, Cyber frauds and Fraud prevention Strategies

Aftab Ahmad Malik[1], Waqar Azeem[2] and Mujtaba Asad[3]

[1]Department of Computer Science, University of Engineering and Technology Lahore.
[2]Faculty of Computer Science, South Eastern Regional College, Down Patrick Ireland, United Kingdom.
[3]Department of Automation and Control, Shanghai Jiao Tong University, Shanghai, China.
Corresponding author: dr_aftab_malik@yahoo.com

Received: December 22, 2023; **Accepted:** January 10, 2024; **Published:** March 15, 2024

ABSTRACT

Online shopping is increasingly being targeted by hackers and cyber criminals, who exploit the anonymity of the internet to deceive unsuspecting shoppers. These scams involve fake websites or ads, posing as legitimate sellers, damaging innocent citizens' bank accounts and databases, and causing damage to customers. Online shopping fraud involves using stolen credit or debit cards for purchases, while identity theft involves stealing personal information for fraudulent purposes like credit or illegal purchases. We discuss, the safety tips to avoid online shopping scams, using these safety tips before making a purchase. In United states a government agency FTC has been entrusted the task of implementation the civil law related to anti-trust; it also indorses and promotes the protection of consumers rights while working with Justice Department. Online shopping scams were the second most common fraud category in 2021, according to the FTC. To avoid them, use safety tips to identify and avoid scams. Cybercriminals steal and can use personal information to make unauthorized purchases or engage in fraudulent activities. Identity theft is a crime involving capturing and the misuse of another's personal identifying information like Id-card, credit card and bank account information. Fraudsters often use stolen credit cards to purchase items, return them for refunds, and then sell the refunded money or goods. Machine learning is a rapidly evolving technology that can significantly enhance online shopping security and user awareness which is coupled with artificial intelligence

Keywords: Cyber threats, Cyber frauds, Common Scams, Forgery, Identity theft.

1. INTRODUCTION

The consumers should be aware of potential scams in online shopping, including common types of frauds spreading in society; to protect themselves from potential scams such as Account Takeovers, Fake Websites and Stores, Identity Theft, Payment Frauds, Phishing Scams, Refund Fraud, Social Engineering and Unsecure Wi-Fi Networks. Online shopping fraud can be minimized by adopting a proactive approach, shopping from reputable websites, and checking for reviews and customer feedback before making a purchase to ensure a secure transaction. Secure websites use encrypted connections, ensuring transactions are processed securely and efficiently. We must double-check the website's legitimacy by verifying the URL, checking for contact information, and examining spelling errors or unusual design elements. Online shopping dodges involve fraudulent activities where criminals pretend to be genuine sellers through fake websites or advertisements on genuine venders' sites. Scammers exploit the anonymity of the internet to create fake online retailers, using advanced technology, sophisticated designs, stolen logos, and domain names to steal unsuspecting shoppers from legitimate online sellers. Websites often sell luxury items like clothing, jewelry, and electronics at low prices, but customers may receive fake or nothing at all.

2. USE OF SOCIAL MEDIA AND ELECTRONIC DEVISES

Online shopping scams are increasingly using social media to set up fake online stores, often selling counterfeit clothing or jewelry. These scammers advertise their fake websites on social media, so trusting a site based on its appearance is not enough. To detect these scams, search for reviews before making a purchase. shopping.

According to [1], it is recommended that electronic devices and software can be very effectively used for detection and Control of the offences of online frauds, white collar crimes, cybercrimes, hacking of others information. According to [2], the frauds in Banking and entrepreneurs may be eradicated and minimizes by implementing software for Network Security with electronic devices and creating a demilitarized Zone in Network consisting of two fire walls, first is software firewall and the second a hardware firewall. The role of legislation and need of strong Legal Framework and Procedures has been discussed in [3]. According [4] to Technology has significantly impacted human life, particularly in the realm of buying and selling. The internet has significantly facilitated this process, offering more convenient and less stressful options. Online shopping and e-commerce have enabled consumers to access products from distant stores, eliminating distance and long queues. The paper [5] recommends to contest effectively with Cybercrime and Money Laundering; and discusses the threats and consumer perceptions of online shopping, emphasizing the need for awareness on cyber security issues. It provides tips on protecting users and merchants from data breaches and attacks, including phishing and adware.

3. HOW TO AVOID FRAUDS REGARDING ONLINE SHOPPING?

The research paper [6], stresses and worry about the requirement of calibration of forensic evidence its attaining preservation and presentation in court. It further advises the using FBI techniques must be used by FIA Pakistan. Online shopping allows consumers to access products from distant stores without long queues. However, it is susceptible to threats, such as data breaches and security compromises, making consumers uncertain about their trust in online shopping is convenient but crucial to avoid scams. By following these tips, you can identify scam signs and feel more confident in your online shopping experience. It's better to be cautious and risk losing deals than losing money. Report any scams to your bank and the Federal Trade Commission. Frequently occurring Bank Frauds Using Digital Devices in Pakistan and the Role of Business Ethics has been elaborated and expounded in [7]. Some safety measures are given below:

- Avoid sellers contending on immediate payments via apps like Zelle®, Venmo, Cash App, or online wire transfers of money.
- Avoid sellers using pressure tactics for immediate purchases.
- Avoid sharing sensitive information like bank account number, PIN, or access code.
- Be cautious of social media ads leading to unfamiliar sites.
- Be doubtful of deals with low prices.
- Be watchful of fake websites and secure websites.
- Research the seller and use the term "scam" before sending money.
- Set up account alerts for unusual activity.
- Trust your instinct and walk away if something feels off.
- Use credit cards, especially if unfamiliar with the seller.

4. THE LEGAL PROVISIONS ON ONLINE FRAUDS

Fraud in Pakistan can occur at anytime, anywhere, and is defined as "fraudulently" under section 25 PPC, if done with intent to defraud. A fraud occurs when someone deceives you into parting with your property under false pretenses, including cash, personal information such as property car, house, phone number, or consent to use the information. There are few legal instruments, we would like to refer on the subject matter like “fraud in Pakistan on online shopping “.

- Consumer Protection Laws
- Electronic Transactions Ordinance, 2002,
- Intellectual Property Laws
- Procedures of Federal Investigation Agency (FIA),
- Pakistan Penal Code PPC relevant Clauses.

The Electronic Transactions Ordinance in Pakistan establishes a legal framework for electronic transactions, including online shopping, and acknowledges electronic documents and digital signatures. Pakistan has consumer protection laws to protect consumer rights and address fraud in online transactions. The Pakistan Penal Code includes provisions for cheating, criminal breach of trust, and forgery, which may apply in online shopping fraud cases. The FIA in Pakistan investigates and combats cybercrimes, including online fraud, and may use specific laws to protect brand owners' rights in cases of counterfeit or pirated goods involved in online shopping. According to [8], the internet has transformed the shopping experience, offering convenience and less stress.

5. THE PROCEDURE FOR TRIAL IN COURT

For the offences related to Frauds and Cyber

Crime is identical with other crimes as shown in the steps given below.

- Arrest warrant
- Arrest
- Initial appearance
- Investigation
- Identification of suspect
- Decision of charging the offender
- Court proceedings of the trial
- Fines, probation, or imprisonment
- Sentencing

The suspect is initially seen by a judge, where charges are read and bail may be set. Prosecutors review evidence and decide whether to file formal charges based on the crime's severity. A judge initially sees the suspect, reads charges, and sets bail. Prosecutors review evidence and decide on filing formal charges based on the crime's severity. Jurisdictional laws and procedures for online shopping-related crimes can vary, and the nature and severity of the crime can impact arrest steps. Legal representation is crucial for protecting individuals' rights during legal proceedings. The court determines the appropriate punishment for a suspect's conviction, which may include fines, probation, or imprisonment.

6. ONLINE SHOPPING FRAUDS IN PAKISTAN

According to [8], legal concerns arise when dealing with online shopping fraud in Pakistan. It's advisable to consult with a legal professional or contact relevant authorities like the Federal Investigation Agency for guidance on current legal provisions and actions. Pakistan's Payment System and Electronic Fund Transfers Act, 2007 establishes a legal framework for online payment transactions. According to [9], Online shopping offers convenience and reduced stress compared to traditional methods, eliminating the need for physical stores and allowing more informed product selection. However, it is susceptible to malicious attacks that compromise data safety, security, and integrity. According to [10], technology developments have significantly impacted business operations, forcing companies to adapt to global dynamics and customer demands. Understanding the roles of technologies in constructing innovative business models, particularly in ecommerce, is crucial. A survey based on secondary research results and an evaluation matrix were developed to summarize technologies and identify gaps in current research. A study conducted and reported in [11] assesses the consumer perceptions and attitudes towards healthy and environmentally friendly foods, focusing on reducing overconsumption, low-nutrient consumption, plant-based diets, and food waste, highlighting the importance of multilevel policies for promoting sustainable food practices.

According to [12] the media industry is experiencing rapid shifts in consumer spending, with consumers shifting towards digital services and media products. Gartner reports show a gradual increase in IT services, making the industry the third-largest IT spender. The study aims to investigate research on Information Systems (IS) in the media industry, particularly in management and economic areas.

In Pakistani FIR is required to be lodged with Police about the occurrence of fraud cases and if the matter is related to online or significant cybercrime, to the FIA, The FIR initiates a criminal investigation, requiring evidence at every stage. Attach screenshots, private conversations, or websites used for fraud. Pakistan has implemented the Payment Systems and Electronic Fund Transfers Act, 2007 to establish and operate Payment System Operators and Providers for online payment systems. Its purpose is to regulate payment systems and electronic fund transfers, providing consumer protection standards and determining financial institutions' rights and liabilities, ensuring efficient and secure financial transactions. Mostly the online shopping fraud cases can be minimized using knowledge of academic databases like PubMed, IEEE Explore, Science Direct, JSTOR, or Google Scholar, and search for relevant papers using keywords like "online shopping fraud", "e-commerce security" and "consumer protection." The FIA advises individuals facing fraud to seek legal assistance, inform relevant organizations and relevant agencies, lodge an FIR and effectively pursue the case. The authors of [13], use online product reviews to extract consumer brand associations and their interconnections, utilizing a network-based understanding of brand image. This approach measures brand image, using consumer-generated content which is tested in an empirical study.

7. RECOMMENDATIONS

i. To ensure online safety, avoid making purchases on unsecured Wi-Fi networks and use a VPN for added security.

ii. Maintain secure online accounts by shopping from reputable websites, using unique passwords.

iii. keep devices and antivirus software updated, being cautious of unauthorized emails, and

iv. Regularly review bank and credit card statements. Before making a purchase, review the seller's ratings and feedback, especially those with a negative track record.

v. Choose secure payment methods like credit cards or reputable online services, and avoid wire transfers or payment methods with limited dispute resolution options.

vi. Immediately report alleged fraudulent activity to online platforms, banks, and authorities.

vii. Arrange caution and verify website legitimacy before making online purchases.

viii. Ensure that the website uses secure and encrypted connections, specifically "https://" in the URL, when processing payments.

ix. Secure websites utilize encrypted connections to ensure efficient and secure transaction processing.

x. Enable “multi-factor authentication” whenever possible to enhance the security of your online accounts.

xi. The suspicious online links must be

ignored and not do not download the attachments of doubtful links and don’t download anything from unwanted emails or messages, particularly those requesting for personal or financial information.

xii. It is advisable to consult with a legal professional or the Federal Investigation Agency for guidance on current legal provisions and actions. The legal landscape may change, and enforcement effectiveness may vary.

xiii. Research on consumer protection laws and cybersecurity measures in online shopping is being conducted to address fraud and ensure consumer safety. This study will enable to explore the implementation of cybersecurity measures to prevent online shopping fraud.

xiv. Exploration of recently developed methods and having knowledge about the detection of frauds in the processes of e-commerce is indeed very important as well as the role of electronic technology for the prevention of online frauds.

xv. Machine learning is a rapidly evolving technology that can significantly enhance online shopping security and user awareness which is coupled with artificial intelligence.

xvi. To ensure a fair and reliable online shopping experience, find out first the genuineness of that website/social media page that it has announced a just refund or returns policy, detailed complaint handling processes, and a clear understanding of the company you are dealing with.

xvii. It is recommended that the research published in “Computers & Security," "Journal of Cybersecurity," " and "Journal of Consumer Affairs" is very effective and notable. Therefor the reader must look into it.

xviii. Familiarize yourself with online store refund and return policies, and be cautious if unclear. Implementing these practices can reduce fraud risk and create a safer shopping experience.

xix. According to [4], Secure online payments with a secure service, 'https' URL, and padlock symbol. Verify URLs and consider virtual currencies like bitcoin, as they lack protections and cannot be refunded once sent.

8. CONCLUSION

Online shopping revolutionizes the shopping experience by eliminating long queues and limited options, allowing for more informed selection of preferred models or products without frequent store searches, thereby enhancing the overall shopping experience. Phishing and adware attacks have led to numerous data breaches affecting reputable firms like Amazon and Google. Online shopping offers convenience and reduced stress, but it also exposes users to malicious attacks that compromise data safety and integrity. Online shopping faces threats from fraud and lack of proper information. Enhancing knowledge on cyber security threats can enhance security and reduce negative perceptions, making it a more appealing option for consumers. Online shopping has faced numerous data breaches due to cyber security threats, including phishing and adware techniques. This has led to reluctance among consumers to adopt online shopping due to concerns about fraud and lack of proper information. Enhancing cyber security awareness can help mitigate these risks.

9. ACKNOWLEDGEMENT

The authors express gratitude for Mr Kaukab Jamal Zuberi and Dr Syed Mona Hassan, Chief Editor for encouragements.

REFERENCES

[1] A. A. Malik, M. Asad and W. Azeem, "Detection and Control over the offences of White Collar Crime, Fraud and Hacking of information by using effectively the relevant software and Electronic Devices", International Journal for Electronic Crime Investigation, vol. 7, no. 1, pp. 1-8, 2023.

[2] A. A. Malik, M. Asad and W. Azeem, "Frauds in Banking and entrepreneurs by electronic devices and combating using Software and employment of demilitarized zone in the Networking", International Journal for Electronic Crime Investigation, vol. 6, no. 4, pp. 1-8, 2022.

[3] A. A. Malik, W. Azeem and M. Asad, "Role of Legislation, need of strong Legal Framework and Procedures to Contest Effectively with Cybercrime and Money Laundering", International Journal of Crimes investigation, vol. 6, no. 2, pp. 1-8, 2022.

[4] A. A. Malik, W. Azeem and M. Asad, "Requirement of strong legal frame work and procedures to contest with Cybercrime in Pandemic Situation", International Journal for Electronic Crimes Investigation, vol 5, no. 1, pp. 3-12, 2021.

[5] A. A. Malik, M. Asad and W. Azeem, "To Combat White Collar Crimes in Public and Private Sector and Need for Strong Legislation and Ethics", International Journal for Electronic Crimes Investigation, vol. 4, no. 3, pp.1-8, 2020.

[6] A. A. Malik, "Standardization of forensic evidence its procurement preservation and presentation in court of using FBI techniques by FIA", International Journal for Electronic Crimes Investigation, vol. 4, no. 1, pp. 1-6, 2020.

[7] A. A. Malik, "Bank Frauds Using Digital Devices and the Role of Business Ethics", International Journal for Electronic Crimes Investigation, vol. 2, no. 4, pp. 1-9, 2020.

[8] Pakistan's Payment System and Electronic Fund Transfers Act, 2007.

[9] A. Aseri, "Security issues for online

shoppers", International Journal of Scientific and Technology Research, vol. 10, no. 3, 2021.

[10] U. Doloto and Y. H. Chen-Burger, "A Survey of Business Models in eCommerce, In Agent and Multi-Agent Systems: Technologies and Applications", 9th KES International Conference, KES-AMSTA, pp. 249-259, 2015.

[11] A. Hoek, D. Pearson, S. James, M. Lawrence and S. Friel, "Shrinking the food-print: A qualitative study into consumer perceptions, experiences and attitudes towards healthy and environmentally friendly food behaviors", Appetite, pp. 117-131, 2017.

[12] A. Lugmayr and J. Grueblbauer, "Review of information systems research for media industry–recent advances, challenges, and introduction of information systems research in the media industry", Electronic Markets, vol. 27, no. 1, pp. 33-47, 2017.

[13] S. Gensler, F. Volckner, M. Egger, K. Fischbach and D. Schoder, "Listen to your customers: Insights into brand image using online consumer-generated product reviews", International Journal of Electronic Commerce, vol. 20, no. 1, pp. 112-141, 2015.

Shairoze *et al. (IJECI)* 2024

International Journal for
Electronic Crime Investigation

DOI: https://doi.org/10.54692/ijeci.2024.0801187

(IJECI)
ISSN: 2522-3429 (Print)
ISSN: 2616-6003 (Online)

Research Article
Vol. 8 issue 1 Jan-Mar 2024

IoT Malware: A Comprehensive Survey of Threats, Vulnerabilities, and Mitigation Strategies

Muhammad Shairoze Malik

Department of Information Technology, Superior University Lahore Pakistan
Corresponding author: msisw-f21-003@superior.edu.pk

Received: December 26, 2023; **Accepted:** January 13, 2024; **Published:** March 15, 2024

ABSTRACT

The proliferation of the Internet of Things (IoT) has ushered in a new era of connectivity and convenience, linking a vast array of devices from household appliances to industrial machinery. However, this interconnectivity also introduces significant security vulnerabilities, making IoT systems attractive targets for malicious actors. This comprehensive survey delves into the multifaceted world of IoT malware, exploring the evolving landscape of threats that plague these systems. We methodically analyze various types of IoT malware, identifying common attack vectors and the intrinsic vulnerabilities that IoT devices often possess. These vulnerabilities range from inadequate security protocols to the use of default credentials and unpatched software. Furthermore, the paper highlights real-world instances where IoT devices have been compromised, leading to significant disruptions and breaches of privacy. In addressing these challenges, we outline an array of mitigation strategies. These strategies include but are not limited to, enhanced encryption methods, regular firmware updates, network segmentation, and the adoption of robust authentication mechanisms. We also discuss the role of machine learning and artificial intelligence in predicting and preventing IoT malware attacks. Moreover, our survey extends to the regulatory and ethical considerations surrounding IoT security, advocating for a more proactive approach in standard-setting and compliance enforcement. The findings of this study aim to serve as a foundational resource for researchers, cybersecurity professionals, and policymakers, emphasizing the need for a collective and informed effort in fortifying the IoT ecosystem against the ever-growing threat of malware.

Keywords: Internet of Things (IoT), IoT Security, Malware Analysis, Cyber Threats, Network Security, Cybersecurity Policies.

1. INTRODUCTION

The advent of the Internet of Things (IoT) has transformed the way we interact with technology, seamlessly integrating it into every facet of our daily lives. From smart home devices to industrial automation systems, the IoT ecosystem has expanded rapidly, offering unprecedented levels of connectivity and convenience. However, this burgeoning network of interconnected devices also presents a significant security challenge. The proliferation of IoT devices has been paralleled by an increase in the complexity and frequency of cyber-attacks, with IoT malware emerging as a critical threat to both individual privacy and global infrastructure [1].

This paper embarks on a comprehensive survey of the threats posed by IoT malware, shedding light on the vulnerabilities inherent in these connected systems. It seeks to understand the nature of these threats, categorizing the different types of malwares specifically designed to exploit IoT devices. These threats are not only diverse in their mechanisms but also in their targets, ranging from consumer devices to critical infrastructure. In exploring these vulnerabilities, the study highlights key areas where IoT devices fall short in terms of security. Common issues include, but are not limited to, inadequate default configurations, lack of regular updates, and poor network security practices. The implications of these vulnerabilities are vast, affecting not just individual device security but also the safety and reliability of entire IoT networks [2].

Recognizing the urgency of addressing these challenges, the paper delves into a range of mitigation strategies. These strategies encompass both technical solutions, such as enhanced encryption and network segmentation, and policy-driven approaches like the development of rigorous security standards and frameworks for IoT devices. The objective of this survey is multifaceted: to provide a comprehensive overview of the current threat landscape in IoT cybersecurity, to identify and analyze common vulnerabilities in IoT systems, and to propose effective strategies for mitigating these risks. By doing so, the paper aims to contribute to the ongoing discourse in cybersecurity, offering valuable insights for researchers, practitioners, and policymakers involved in the field of IoT [2].

2. IOT MALWARE: AN OVERVIEW

The concept of malware, malicious software designed to disrupt, damage, or gain unauthorized access to computer systems, takes on a new dimension in the context of the Internet of Things (IoT). IoT malware refers to a variety of malicious software specifically crafted to target IoT devices, which include a wide range of internet-connected gadgets, appliances, and industrial equipment. This section provides an overview of the landscape of IoT malware, discussing its characteristics, types, and the reasons behind its rising prominence [3].

2.1. Evolution of IoT Malware

The evolution of IoT malware can be traced back to the early days of internet connectivity, where the primary targets were computers and servers. However, as IoT devices began to proliferate, these became the new frontier for cyber-attacksClick or tap here to enter text.. The evolution is marked by several high-profile incidents, such as the Mirai botnet attack in 2016, which highlighted the vulnerabilities in IoT devices and the potential for large-scale disruption [3].

2.2. Characteristics of IoT Malware

IoT malware is distinguished from traditional malware in several key aspects:

- Target Diversity: Unlike conventional malware that typically targets computers and servers, IoT malware can infect a wide range of devices, from smart thermostats to industrial control systems.

- Propagation Methods: IoT malware often exploits basic security flaws, such as default passwords and unpatched software, to propagate rapidly across networks.
- Stealth and Persistence: Due to the often-limited security features on many IoT devices, malware can remain undetected for extended periods.
- Functionality: The functionality of IoT malware varies, including data theft, forming botnets, or causing physical damage by hijacking device controls.

2.3. Impact of IoT Malware

The impact of IoT malware extends beyond the compromised devices themselves, posing broader implications:

- Privacy Concerns: IoT devices often collect sensitive personal data, making them prime targets for cybercriminals looking to steal personal information.
- Infrastructure Disruption: Malware-infected IoT devices in critical infrastructure can lead to significant disruptions, including utility outages or compromised industrial operations.
- Economic and Social Implications: The economic cost of IoT malware attacks can be substantial, affecting businesses and consumers alike. Moreover, the erosion of trust in IoT technology can have long-term social implications.

2.4. Attack Vectors and Vulnerabilities

The susceptibility of IoT devices to malware is largely due to a combination of diverse attack vectors and inherent vulnerabilities. This section dissects the various pathways through which IoT devices can be compromised and the systemic weaknesses that exacerbate these risks [4].

2.5. Common Attack Vectors

This subsection should outline the primary methods used by attackers to infiltrate IoT systems:

- Default Credentials: Many IoT devices come with default usernames and passwords, which are often left unchanged by users, making them easy targets.
- Unpatched Software: IoT devices with outdated firmware are vulnerable to exploits targeting known security flaws.
- Network Eavesdropping: Unsecured wireless communication can allow attackers to intercept data and gain unauthorized access.
- Physical Tampering: In some cases, physical access to IoT devices can enable the installation of malicious firmware or software.

2.6. Inherent Vulnerabilities in IoT Devices

This subsection should discuss the common vulnerabilities inherent in many IoT devices:

- Limited Processing Power and Memory: Constraints on computational resources can limit the ability of IoT devices to employ advanced security measures.
- Lack of Standardized Security Protocols: The IoT landscape is characterized by a lack of unified security standards, leading to inconsistent security practices.
- Insecure Interfaces: Web interfaces, APIs,

and mobile interfaces often lack robust authentication and encryption, presenting another attack surface.

2.7. Exploiting Device Connectivity

This part should explore how the interconnected nature of IoT devices can amplify vulnerabilities:

- Propagation Within Networks: Once one device is compromised, malware can quickly spread to other connected devices within the network.
- Lateral Movement: Attackers can leverage compromised IoT devices as a foothold to infiltrate more secure networks and systems [5].

3. IMPACT OF IOT MALWARE

The proliferation of IoT malware not only poses a significant security risk but also has far-reaching implications across various domains. This section provides an in-depth analysis of the impact of IoT malware, considering both the direct and indirect consequences of such security breaches.

3.1. Personal and Privacy Impact

- Data Theft and Privacy Breaches: Discuss how IoT malware can lead to the unauthorized access and theft of personal data, compromising individual privacy.
- Home Network Infiltration: Elaborate on how compromised IoT devices can serve as entry points to broader home networks, endangering personal information stored on other devices.

3.2. Economic Consequences

- Financial Losses for Businesses and Consumers: Analyze the financial impact, including the costs of mitigating malware infections, potential fines for data breaches, and loss of consumer trust[5].
- Downtime and Productivity Loss: Explore how malware attacks can lead to operational downtime for businesses, resulting in significant productivity and financial losses.

3.3. Impact on Infrastructure and Services

- Disruption of Critical Infrastructure: Examine cases where IoT malware has disrupted essential services (e.g., electricity, water supply, transportation systems).
- Compromise of Industrial Control Systems: Discuss the implications of IoT malware in industrial settings, including potential hazards and operational disruptions.

3.4. Social and Ethical Implications

- Erosion of Trust in IoT Technology: Discuss how recurring malware incidents can lead to public mistrust in IoT technologies and hinder their adoption.
- Ethical Concerns: Address ethical issues related to data privacy and security in the IoT ecosystem.

4. DETECTION AND ANALYSIS OF IOT MALWARE

In the dynamic landscape of cybersecurity, the detection and analysis of IoT malware stand as pivotal elements in the fight against cyber threats. As Internet of Things (IoT) devices increasingly permeate our daily lives, from smart home appliances to industrial control systems, the need for sophisticated methods to detect and analyze malware in these devices has never been more pressing [6].

4.1. Challenge of IoT Malware Detection

Detecting malware in IoT devices poses unique challenges. Unlike traditional computing environments, IoT ecosystems are diverse, with devices often differing in operating systems, processing power, and functionality. This heterogeneity makes applying uniform security measures or malware detection techniques challenging. Additionally, the limited computational resources of many IoT devices restrict the implementation of complex security software [6].

4.2. Advancements in IoT Malware Detection Techniques

Despite these challenges, significant advancements have been made in the detection of IoT malware. One approach involves network behavior analysis. Since IoT devices typically exhibit predictable network behaviors, any deviation from this pattern could indicate a compromise. By monitoring network traffic for anomalies, such as unusual outbound connections or spikes in data transmission, potentially malicious activities can be flagged. Another innovative approach is the use of honeypots – decoy systems designed to attract attackers. Honeypots mimic IoT devices and can be used to study attack methods and the behavior of IoT malware in a controlled environment. This information is invaluable for understanding and mitigating new threats [7].

5. MACHINE LEARNING: A GAME CHANGER

Machine learning is increasingly being recognized as a game-changer in the detection of IoT malware. By training algorithms on datasets of normal device behavior and known malware signatures, machine learning models can learn to identify potential threats. These models can adapt to new and evolving malware, offering a dynamic solution to the ever-changing threat landscape [8].

5.1. IoT Malware Analysis

IoT Malware Analysis is a critical and complex facet of cybersecurity, necessitated by the unique and diverse nature of the Internet of Things (IoT) ecosystem. It involves a multifaceted approach to understand and mitigate threats posed by malware specifically targeting IoT devices. Key strategies include static analysis, which examines the malware's code without execution to identify malicious signatures, and dynamic analysis, where malware is observed in a controlled environment to understand its behavior and interaction with other systems. Network traffic analysis is also pivotal, given the interconnected nature of IoT devices, to detect unusual patterns indicative of malware activity. Additionally, reverse engineering plays a significant role in deconstructing the malware to understand its components and functionalities. However, IoT malware analysis is challenged by the diversity of device architectures and the rapid evolution of threats, which often outpace defensive measures. To address these challenges, there's a growing reliance on artificial intelligence (AI) and machine learning to automate analysis processes, identify new malware types, and predict future trends, underscoring the need for continuous advancement and adaptation in IoT cybersecurity methodologies [6].

5.2. Techniques for analyzing IoT malware

Analyzing IoT malware involves specialized techniques tailored to the unique characteristics and constraints of Internet of Things (IoT) devices. Given the diversity and often resource-limited nature of these devices, traditional malware analysis methods used for PCs or servers may not always be applicable. Here are key techniques used in the analysis of IoT malware:

5.2.1. Network Traffic Analysis

Since IoT devices frequently communicate over networks, analyzing network traffic can reveal a lot about malware activity. Unusual data flows or communications to suspicious IP addresses can be indicators of compromise. This method is particularly effective in identifying malware that uses network vectors for propagation or command and control [3].

5.2.2. Firmware Analysis

Many IoT devices operate using firmware. Analyzing the firmware for vulnerabilities or embedded malware is a critical aspect of IoT malware analysis. This can involve extracting the firmware from the device and scrutinizing it for potential backdoors or malicious code. Machine Learning and AI Techniques: Advanced techniques using machine learning and artificial intelligence are increasingly employed in IoT malware analysis. These tools can automate the detection of malware patterns, analyze large volumes of data for anomalies, and even predict and identify new, unknown types of malware based on learned data patterns [9].

6. CHALLENGES IN DETECTION AND ANALYSIS

The detection and analysis of IoT malware present several challenges, stemming from the unique characteristics of IoT devices and the complexity of the IoT ecosystem. These challenges require innovative approaches and solutions to ensure effective cybersecurity. Key challenges include:

Diversity and Fragmentation of Devices: The IoT ecosystem is incredibly diverse, encompassing a wide range of devices with different operating systems, hardware capabilities, and functions. This fragmentation makes it difficult to develop uniform security protocols and malware detection systems that are effective across all devices.

Limited Processing Power and Memory: Many IoT devices are designed with minimal processing power and memory to keep costs low and optimize efficiency. This limitation restricts the ability to run complex security software or conduct in-depth, real-time analysis of potential threats.

Lack of Standardization: There is a lack of standardization in IoT device security, leading to inconsistencies in how devices are protected and how malware is detected. This lack of uniformity can create security gaps and make comprehensive protection challenging.

Evolving Nature of Threats: IoT malware is continuously evolving, with attackers constantly developing new methods to exploit vulnerabilities. This rapid evolution makes it difficult for security measures to keep pace and requires ongoing research and adaptation.

Scale and Scope of Networks: IoT devices are often deployed in large networks, increasing the potential attack surface. The extensive interconnectivity can also lead to widespread impacts of malware infections, as malware can quickly propagate across the network.

Data Privacy Concerns: The detection and analysis of IoT malware often involve monitoring network traffic and device behavior, which can raise data privacy concerns. Balancing effective malware detection with the privacy rights of users is a delicate and challenging task.

Resource Constraints for Security Updates: Ensuring that IoT devices are regularly updated with security patches is a challenge, particularly for older devices or those deployed in hard-to-reach locations. There may also be resource constraints in terms of bandwidth and downtime, which can hinder regular updates [10].

7. PREVENTION AND MITIGATION STRATEGIES

In the realm of IoT cybersecurity, the significance of robust prevention and mitigation strategies against malware cannot be overstated. The landscape of IoT devices, marked by their diversity and widespread application, presents unique challenges, making the implementation of comprehensive security measures both crucial and complex. At the forefront of these strategies is the integration of security into the design and development phase of IoT devices. This proactive approach not only embeds essential security features from the outset but also facilitates regular security audits to identify and address vulnerabilities early on. Equally important is the establishment of strong authentication protocols and access controls. Implementing multi-factor authentication and role-based access significantly enhances security by mitigating the risks associated with unauthorized access. In the context of network security, the segmentation of IoT devices plays a critical role. By isolating these devices on separate network segments, the potential spread of malware can be significantly curtailed, effectively reducing the overall attack surface. Complementing this, the deployment of firewalls and intrusion detection systems offers an additional layer of defense, monitoring and controlling the traffic to and from IoT devices [11].

Another cornerstone of IoT security is the rigorous management of software updates and patches. Regular firmware updates are essential to address emerging security vulnerabilities, and automated patch management systems ensure these updates are consistently applied. Furthermore, the encryption of data, both in transit and at rest, coupled with data integrity checks, safeguards sensitive information against interception and tampering [12]. However, technical solutions alone are not sufficient. Employee training and awareness programs are paramount in creating a culture of security mindfulness. Educating employees about the risks and best practices, especially in the context of phishing attacks, equips them with the knowledge to act as the first line of defense against potential breaches. In the event of a security incident, a well-crafted incident response plan is invaluable. Such a plan should outline clear steps for containment, eradication, and recovery, and be reinforced through regular drills and simulations to ensure preparedness and efficacy [13].

Lastly, the advent of advanced technologies like AI and machine learning is revolutionizing IoT malware detection and response. These technologies offer predictive capabilities and enhanced detection mechanisms, further fortifying the security posture [14]. Additionally, the exploration of blockchain technology for device authentication and data integrity is emerging as a promising avenue in enhancing IoT network security [15].

8. CONCLUSION

In the rapidly evolving landscape of IoT cybersecurity, the comprehensive survey conducted in this paper sheds light on the multifaceted challenges posed by IoT malware. The proliferation of interconnected devices, marked by their diversity and widespread adoption, has given rise to a complex security ecosystem, where the stakes are high, and the consequences of a security breach can be severe. Throughout this paper, we have delved into the intricacies of IoT malware, examining its various forms, attack vectors, and vulnerabilities that make IoT devices susceptible to compromise. We have explored the profound impact of IoT malware, from personal privacy violations to economic losses and disruptions of critical infrastructure, emphasizing the urgency of robust security measures.

Crucially, we have discussed the techniques and tools employed in the analysis of IoT malware, acknowledging the need for specialized approaches given the constraints of IoT devices. We have recognized the challenges that researchers and cybersecurity professionals face in this domain, from the diversity of devices to the rapid evolution of threats. In the sixth section, we outlined prevention and mitigation strategies that encompass secure design, robust authentication, network segmentation, and incident response planning, among others. These strategies are indispensable in safeguarding IoT ecosystems against malware and minimizing potential risks.

Looking forward, the integration of advanced technologies like AI, machine learning, and blockchain offers promising avenues to bolster IoT security, providing predictive capabilities and enhancing detection mechanisms. However, the ever-evolving nature of threats demands continuous adaptation and innovation. In conclusion, IoT malware represents a formidable challenge in the digital age, but it is one that can be addressed effectively with a combination of proactive measures, vigilant analysis, and ongoing collaboration among industry stakeholders, researchers, and policymakers. As IoT technology continues to advance and infiltrate various aspects of our lives, the importance of robust cybersecurity measures cannot be overstated. It is our hope that this comprehensive survey serves as a valuable resource in the ongoing effort to secure IoT ecosystems against the ever-present threat of malware.

REFERENCES

[[1] P. Dahiya, “Malware Detection in IoT,” Computer Networks. vol 20, no. 4, pp. 133–164. 2022.

[2] A. T. Salim and Ban Mohammed Kham-

mas, “Performance Evaluation of Deep Learning Techniques in The Detection of IOT Malware,” Iraqi Journal of Information and Communication Technology, vol. 6, no. 3, pp. 12-25, 2023.

[3] S. H. Olsen and T. OConnor, “Toward a Labeled Dataset of IoT Malware Features,” in 2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC), IEEE, pp. 924-933, 2023.

[4] B. I. Mukhtar, M. S. Elsayed, A. D. Jurcut, and M. A. Azer, “IoT Vulnerabilities and Attacks: SILEX Malware Case Study,” Symmetry (Basel), vol. 15, no. 11, pp. 1978-1198, 2023.

[5] K. Murakami, T. Kasama, and D. Inoue, “A Large-Scale Investigation into the Possibility of Malware Infection of IoT Devices with Weak Credentials,” IEICE Transactions on Information and Systems, vol. 106, no. 9, pp. 202-211, 2023.

[6] P. Victor, A. H. Lashkari, R. Lu, T. Sasi, P. Xiong, and S. Iqbal, “IoT malware: An attribute-based taxonomy, detection mechanisms and challenges,” Peer to Peer Networking and Applications, vol. 16, no. 3, pp. 1380-431, 2023.

[7] X. Zhu, J. Huang, and C. Qi, “Modeling and Analysis of Malware Propagation for IoT Heterogeneous Devices,” IEEE Systems Journal, vol. 17, no. 3, pp. 3846-3857, 2023.

[8] Y. Zi Wei, M. Md-Arshad, A. Abdul Samad, and N. Ithnin, “Comparing Malware Attack Detection using Machine Learning Techniques in IoT Network Traffic,” International Journal of Innovative Computing, vol. 13, no. 1, pp. 21-27, 2023.

[9] S. Kakati, D. Chouhan, A. Nag, and S. Panja, “Survey on Recent Malware Detection Techniques for IoT,” Lecture Notes in Electrical Engineering. vol, 2, no.3, pp. 647-659. 2022.

[10] J. Jeon, B. Jeong, S. Baek, and Y.-S. Jeong, “Static Multi Feature-Based Malware Detection Using Multi SPP-net in Smart IoT Environments,” IEEE Transactions on Information Forensics and Security, vol. 19, pp. 2487-2500, 2024.

[11] H. Alrubayyi, G. Goteng, and M. Jaber, “AIS for Malware Detection in a Realistic IoT System: Challenges and Opportunities,” Networking, vol. 3, no. 4, pp. 522-537, 2023.

[12] S. Sasikala and S. Janakiraman, “A Review on Machine Learning-based Malware Detection Techniques for Internet of Things (IoT) Environments,” Wireless Personal Communications, vol. 132, no. 3, pp. 1961-1974, 2023.

[13] A. A. Almazroi and N. Ayub, “Enhancing Smart IoT Malware Detection: A GhostNet-based Hybrid Approach,” Systems, vol. 11, no. 11, p. 547-554, 2023.

[14] S. Kasarapu, S. Shukla, and S. M. Pudu-

kotai Dinakarrao, "Resource- and Workload-Aware Model Parallelism-Inspired Novel Malware Detection for IoT Devices," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 42, no. 12, pp. 4618-4628, 2023.

[15] K. Nakao, "Mitigate: Toward Comprehensive Research and Development for Analyzing and Combating IoT Malware," IEICE Transactions on Information and Systems, vol. 106, no. 9, p. 20-29, 2023.

Zohaib *et al. (IJECI)* 2024

International Journal for
Electronic Crime Investigation

DOI: https://doi.org/10.54692/ijeci.2024.0801188

(IJECI)
ISSN: 2522-3429 (Print)
ISSN: 2616-6003 (Online)

Research Article

Vol. 8 issue 1 Jan-Apr 2024

Enhancement of Security and Privacy of Smart Contracts in Blockchain

Syed Khurram Hassan[1] and Muhammad Asif Ibrahim[2]

[1] Institute of Quality and Technology Management, University of the Punjab, Lahore, Pakistan.
[2]Department of Mathematics, The University of Lahore, Lahore.
Corresponding author: khuramshah6515@gmail.com

Received: December 29, 2023; **Accepted:** January 17, 2024; **Published:** March 15, 2024

ABSTRACT

Smart contracts, leveraging the power of blockchain technology, have revolutionized the execution and enforcement of agreements. However, their adoption also brings forth substantial challenges in terms of security and privacy. This research paper aims to identify the recent areas of focus and provide a comprehensive perspective on blockchain applications and smart contracts, highlighting their main issues and corresponding solutions. Furthermore, it seeks to address the gaps in current research and outline future avenues of investigation. The primary objective is to assess the security and privacy concerns associated with smart contracts in blockchain and propose effective measures to enhance their robustness. By conducting a thorough analysis of vulnerabilities, attack vectors, and privacy considerations, this study offers valuable insights into the risks involved in smart contracts. It also puts forth practical solutions and best practices to mitigate these risks, ensuring a more secure and privacy-preserving environment for the deployment and execution of smart contracts.

Keywords: Blockchain, privacy, security, technology, risks.

1. INTRODUCTION

The Smart contracts, driven by the innovative potential of blockchain technology, have emerged as a groundbreaking solution in the domain of decentralized digital transactions. In 2008, Satoshi Nakamoto introduced the concept of peer-to-peer cash transactions without the reliance on a centralized system. Blockchain, a digital ledger that publicly stores and verifies transactions through nodes, forms the foundation of this technology. The underlying structure of blockchain involves the validation of transactions by nodes and the security of these transactions through cryptographic hash functions. On the other hand, a smart contract represents a self-executing agreement encoded within a blockchain, enabling the automatic enforcement of contractual terms without intermediaries [1]. Essentially, it is a computer program composed of a set of rules that operate on the blockchain. By harnessing the transparent and immutable characteristics of blockchain, smart contracts provide heightened efficiency, security, and trustworthiness across various industries such as finance, supply chain management, and healthcare [2].

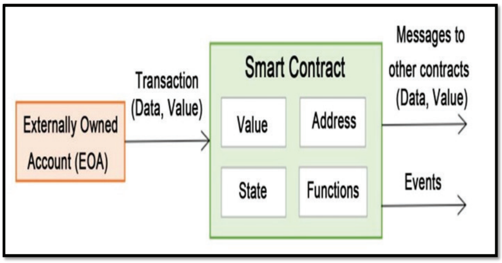
Externally Owned Account (EOA)
Transaction (Data, Value)
Smart Contract
Value
Address
State
Functions
Messages to other contracts (Data, Value)
Events

Fig 1: A basic structure of Smart Contract

The adoption of smart contracts has gained significant momentum, revolutionized traditional business processes, and enabled new forms of decentralized applications. A smart contract consists of the value, address, functions, and state. However, this rapid expansion brings forth a range of challenges and considerations, particularly in terms of security and privacy. As smart contracts increasingly handle sensitive data and control valuable assets, it becomes imperative to assess and address the security and privacy implications associated with their deployment and execution [3].

The research problem at hand revolves around the assessment of the security and privacy issues related to smart contracts in blockchain. It is crucial to identify and understand the vulnerabilities, risks, and potential attack vectors that can compromise the integrity, availability, and confidentiality of smart contracts. Furthermore, ensuring the privacy of contract participants and protecting their sensitive information from unauthorized access is of paramount importance. The objectives of this research article are twofold [4]. Firstly, it aims to conduct a comprehensive analysis of the security challenges faced by smart contracts. This analysis will encompass the identification and examination of vulnerabilities in smart contract code, potential attacks on contract execution, and governance-related risks. Additionally, it will explore the privacy concerns associated with smart contracts, including issues of identity and transaction privacy, as well as data confidentiality and access control [3].

Secondly, this research article seeks to propose solutions to enhance the security and privacy of smart contracts in blockchain. By leveraging best practices, auditing approaches, and secure execution environments, we aim to mitigate the identified security risks and vulnerabilities. Similarly, through the adoption of privacy-preserving cryptographic techniques and the exploration of privacy-focused blockchain protocols, we intend to enhance the privacy of smart contract transactions and safeguard sensitive data [5]. By addressing the security and privacy challenges in smart contracts, we can foster a more trustworthy and resilient ecosystem for their deployment and execution. This not only instills confidence in stakeholders but also encourages wider adoption of smart contracts, enabling their potential to be fully realized across diverse industries. The subsequent sections of this article will delve deeper into the identified research problem, analyzing security and privacy issues, and proposing practical solutions to enhance the robustness of smart contracts in blockchain [6].

2. BACKGROUND STUDY AND RELATED WORK

The section on background study and related work presents a comprehensive overview of prior research, studies, and advancements in the realm of enhancing the security and privacy of smart contracts in blockchain. The concept of smart contracts was initially proposed by Nick Szabo in 1994 [7]. This section aims to provide the necessary context for the current research article by highlighting the existing knowledge, identifying gaps, and showcasing the contributions made by previous studies in addressing the security and privacy challenges associated with smart contracts [8].

For instance, Hiroki Watanabe et al. [9] delve into the application of smart contracts in digital rights management and propose a consensus method. Ahmed Kosba et al. introduce Hawk, a cryptographic approach to writing secure smart contracts. Joshua Ellul and Gordon J. Pace [10] describe Alkyl VM, a virtual machine tailored for creating smart contracts in Internet of Things applications. The authors of [11] and [12] explore the execution process of smart contracts and highlight specific security issues. Through an examination of various articles related to smart contracts, it becomes evident that Solidity, a high-level language, is widely used for smart contract implementation [13].

Multiple blockchain platforms, including Ethereum, Eris DB, Zeppelin, and Counterparty, employ Solidity as their development language. Ethereum, in particular, utilizes a stack-based bytecode language for writing smart contract code, which is executed in the Ethereum Virtual Machine (EVM) [14].

3. SMART CONTRACT SECURITY RESEARCH

Previous research has extensively focused on identifying vulnerabilities and security flaws in smart contract code. Studies have analyzed common coding flaws and weaknesses, such as reentrancy, integer overflow/underflow, and unchecked external calls, to highlight the potential risks and propose best practices for secure smart contract development. Researchers have also explored automated tools and formal verification techniques to improve the security of smart contracts. These advancements aim to detect and prevent vulnerabilities during the development phase and provide rigorous analysis of smart contract code for potential flaws. Auditing and testing approaches have been investigated to assess the security of deployed smart contracts. Third-party audits, bug bounties, and code review processes have been employed to identify vulnerabilities and enhance the overall security of smart contracts [15].

4. HOW SMART CONTRACT WORKS IN BLOCKCHAIN

The Smart contracts are digitized contracts that are programmed using code and implemented on blockchain networks. They facilitate the exchange of assets in a transparent and conflict-free manner without the involvement of an intermediary. Once deployed on a blockchain, a smart contract serves as an automated agent that executes the terms of an agreement between parties. The code of a smart contract contains conditional statements that reflect the terms and outcomes of the agreement. For instance, if two parties agree to trade money for property, the smart contract can automatically transfer funds from one party to the other upon the transfer of the property [14].

Self-sufficiency: Once deployed, a smart contract independently enforces an agreement between parties without human intervention. It is difficult to tamper with or terminate.

Decentralization: Smart contracts are deployed on decentralized blockchains, so no single entity controls them. This makes them transparent and censorship-resistant.

Security: Blockchains are inherently secure due to cryptography. Hacking or tampering with a smart contract would require an enormous amount of computing power and is economically unfeasible in most cases.

Accuracy: Smart contracts are extremely precise because they are programmed using code. Vague terms in written legal contracts often lead to disputes, but smart contracts leave no room for ambiguity.

Trustlessness: Blockchains and smart contracts eliminate the need for a trusted third-party to oversee agreements and transactions. The parties do not need to know or trust each other to do business.

Immutability: Once deployed, a smart contract can never be modified or deleted. It is recorded permanently on the blockchain, ensuring the records and terms of the agreement cannot be altered [16]. While smart contracts provide many benefits, they also have some limitations. Because they cannot access data outside the blockchain, they cannot be used for agreements reliant on real-world data [17]. They also cannot be halted or reversed easily in the event of a hack or bug. Finally, current smart contract platforms lack mature, standardized programming languages [18].

5. ADVANTAGES OF SMART CONTRACTS

The use of smart contracts in blockchain technology offers several advantages. These include:

Transparency: Smart contracts are transparent, and the terms of the agreement are visible to all parties involved. This ensures that there is no room for ambiguity or misunderstanding.

Security: Smart contracts are executed automatically, and their execution is recorded on the blockchain network. This ensures that the agreement is tamper-proof, and the contract cannot be altered once it is executed.

Cost-Effective: The use of smart contracts eliminates the need for intermediaries, which reduces the cost of the transaction [19].

6. LIMITATIONS OF SMART CONTRACTS

While smart contracts offer several advantages, there are also some limitations to their use. These include:

Complexity: Smart contracts are complex, and their creation requires expertise in programming and blockchain technology.

Immutability: Once a smart contract is executed, it cannot be altered. This can be a disadvantage if there are errors in the code or if the terms of the agreement need to be changed [20].

Legal Validity: The legal validity of smart contracts is still a matter of debate, and there is no clear legal framework for their use.

In summary, smart contracts have the potential to streamline and secure many types of agreements in the future, but further progress is still required to address some of their existing limitations [21]. With continued improvement, smart contracts could transform how businesses and individuals conduct transactions and enable new blockchain-based business models [22].

7. REAL-LIFE APPLICATIONS OF BLOCKCHAIN

Blockchain technology has gained attention for its potential to revolutionize industries by providing secure, transparent, and immutable records. It enables decentralized consensus, allowing participants to trust and transact without intermediaries. While its origins lie in the realm of cryptocurrencies [18], blockchain's applications extend far beyond digital currencies. In this article, we explore real-life applications of blockchain in supply chain management, healthcare, and financial services, energy, voting, and education, intellectual property, and digital identity. Through these applications, blockchain is poised to transform industries by enabling transparency, traceability, and trust [23].

7.1. Supply Chain Management

The supply chain industry faces challenges related to transparency, traceability, and counterfeit products. Blockchain technology offers a solution by creating an auditable and immutable record of every transaction and movement within the supply chain. This ensures transparency and accountability throughout the entire process. The Food Trust initiative, a partnership between IBM and Walmart, demonstrates the potential of blockchain in enhancing food safety and traceability [24].

7.2. Healthcare

Blockchain holds promise for revolutionizing the healthcare industry by addressing issues such as data interoperability, patient privacy, and medical record management. Med Rec, developed by researchers at MIT, leverages blockchain to create a secure and auditable ledger of medical records. This approach empowers patients to control their own data while enabling secure sharing with healthcare providers. Blockchain-based medical record systems reduce errors, redundancies, and delays, leading to improved patient outcomes [24].

7.3. Financial Services

Financial services are among the early adopters of blockchain technology. Blockchain offers secure, transparent, and efficient solutions for transactions, identity management, and smart contracts. Ripple, a blockchain-based payment protocol, facilitates fast and low-cost cross-border transactions, bypassing intermediariesClick or tap here to enter text.. This technology reduces transaction fees, enhances speed, and increases financial inclusion, transforming traditional financial systems [25].

7.4. Blockchain in Energy

The energy sector can leverage blockchain to improve efficiency, transparency, and decentralized energy trading. Blockchain-based platforms enable peer-to-peer energy transactions, optimizing energy usage and reducing reliance on centralized authorities. The Brooklyn Microgrid project is an example of blockchain-powered energy trading, where participants can buy and sell excess solar energy in a secure and transparent manner [26].

7.5. Blockchain in Voting

Blockchain can introduce transparency, security, and trust in the voting process. By leveraging blockchain technology, the voting process can become tamper-proof, ensuring the integrity of the electoral system. Agora, a blockchain voting platform, successfully conducted a pilot project in Sierra Leone, allowing citizens to verify their votes and ensuring transparency in the electoral process [22].

7.6. Blockchain in Education

Blockchain technology can revolutionize the education sector by enabling secure and verified credentials, preventing fraud, and ensuring lifelong learner records. Blockchain-based platforms can authenticate degrees, certifications, and achievements, allowing individuals to have a portable and immutable record of their educational qualifications [24].

7.7. Blockchain in Intellectual Property

Blockchain provides a decentralized and transparent platform for protecting intellectual property rights. By recording transactions and ownership on a blockchain, creators can establish proof of ownership and protect their creations from infringement. The IP Chain project in Russia is an example of blockchain application in intellectual property management, streamlining the registration and protection of patents, trademarks, and copyrights [25].

7.8. Blockchain in Digital Identity

Blockchain technology can enhance digital identity management by providing secure, decentralized, and verifiable identities. Self-sovereign identity solutions allow individuals to control and manage their digital identities, reducing the risk of identity theft and enhancing privacy. Port, a blockchain-based identity platform, enables individuals to create and manage their digital identities, facilitating secure interactions in the digital world [26].

8. CHALLENGES AND LIMITA TIONS OF BLOCKCHAIN

Blockchain technology has gained significant attention as a decentralized, secure, and transparent system for managing digital transactions. However, the technology still faces several limitations and challenges that limit its widespread adoption. In this paper, we explore some of the key challenges and limitations of blockchain technology.

8.1. Scalability Limitations

The traditional proof-of-work consensus algorithm used by many blockchains limits their transaction processing capacity to a few dozen transactions per second. This is far below the processing capacity of centralized payment systems such as Visa or Mastercard. Several solutions have been proposed to address this issue, such as sharding and off-chain transactions. However, these solutions require consensus among the network participants and may compromise the security and decentralization of the blockchain [27].

8.2. Security and Privacy Concerns

Security and privacy concerns remain a significant challenge for blockchain adoption. Despite the use of cryptographic techniques to secure the blockchain, hackers have exploited vulnerabilities in smart contracts, wallets, and exchanges to steal millions of dollars' worth of cryptocurrencies. The lack of standardized security protocols and regulations also exposes blockchain users to legal and regulatory risks

[28].

9. INTEROPERABILITY AND GOVERNANCE CHALLENGES

The interoperability and governance of blockchain technology pose significant challenges. Different blockchains have different standards, protocols, and rules that limit their compatibility and coordination. The decentralized and distributed nature of blockchain technology also makes it difficult to establish a unified governance framework and ensure the accountability and transparency of its stakeholders [29].

9.1. Organizational Challenges

Lack of awareness and understanding about blockchain technology. Resistance to change and cultural barriers within organizations. High implementation costs and the need for specialized expertise. The requirement for large storage systems and increased computational power [28].

9.2. Governmental Challenges

Lack of laws and regulations addressing technical disputes and smart contracts. The need for standardization in blockchain applications. The importance of government support and regulatory frameworks for blockchain adoption [30].

9.3. Technical Challenges

Complexity of blockchain technology, making it difficult for users to understand. The increasing size of the blockchain, leading to performance degradation. Demands for substantial storage and computational resources, resulting in environmental concerns. Latency issues in adding verified blocks to the ledger. Integration challenges with legacy systems and the interoperability of different blockchains [31].

9.4. Energy Consumption

Bitcoin and other cryptocurrencies use blockchain technology for transactions. To validate transactions, a process called mining is performed using specialized hardware. However, this mining process consumes a lot of energy, leading to a significant carbon footprint. For example, Bitcoin mining consumes about 125 TWh of energy annually. This raises concerns about the environmental impact of cryptocurrencies. Efforts are being made to find energy-efficient solutions for blockchain technology [32].

9.5. Selfish Mining

In a selfish mining attack, an individual or group aims to maximize their own rewards by exploiting the mining process and causing other miners to waste their resources. Here's how it works in simpler terms: The attacker mines a block but keeps it private instead of sharing it with the network. This means that other miners are unaware of this block's existence. Meanwhile, the honest miners continue their efforts to validate a block, unaware that the attacker has already found one. This leads to a waste of their time and resources. The attacker continues to work on their private chain, trying to build a longer chain than the public one known to the rest of the network. By withholding their discovered blocks and secretly building their own chain, the attacker gains an advantage over honest miners. They can release their private chain at a strategic moment, causing the honest miners' efforts to go to waste, and potentially reaping more rewards for them. In essence, the selfish mining attack is a strategy where an individual or group manipulates the mining process to their advantage, causing other miners to waste resources while they maximize their own profits [31].

9.6. Personal Identifiable Information (PII)

There is a misconception surrounding the use of blockchain for identity management. Many believe that blockchain provides an ideal decentralized alternative to store Personal Identifiable Information (PII), replacing centralized databases. However, the reality is that blockchain can be used in two ways for PII: either by directly storing PII on the blockchain or by creating attestations on the blockchain that point to off-chain storage for PII. Elmagharaby and Losavio have delved into the concept of PII, specifically focusing on communication and location privacy [33].

10. FUTURE OF BLOCKCHAIN

Blockchain technology is a revolutionary way to securely record and store information. Initially introduced through the digital currency Bitcoin, blockchain has evolved to offer various applications beyond cryptocurrencies. This technology operates as a decentralized digital ledger that records transactions or information. Instead of being controlled by a single authority, such as a bank or government, it is maintained by a network of computers [34].

In blockchain, each transaction or piece of information is stored in a "block" and linked together in a "chain" of blocks, forming a complete record of all the transactions. The blockchain platform provides a distributed ledger that is scalable, secure, tamper-proof, and accessible by each peer on the network. Bitcoin, which operates on the blockchain, was created as a digital currency by an anonymous individual or group known as Satoshi Nakamoto [35]. Transactions on the Bitcoin blockchain are verified by network participants called "miners," who use powerful computers to solve complex mathematical problems [34]. The blockchain platform utilizes Public Key Cryptography Asymmetric Encryption algorithms, which involve public and private keys to encrypt and decrypt data. Public keys are used to encrypt messages, and private keys are necessary to decrypt them. Conversely, private keys are used to encrypt messages, and public keys are required for decryption [36].

Blockchain technology offers numerous benefits. It ensures transparency, as anyone can view the blockchain and verify transactions. The information stored on the blockchain is encrypted and resistant to alteration, enhancing security. Additionally, blockchain enables faster and more efficient transactions by eliminating the need for intermediaries. The potential of blockchain extends beyond Bitcoin. It can revolutionize various industries by increasing transparency and accountability. Blockchain's decentralized nature enhances security and transparency, making it valuable in supply chain management. It can track goods from manufacturers to end-users, ensuring authenticity and ethical production conditions.

The future of blockchain technology is promising. Beyond cryptocurrencies, blockchain can facilitate smart contracts, which are self-executing agreements with predefined conditions. It also enables secure transfer and storage of digital assets like property deeds and intellectual property rights. However, it is worth noting that public blockchain networks with a large number of nodes can experience limitations in transaction throughput and high latency, resulting in slower propagation of transactions and blocks [37].

11. CONCLUSION

Blockchain technology has evolved beyond cryptocurrencies, finding applications in various sectors. The examples discussed in this article illustrate how blockchain enhances transparency, security, and efficiency in supply chain management, healthcare, financial services, energy, government, and education. As blockchain continues to advance, it holds the potential to reshape industries, enabling trust, and transforming the way we conduct business and manage data. A number of endeavors have been made to improve consensus algorithms in blockchain, such as Peer Census, Kickstarter, and Chepurnoyet al.'s new consensus algorithm. Peer Census decouples block creation and transaction confirmation, while Kickstarter proposes the Greedy Heaviest-Observed Sub-Tree (GHOST) chain selection rule. Chepurnoyet al. proposed a new consensus algorithm for peer-to-peer blockchain systems where anyone who provides non-interactive proofs of retrievability is agreed to generate the block. The traditional proof-of-work consensus algorithm used by many blockchains limits their transaction processing capacity to a few dozen transactions per second. Security and privacy concerns remain a significant challenge for blockchain adoption, as hackers have exploited vulnerabilities in smart contracts, wallets, and exchanges to steal millions of dollars' worth of cryptocurrencies. The interoperability and governance of blockchain technology pose significant challenges, such as lack of awareness and understanding, resistance to change, high implementation costs, lack of laws and regulations, need for standardization, government support and regulatory frameworks, complexity of blockchain technology, increasing size of the blockchain, environmental concerns, and integration challenges with legacy systems.

In conclusion, blockchain technology has several limitations and challenges that hinder its widespread adoption. The scalability limitations, security and privacy concerns, interoperability and governance challenges, organizational challenges, governmental challenges, technical challenges, energy consumption, selfish mining, and the management of personal identifiable information are significant barriers to the adoption of blockchain technology. Addressing these challenges and developing innovative solutions are essential to realize the full potential of blockchain technology. Blockchain technology, introduced through Bitcoin, offers a decentralized and secure way to record and store information. Its potential applications are vast, and it holds promise for transforming various industries. As we look to the future, it's important to explore how blockchain can improve efficiency, transparency, and security while addressing challenges related to scalability and regulatory frameworks. Please note that this explanation is simplified to provide a basic understanding of blockchain technology and Bitcoin. There are many technical and complex aspects involved in these topics, but this overview should give you a general idea of how they work.

REFERENCES

[1] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An Overview of Blockchain Technology: Architecture, Consensus,

and Future Trends," Proceedings - 2017 IEEE 6th International Congress on Big Data, Big Data Congress 2017, pp. 557-564, 2017.

[2] M. Atzori, "Blockchain Technology and Decentralized Governance: Is the State Still Necessary?," SSRN Electronic Journal, 2015.

[3] A. Bahga and V. K. Madisetti, "Blockchain Platform for Industrial Internet of Things," Journal of Software Engineering and Applications, vol. 9, pp. 533-546, 2016.

[4] C. Delgado-Von-eitzen, L. Anido-Rifón, and M. J. Fernández-Iglesias, "Blockchain Applications in Education: A Systematic Literature Review," Applied Sciences, vol. 11, no. 24, p. 11-18, 2021.

[5] L. Luu, V. Narayanan, C. Zheng, K. Baweja, S. Gilbert, and P. Saxena, "A Secure Sharding Protocol for Open Blockchains," 2016.

[6] G. Zyskind and A. Pentland, "Enigma: Decentralized Computation Platform with Guaranteed Privacy," New Solutions for Cybersecurity, pp. 425-456, 2019.

[7] N. Szabo, "Formalizing and Securing Relationships on Public Networks," First Monday, vol. 2, no. 9, 1997.

[8] H. Watanabe, S. Fujimura, A. Nakadaira, Y. Miyazaki, A. Akutsu, and J. Kishigami, "Blockchain contract: Securing a blockchain applied to smart contracts," 2016 IEEE International Conference on Consumer Electronics, ICCE 2016, pp. 467-468, 2016.

[9] A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, "Hawk: The Blockchain Model of Cryptography and Privacy-Preserving Smart Contracts", 2015.

[10] T. Dickerson, P. Gazzillo, M. Herlihy, and E. Koskinen, "Adding Concurrency to Smart Contracts," Distrib Comput, vol. 33, no. 3–4, pp. 209–225, 2017.

[11] V. Gatteschi, F. Lamberti, C. Demartini, C. Pranteda, and V. Santamaría, "Blockchain and Smart Contracts for Insurance: Is the Technology Mature Enough?," Future Internet vol. 10, no. 2, pp. 20-23, 2018.

[12] B. Nissen, L. Pschetz, D. Murray-Rust, H. Mehrpouya, S. Oosthuizen, and C. Speed, "GeoCoin: Supporting ideation and collaborative design with smart contracts," Conference on Human Factors in Computing Systems - Proceedings, vol. 2, 2018.

[13] M. Shurman, A. A. R. Obeidat, and S. A. D. Al-Shurman, "Blockchain and Smart Contract for IoT," 2020 11th International Conference on Information and Communication Systems, ICICS 2020, pp. 361-366, 2020.

[14] T. Feng, X. Yu, Y. Chai, and Y. Liu, "Smart contract model for complex reality transaction," International Journal of Crowd Science, vol. 3, no. 2, pp. 184–197, 2019.

[15] M. Alharby and A. van Moorsel, "Block-

chain-based Smart Contracts: A Systematic Mapping Study," pp. 125-140, 2017.

[16] I. Eyal and E. G. Sirer, "Majority is not Enough: Bitcoin Mining is Vulnerable," Commun ACM, vol. 61, no. 7, pp. 95-102, 2013.

[17] Y. Yue, X. Li, D. Zhang, and S. Wang, "How cryptocurrency affects economy? A network analysis using bibliometric methods," International Review of Financial Analysis, vol. 77, 2021.

[18] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System", Accessed: Feb. 23, 2024.

[19] "Blockchain Enigma. Paradox. Opportunity". vol. 9, no. 2, pp. 39-41, 2016.

[20] D. Kraft, "Difficulty control for blockchain-based consensus systems," Peer Peer Netw Appl, vol. 9, no. 2, pp. 397-413, 2016.

[21] Y. Sompolinsky and A. Zohar, "Accelerating Bitcoin's Transaction Processing. Fast Money Grows on Trees, Not Chains," Cryptology ePrint Archive, 2013.

[22] A. Chepurnoy, M. Larangeira, and A. Ojiganov, "A Prunable Blockchain Consensus Protocol Based on Non-Interactive Proofs of Past States Retrievability," arXiv.org, 2016.

[23] A. Chandratre and A. Pathak, "Blockchain Based Intellectual Property Management," SSRN Electronic Journal, 2019.

[24] A. Ekblaw, A. Azaria, J. D. Halamka, A. Lippman, and T. Vieira, "A Case Study for Blockchain in Healthcare: 'MedRec' prototype for electronic health records and medical research data White Paper MedRec: Using Blockchain for Medical Data Access and Permission Management IEEE," 2016.

[25] K. Rarhi, "Melanie Swan Blockchain blueprint for a new economy." Infosec, 2023.

[26] Z. Gao, L. Xu, L. Chen, N. Shah, Y. Lu, and W. Shi, "Scalable blockchain based smart contract execution," Proceedings of the International Conference on Parallel and Distributed Systems - ICPADS, vol. 6, pp. 352-359, 2017.

[29] D. J. Daluwathumullagamage and A. Sims, "Blockchain-Enabled Corporate Governance and Regulation," International Journal of Financial Studies 2020, vol. 8, no. 2, p. 36-44, 2020.

[30] M. Pilkington, "Blockchain Technology: Principles and Applications." vol. 2, pp. 23-31, 2024.

[31] D. Meva, "Issues and Challenges with Blockchain a Survey," International Journal of Computer Sciences and Engineering, vol. 6, no. 12, pp. 488-491, 2018.

[32] C. Decker, J. Seidel, and R. Wattenhofer, "Bitcoin Meets Strong Consistency," vol. 4, pp. 31-34, 2016.

[34] H. Hellani, "On Blockchain Technology: Overview of Bitcoin and Future

Insights". pp. 41-45. 2019.

[35] F. J. de Haro-Olmo, Á. J. Varela-Vaca, and J. A. Álvarez-Bermejo, "Blockchain from the Perspective of Privacy and Anonymisation: A Systematic Literature Review," Sensors, vol. 20, no. 4, p. 71-76, 2020.

[36] W. Zhang, X. Zeng, H. Liang, Y. Xue, and X. Cao, "Understanding How Organizational Culture Affects Innovation Performance: A Management Context Perspective," Sustainability, vol. 15, no. 8, p. 60-64, 2023.

[37] M. Attaran and A. Gunasekaran, "Blockchain-enabled technology: The emerging technology set to reshape and decentralise many industries," International Journal of Applied Decision Sciences, vol. 12, no. 4, pp. 424-444, 2019.

Aftab *et al. (IJECI)* 2024

International Journal for
Electronic Crime Investigation

DOI: https://doi.org/10.54692/ijeci.2024.0801189

(IJECI)
ISSN: 2522-3429 (Print)
ISSN: 2616-6003 (Online)

Research Article

Vol. 8 issue 1 Jan-Mar 2024

Digital Investigations: Navigating Challenges in Tool Selection for Operating System Forensics

Kausar Parveen and Ghulam Haider

Department of Computer Sciences, University of Engineering and Technology, Lahore
Corresponding author: kausarnawaz6@gmail.com

Received: December 31, 2023; **Accepted:** January 18, 2024; **Published:** March 15, 2024

ABSTRACT

The process of gathering, identifying, extracting, and documenting electronic evidence for use in court is known as "digital forensics." We have a lot of tools at our disposal to make this procedure quick and straightforward. Four tools have been selected for investigation and analysis in this work. For every kind of digital forensics, the top tools have been selected based on several criteria. For computer forensic tools, (Stellar and Forensic Tool Kit) have been investigated; for network forensic tools, Network Map has been selected, and OSF mount has been examined as a live forensic tool. Other forensic tool types, such as database, operating system, and mail forensic tools, are also covered in this work. The role of Artificial intelligence in Digital Forensic tools has been discussed in this paper by using both Decision Stump and Bayes net machine learning techniques. After making an investigation of the IoT device traffic dataset using these two techniques, Decision Stump gives us less accurate results compared with Bayes net.

Keywords: Forensics tool, Digital Evidence, Artificial intelligence, Forensic Analysis, Challenges.

1. INTRODUCTION

The market for electronic devices, such as laptops, PCs, and portable electronics, is growing rapidly. Since these gadgets are networked and consume a lot of data, cybercrime is thought to be mostly caused by the Internet. Comparing digital forensics (DF) to other forensic sciences, DF is still in its infancy. DF science's involvement begins after the crime has committed [1]. The process of gathering, identifying, extracting, and documenting electronic evidence from various electronic devices for use as admissible evidence in a court of law is known as digital forensics (DF). [2]. The inquiry method mostly relies on the DF tools, which will yield efficient and productive outcomes. They are different types of data to deal with these tools like the Internet of Things (IoT) devices data, computer devices, mobile devices cloud computing, etc. [3]. Most of these tools' goals are to collect and recover the original files from the devices. DF tools are used for solving problems related to computer crimes like phishing, money laundering, bank Fraud, and child exploitation. The most of shreds of evidence have been found on computers [4]. As shown in figure 1, DF tools are divided into computer forensics network forensics, live forensics, Operating System forensics, database forensics, and Mail forensics. As a part of Artificial intelligence (AI) machine learning (ML) generally and deep learning especially have an important role in DF. As we know the AI technique can work with big data in a short time with accurate results. So, AI helps the investigators in the DF analysis process. The exuberance of forensic tools will make it hard for users to choose the relevant tool for their requirements [4], [5]. So, we explored the most popular tools and collect information about others to make a comparison between them. However, the investigator can choose the appropriate one for him and for the crime that he will investigate.

The top DF tools for computer, network, and live forensic tools were identified in this study based on a number of crucial factors that are taken into account for each category. For instance, whereas port scanning and packet analysis are crucial elements for network forensic tools, imaging and hashing are significant factors in computer forensic tools. RAM dumps and live log analysis are crucial requirements for real-time forensic technologies. The use of Artificial intelligence in DF tools has been discussed in this paper by using both Decision Stump and Bayes network machine learning techniques on IoT device datasets and a comparison between them has been made. Decision Stump gives us less accurate Results were compared with Bayes net, which is less concerned about the attributes or their relationships. On the other hand, the best outcomes were obtained from Bayes Net, as it effectively represented the conditional dependencies among a set of random variables. Each node in the network signifies a variable, and each directed edge represents a conditional relationship.

2. LITERATURE REVIEWS

They examined many DF analysis techniques in [4]. It claimed that the pattern recognition method is ideal for the DF's analysis step. Numerous DF tools are developed using characteristics derived from the detected patterns. Therefore, a variety of tools are crucial for finding solutions to all of the disputes that arise throughout the execution phase, in addition to being employed for the preservation and analysis of individual pieces of evidence data. Various methods for both live and dead forensic analysis were discussed in [6]. In addition to creating an understandable environment to aid a detective, it retains the crucial instructions from several DF programs, like WIRESHARK, Autopsy, O.S. forensic, TRUECRYPT, Forensic Tool Kit (FTK) Imager, and SANS SIFT. Besides, they accumulate information that can be transformed using live analysis, which sidesteps destroying the information due to the stoppage of the target node. [7] said the major process done by criminals is for destroying files by deleting, damaging, or overwriting hard disks, etc. The team only focused on how to recover the destruction data. To recover the damaged data with the help of different tools such as WIRESHARK, Autopsy, TRUECRYPT, FTK Imager, Operating system forensic, X-WAYS, and SANS SIFT. Researchers in [8] explained the attributes, constrictions, and applications of DF tools and compared them with other tools in assisting investigators or users in employing composite DF tackles for their inspection. [9] employed a machine learning technique and advising a scheme to diagnose abnormal packets and attacks. Naive Bayesian provided the best accuracy against other classifiers. [10] used NLP techniques to analyze DF shreds of evidence. [11] focused on the recent readiness and advances of DF tools in the composite atmosphere. [12] proposed a method to build a new intelligence DF model for storehouse willingness. [13] suggested an effective model for DF cloud Investigation called Cloud Forensics Investigation Model (CFIM) to pattern the crimes happening in the cloud forensically. [14] proposed a DF framework methodology for the social media network community. This system contains operative classifying digital devices, procedures, analyzing and obtaining DF pieces of evidence. [7] showed DF terms in the cyber world and informed a comparative analysis of the current stream state of forensics. [15] proposed building architecture for AI applications in the DF especially in the analysis stage. [16] described an analysis of up-to-date DF artificial intelligent schemes to raise these procedures in forensic correction. [17] said that the compression of data can disturb different DF stages. [18] analyzed different ML techniques and their usability in recognizing evidence by tracking file systems. The Machine Learning algorithms achieved good outcomes. [19] proposed a classification model for network traffic using Machine Learning techniques. The results revealed that the best outcome had been done by a random forest classifier. [20] analyzed network traffic to discover windows ransomware by spread on ML and accomplished a Total Form (TF) a percentage of 97.1% with the decision tree method. Researchers in [21] proposed a process of the text description of Natural Language Processing and spam email discovery. [22] suggested a model for the cataloging of attacks in the cloud atmosphere using ML procedures with a DF method. [23] proposed model of managing intellectual cybersecurity. The model practices AI procedures to make the analysis procedure of cybersecurity more proficient compared with old-style security instruments. Through the speedy development of technologies, it is important to select DF methods and frameworks. DF methods from 2015 to 2022 are offered in the next lines. [24] examine the environments, cruise anomaly information, and control relation report. [25] Conformist data collection process strategy, provision law of shaping the consistency of the DF pieces of evidence. [26] study the DF on IEC/ISO ethics. [27] employing Digital Forensic Readiness (DFR) mechanism in amenability through the IEC/ISO ethics. [28] proposed a model based on Online Natural Language Processing (NLP) for forensic investigation. The paper compared different DF tools in different groups such as computer Forensic Tools, Network Forensic Tools, O.S Forensic Tools, Live Forensic Tools, Database forensic tools, and Email Forensic Tools. Consequently, the investigators can choose the accurate tool used for their requirements easily.

Table 1: Analysis by using various techniques.

Ref.	Key Focus	Techniques/Tools	Findings/Contributions	Loopholes Analyzed
[4]	DF Analysis	Pattern Recognition Method	Examined various DF analysis techniques, emphasized the importance of pattern recognition in DF's analysis step.	-
[6]	Live and Dead Forensic Analysis	WIRESHARK, Autopsy, O.S. forensic, TRUECRYPT, FTK Imager, SANS SIFT	Explored methods for both live and dead forensic analysis.	-
[7]	Data Recovery	WIRESHARK, Autopsy, TRUECRYPT, FTK Imager, OS Forensic, X-WAYS, SANS SIFT	Focused on recovering data destroyed by criminals. Used different tools for data recovery.	Destruction of files by deleting, or overwriting hard disks, etc.
[8]	DF Tools Comparison	Various DF Tools	Explained attributes, constraints, and applications of DF tools. Compared DF tools with other tools for investigator assistance.	-
[9]	Machine Learning	Naive Bayesian	Used machine learning to diagnose abnormal packets and attacks. Naive Bayesian provided the best accuracy.	Abnormal packets and attacks.
[10]	NLP Techniques	NLP	Used NLP techniques to analyze DF shreds of evidence.	-
[11]	Recent Advances	-	Focused on recent readiness and advances of DF tools in a composite atmosphere.	-
[12]	Intelligence DF Model	-	Proposed a method to build a new intelligence DF model for storehouse willingness.	-

[13]	Cloud Forensics	Cloud Forensics Investigation Model (CFIM)	Suggested an effective model for DF cloud investigation called CFIM.	Crimes happening in the cloud
[14]	Social Media DF	DF Framework Methodology	Proposed a DF framework methodology for the social media network community.	Operative classifying digital devices
[15]	AI Applications	AI in DF Analysis	Proposed building architecture for AI applications in DF, especially in the analysis stage.	-
[16]	AI Schemes	AI in Forensic Correction	Described an analysis of up-to-date DF artificial intelligent schemes to improve forensic	-
[17]	Data Compression	-	Stated that data compression can disturb different DF stages.	-
[18]	ML Techniques	Machine Learning Algorithms	Analyzed different ML techniques for recognizing evidence by tracking file systems. Achieved good outcomes.	Usability in recognizing evidence by tracking file systems.
[19]	Network Traffic	Machine Learning Techniques (Random Forest)	Proposed a classification model for network traffic using ML techniques..	-
[20]	Ransomware Detection	ML (Decision Tree)	Analyzed network traffic to discover Windows ransomware using ML.	Windows ransomware detection.
[21]	NLP and Spam Email	Natural Language Processing	Proposed a process for the text description of NLP and spam email discovery.	Spam email discovery.
[22]	Cloud Attacks	ML Procedures with DF	Suggested a model for cataloging attacks in the cloud atmosphere	Cataloging attacks in the cloud atmosphere.
[23]	Cybersecurity	AI Procedures	Proposed a model of managing intellectual AI procedures.	Analysis procedure of cyber security.

3. DIGITAL FORENSIC STAGES

In DF the first prototypical suggested has four stages: Collection, Identification, Assessment, and Admission. different prototypical is suggested to describe the stages of collecting, analyzing, preservation, and reportage of the pieces of evidence produced by many devices. Recently, a growing number of extra complicated prototypical are suggested. The goal of these models is to speed up the whole investigation procedure. The variety of sources and devices of digital shreds of evidence results in a variety of DF procedure models [29]. There is no common procedure model appropriate to use for all forms of the investigation process. [30]. Figure 2 shows the different stages of DFs. The role of each phase is discussed below:

Despite DF being a new study zone, already has completed important growth. The growth is done by the enhancement of methodologies and technology, for example, tools for gathering and analyzing DF pieces of evidence. In DF, a method to do an investigation process is called a process model which is a context with a sum of stages to do an investigation. In DF investigation a standard methodology should define the sequence of actions need in the investigation process. A perfect process model should be wide-ranging, which means it should be applied to a large number of cases. If a framework is very simple and has fewer phases, the result is not provided good guidance to the process of investigation. Otherwise, if a framework has more stages with sub-steps of each stage, the result is more limited to its usage. Many studies place special attention on outlining the whole DF investigation process; significant DF frameworks were covered in [31]. More recently, development on the DF framework has focused on addressing more specialized issues such as gathering, examining, analyzing, and preserving evidence in a single phase. For instance, the triage paradigm [4], works well in circumstances when time is a crucial factor. By using the DF pledge, investigators
may obtain information about the illegal more quickly than they would have to wait for all reports, which might take weeks, months, or even longer.

4. IDENTIFICATION STAGE

At this point, the evidence is defined, examined, and its position and source are determined. Evidence shreds need to be handled carefully and correctly. This stage's objective is to safeguard the evidence's integrity. It should be safeguarded in conjunction with a record known as the Chain of Custody (COC), which is identified by the DF connection, the paper trail, or the DF evidence's sequential certification. It shows the gathering, transfer, control sequence, and analysis.

5. ACQUIRING STAGE

For more analysis, this stage helps to save the state of the pieces of evidence. In this stage, hard disk imaging is done as a copy of the data on the hard disk. Three kinds of acquisition are accepted according to law enforcement forensic duplication, mirror image, and live acquisition. A mirror image makes a forensics duplication which saves the backup of the device’s hard disk as a bit-for-bit cloning copy.

6. ANALYSIS STAGE

Three types of analysis can be performed at the analysis stage: restricted, partial, or complete analysis. The narrow study only considers a small portion of the available data. While full inquiry aids in determining the initial cause of the crime, partial analysis works with cookies, log papers, email files, etc. [4] Several tools made for the analysis step, such as Encase and FTK, which can handle a lot of scripts to extract information from the data that has to be examined.

7. REPORTING STAGE

The reporting stage helps to deduce, in a documented report form based on pieces of evidence. This is done with the help of digital crime laws represent the information for further investigation.

8. DIGITAL FORENSIC TOOLS

The software programs created for the DF investigation process in digital crimes are known as digital forensic tools. There are several DF tools available on the market. They also come in generic or commercially licensed versions. In this paragraph, we shall discuss DF tools in several groups and conduct a proportionate analysis of different tools within each group. A variety of factors, including technological considerations, general concerns, disk imaging, string searching, and legal difficulties, have been taken into account while choosing DF tools.

8.1. Computer forensic tools

Computer DF tools are intended to certify that the pieces of evidence taken out from computers are correct and dependable. There are different types of computer DF tools like Data and Disk seizure DF tools. A comparative investigation of five Computer forensic tools based on feature parameters questions have been made. For example, hashing, imaging, and data recovery. In this paper, the Stellar tool and Forensic Tool Kit (FTK) have been explored in this review for computer forensic analysis. Stellar: Stellar tool helps the investigator to find all files they want from the computer disk. Stellar is designed to be a comprehensive recovery tool to help its users to deal with all types of data loss scenarios, without needing any expert knowledge. Digital investigators can do normal or deep scanning. Figure 5 show the deep scanning mode. It does a whole signature-based file search which is useful for recovering the files that normal scanning could not found it.

Access Data's Forensic Toolkit, or FTK, is a computer forensics tool that searches a hard disk for various types of data. For instance, it can look for text strings on a disk or in deleted emails in order to decrypt encryption by utilizing them as a dictionary of passwords. Forensic Tool Kit Imager is a disk imaging application that is also connected to FTK. This program creates an image clone of a hard drive, generates hash values (such as Secure Hash Algorithms (SHA1) or Message-digest Algorithms (MD5)), and verifies the integrity of the result by comparing it to the original. Using the FTK tool throughout the file analysis process, the forensic data picture may be stored and examined in a variety of formats, including E01, DD/raw, and AD1.

Table 1 has the comparison of key parameters like imaging which is a technique of copying physical storage for making investigations and gathering shreds of evidence. The copy does not only include files, but every bit, sector, partition, files, deleted files, folder, and also unallocated spaces. The copy image is identical to all the device or drives architecture and contents. The second key parameter is hashing, the professionals in Digital forensics should use hashing algorithms, like MD5 and SHA1, to produce hash values of the original files which they use in an investigation to ensure that the pieces of evidence are not changed or modified during the investigation, pieces of evidence collection and analysis so they protect their integrity. Another reason for using hash values is that electronic pieces of evidence are shared with various parties during the investigation process like legal professionals, law enforcement, etc. So, we need to ensure that everybody has the same copies of the pieces of evidence. Stellar forensics that we chose to explore in this study calculates hash values automatically.

8.2. Network Forensic Tools

Network forensics works through interpreting and controlling networks to make an intrusion detection and find unknown malicious and abnormal threats through networks and their associated devices. The Nmap DF tool has been discovered in network investigation in this paper as shown in Figure 8. Network Map: This program examines the replies to packets sent in order to identify hosts and services on a computer network. We may probe networks using Nmap's many services, which include sophisticated services like vulnerability detection, host finding, and O.S. detection. Nmap may also work in varied situations of the network such congestion, latency, and high traffic during the scan process.

8.3. Live Forensic Tools

Active systems are the focus of live forensic. It concentrates on RAM attribute extraction and does a forensic analysis for it. Therefore, live forensics offer reliable and accurate data for investigations, which is far superior than the insufficient data from previous DF processes. We investigated the OSF mount tool as a live forensic tool in this article. Because it mounts image files produced by disk cloning programs such as OSF Clone, it is known as OSF mount. With OS Forensics, which is mounted as a virtual disk on Windows, the picture file may be examined. You may also use the OSF Mount Forensic program to mount CD-ROMs and DVDs as RAM drives.

Four live DF tools are chosen grounded on key parameters like dealing with Search, Logs analysis, memory dumping, and Live logs analysis which is the process of taking all the content in RAM and writing it to a storage device.

8.4. Other Forensic Tools

There are sub-branches of forensic tools like Database Forensic tools, O.S Forensic tools, and Email Forensic tools. These three types of tools have been explored in this section. Critical data is warehoused in various Database Management System (DBMS) i.e., Oracle as a Relational Database Management System store commercial data, MySQL work with web stores as a back-end packing, while SQLite stores personal data like SMS and browser bookmarks. So, databases need their special set of forensic tools. DF investigators still need the necessary DF tools to investigate Database Management Systems forensic objects. Also, we require to establish a special standard for artifact storage and its mechanisms to develop advanced analysis tools for Database. Operating System Forensics tools are used for recovering and gathering important information from the Operating System of the device. The goal is to find practical proof against the criminal. Four methods are used for Operating system forensics: disk-to-disk clone, disk-to-image file, disk-to-data file, and the backup of a file. This tool identifies abnormal files and makes a hash-matching signature. In a live manner, the data has been loaded and exported with all key parameters like module, run count, title, file size, category, last run time, date, time, etc. then, the report is generated and presented to the investigator includes I/O read-write, threads, total CPU, etc. Emails played an important role in communication through the internet like business communications and transmitting information between different devices. Unfortunately, there are a lot of encounters in email DF, for example, spoofing, forged emails, and Unsigned Re-emailing. Investigator has to collect the proof, identify the criminal, and show up the judgments. It can work with various Email formats, for example, .msg, emlx, .pdf, .mht, xps, etc. by examining header information, message body content, and other key parameters like time. In addition, it has a filtering option, exporting, saving, and analysis.

8.5. Digital Forensics Tools Evaluation Metrics

To enable the community of investigators to independently assess different tools, it is crucial to verify DF technologies using a variety of criteria. Additionally, developers will identify the areas of a tool that need improvement. Metrics should encompass all of the DF Tools' properties in order to achieve exceptional accuracy and meet all criteria. There hasn't been much study on the metrics of DF Tools in this subject, despite the proposal of a few techniques up until recently. A solution was put out by [32] who defined metrics to count the number of files generated by the list of files (referred to as the precision rate) and the number of pieces of evidence correctly created from the list of pieces of evidence (referred to as the accuracy rate).

The publication proposed a mechanism for assessing the tool's performance. The outcome is accurate if the evidence that has been replicated is identical to the original. An MD5 hashing technique can be used for this. Unfortunately, there are drawbacks to this approach as well. For instance, since the signatures will be altered, it is ineffective if a single bit is lost or altered during the collecting phase. Additionally, while the tool is not the source of issues like disk damage, the collection step may not retrieve the exact pieces of evidence.

8.6. Use Of Artificial Intelligence in Digital Forensic Tools

Digital investigators have a difficult time finding pieces of evidence in digital information. It has become difficult to specify an investigation and its source of proof. The various technology, specific procedures, and processes used in the DF investigation are not keeping up with the development of criminals. So, criminals use these weaknesses to do their crimes. Artificial intelligence (AI) is very important in identifying crime in DF investigations. An algorithm based on AI is very effective and highly recommended in detecting and preventing risks and criminal activity. Also, it is important in forecasting illegal activity. Researchers have used the available evidence data in court to condemn a person. The pattern recognition techniques are the best for the Analysis stage of the DF. Recognition of the Pattern has two procedures. The first is an examination and the other one is recognition. The features are taken out from the patterns to be recognized in the analysis step. Then, applying different methods of pattern recognition to these features are practical for DF investigation. These techniques are projected to improve diverse DF tools to identify and gather pieces of evidence that would be cooperative to deal with explicit kinds of digital criminalities. For example, the Jaro Winkler algorithm [33]and Cosine similarity function [34] are considered advanced pattern recognition algorithms for identity resolution in DF they are typically based on making similarity metrics for more complex strings. The increasing popularity of IoT devices and their privacy concerns encourage us to choose an IoT device traffic to analyze and make some investigations. We chose an IOT Fridge device traffic dataset from the University of New South Wales (UNSW) Canberra at Australian Defense Force Academy (ADFA). It contains six attributes (Time, date, temperature, condition, label, type) with full training set classifier. We examine this dataset using two different machine-learning techniques. We chose these techniques because they are two separate concepts. The first one is the decision stump tree which is the ML technique of a single-level of decision tree. Decision stumps frequently work with apparatuses named base learners or weak learners in ML. For nominal attributes, it builds a stump that has a sprig for every probable attribute rate or a stump has a double of leaves, the first one matches a specific class, and the second one has matched all the other classes. The second machine learning technique we used is the Bayesian network, it is ideal for predicting the probability of several possible known causes the occurrence of an event was the contributing factor. As we see in Table 6, compared with Bayes Net, Decision Stump gives us less accurate results because it does not care that much about the attributes or their relationships. It focuses only on how these attributes affect the target.

On another hand, the best results, we got from Bayes Net were because it represents a conditional dependency of a set of random variables. Each node in the network represents a variable, and each directed edge in the network represents a conditional relationship the Confusion Matrix of Bayes net.

These findings highlight the need for a digital inquiry and resolution in order to safeguard an IoT device owner's privacy. It also shows how important artificial intelligence methods are in this industry, particularly machine learning methods, and how continuing legislative discussions around ISP data collecting and utilization need to take IoT-specific issues into account.

9. CHALLENGES

The limits of the DF tools are highlighted in this section. In [35], four DF issues have been highlighted. The first is the difficulties with law enforcement and the legal system, which include issues with jurisdiction, privacy, legal procedure, inadequate provisions for criminal cases, standards, and the paucity of research on DF Tools. Second, technological difficulties with huge data, cloud computing, encryption, instability, and overuse of bandwidth. Thirdly, a lack of defined procedures, skilled specialists in DF, and a lack of unified formal representation and forensic understanding. Fourth, the difficulties in determining the incidence response, the reliability of audit trails, and the preparedness of DF. The globule in the hard drive with the computer storage capacity was the main focus of the researchers in [17]. The development of cameras, computers, and portable electronics is another. [36] highlights the massive data of DF challenge, especially in the Internet of Things (IoT), and suggested a data modification process in DF by distinguishing the imaging in a massive amount of forensic data. [37] emphasized DF process limitations with cloud atmosphere like volatility, namely records, data integrity, and creation of the forensic image. [38] presented DF process difficulties with the smartwatches.

10. CONCLUSION

An adequate investigation and incident response strategy must be employed to complete the inspection in the event of any digital crime or assault. The steps of DF inspection were discussed before along with a comparison of several DF Tools. The sort of crime or attack will determine which instruments are used for the investigation. Artificial Intelligence (AI) is playing a significant role in analysis and prediction. To determine which is better, many machine learning techniques are used, and they are validated using various metrics. The paper analyzed various tools like Computers, Networks, Databases, O.S, Live, and Mail DF Tools. In computer forensics, the Stellar tool has been chosen relatively to a comparison with other tools according to some features like imaging, hashing, recovery data, reparation capability, seizure, acquisition, and availability. In network forensic Nmap tool has been chosen according to some features like Port scanning, Packet analyzing & spoofing topology and protocol analyzing, and availability. OSF mount for the live forensic tool has been chosen according to some features in this study according to Live log analysis, RAM dumping, search, and availability. It likewise introduced the tender of AI in the DF framework. Additionally, some challenges are emphasized through supplementary the DF examination procedure. The future road of DF research should focus on the main challenges in this field like IoT forensics, Cloud DF as a service, big data, and new tools of DF. For example, determining specific data in IoT is stimulating the investigator to identify where to locate or straight the examination. Accordingly, the above challenges can consider as a research opportunity to continue in this field. As we mention before, the main problem in DF is the big forensic data, especially in network forensics and IOT forensics. Therefore, handling huge data in a trustworthy forensic manner is a major difficulty in DF, and this is seen to be an excellent chance for the researchers to develop new methods and tools to handle this large data. With DF, researchers may also employ artificial intelligence approaches. For instance, they can use natural language processing (NLP) to analyze DF data and Artificial Neural Networks (ANN) to recognize complex patterns in a variety of DF branches. In order to provide us with the ideal inquiry outcomes, future study should also concentrate on creating cutting-edge methods and instruments to examine more complex settings, such as clouds and networks that resemble cyberspace.

REFERENCES

[1] K. K. Sindhu and B. B. Meshram, “Digital Forensics and Cyber Crime Datamining” Journal of Information Security, vol. 3, no. 3, pp. 196-201, 2012.

[2] J. K. Alhassan, R. T. Oguntoye, S. Misra, A. Adewumi, R. Maskeliunas, and R. Damasevicius, “Comparative evaluation of mobile forensic tools”, Advances in Intelligent Systems and Computing, vol. 721, pp. 105-114. 2018.

[3] O. Osho, U. L. Mohammed, N. N. Nimzing, A. A. Uduimoh, and S. Misra, “Forensic Analysis of Mobile Banking Apps”, Computational Science and Its Applications, 2019, pp. 613-626, 2019.

[4] S. Sachdeva, B. L. Raina, and A. Sharma, “Analysis of Digital Forensic Tools”, Journal of Computer Theoratical Nanoscience, vol. 17, no. 6, pp. 2459-2467, 2020.

[5] H. Hibshi, T. Vidas, and L. Cranor, “Usability of forensics tools: A user study”, 6th International Conference on IT Security Incident Management and IT Forensics, pp. 81-91, 2011.

[6] C. H. Yang and P. H. Yen, “Fast Deployment of Computer Forensics with USBs”, Journal of Computer Theoratical Nanoscience, vol. 10, no.6, pp. 114-123. 2010.

[7] D. Joseph and K. Singh, “Review of Digital Forensic Models and A Proposal For Operating System Level Enhancements”, International Journal of Computer Science and Information Security, vol. 14, pp. 797-806, 2016.

[8] J. U. Lee and W. Y. Soh, “Comparative analysis on integrated digital forensic tools for digital forensic investigation”, IOP Conf Ser Mater Sci Eng, vol. 834, no. 1, p. 12-34, 2020.

[9] A. Abirami and Palanikumar, “Proactive Network Packet Classification Using Artificial Intelligence”, Computational Science, pp. 169-187, 2021.

[10] F. Amato, G. Cozzolino, V. Moscato, and F. Moscato, “Analyse digital forensic evidences through a semantic-based methodology and NLP techniques”, Computer Systems, vol. 98, pp. 297-307, 2019.

[11] T. Wu, F. Breitinger, and S. O’Shaughnessy, “Digital forensic tools: Recent advances and enhancing the status quo”, Digital Investigation, vol. 34, p. 30-39, 2020.

[12] J. Cosic, C. Schlehuber, and D. Morog, “Digital Forensic Investigation Process in Railway Environment”, International Conference on New Technologies, Mobility and Security, pp. 1-6. 2021.

[13] E. E. D. Hemdan and D. H. Manjaiah, “An Efficient Digital Forensic Model for Cybercrimes Investigation in Cloud Computing”, Multimedia Tools Applications, vol. 80, no. 9, pp.

14255-14282, 2021.

[14] Y. J. Jang and J. Kwak, “Digital forensics investigation methodology applicable for social network services”, Multimedia Tools Applications, vol. 74, no. 14, pp. 5029-5040, 2015.

[15] S. Costantini, G. D. Gasperis, and R. Olivieri, “Digital forensics and investigations meet artificial intelligence”, Ann Math Artif Intell, vol. 86, no. 1, pp. 193-229, 2019.

[16] A. Krivchenkov, B. Misnevs, and D. Pavlyuk, “Intelligent Methods in Digital Forensics: State of the Art”, Networks and Systems, pp. 274-284. 2019.

[17] D. Quick and K.-K. R. Choo, “Impacts of increasing volume of digital forensic data: A survey and future research challenges”, Digit Investigation, vol. 11, no. 4, pp. 273-294, 2014.

[18] R. Mohammad and M. Alq, “A comparison of machine learning techniques for file system forensics analysis”, Journal of Information Security and Applications, vol. 46, pp. 53-56, 2019.

[19] J. Pluskal, O. Lichtner, and O. Rysavy, “Traffic Classification and Application Identification in Network Forensics”, Advances in Digital Forensics, pp. 161-181. 2018.

[20] O. M. K. Alhawi, J. Baldwin, and A. Dehghantanha, “Leveraging machine learning techniques for windows ransomware network traffic detection”, Advances in Information Security, vol. 70, pp. 93-106, 2018.

[21] S. Srinivasan, V. Ravi, M. Alazab, S. Ketha, A. Al Zoubi, and S. Padannayil, “Spam Emails Detection Based on Distributed Word Embedding with Deep Learning”, Digital Investigation, pp. 161-189, 2018.

[22] S. Sachdeva and A. Ali, “Machine learning with digital forensics for attack classification in cloud network environment”, International Journal of System Assurance Engineering and Management, vol. 13, no. 1, pp. 156-165, 2022.

[23] I. Sarker, “Machine Learning for Intelligent Data Analysis and Automation in Cybersecurity: Current and Future Prospects”, Annals of Data Science, pp. 1-26, 2022.

[24] A. R. Jadhao and A. J. Agrawal, “A Digital Forensics Investigation Model for Social Networking Site,” Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies, 2016.

[25] R. Montasari, “A standardised data acquisition process model for digital forensic investigations”, pp. 23-28, 2017.

[26] I. Kigwana, V. R. Kebande, and H. S. Venter, “A proposed digital forensic investigation framework for an eGovernment structure for Uganda”, IST-Africa Week Conference

(IST-Africa), pp. 1-8. 2017.

[27] A. Singh, I. Adeyemi, and H. Venter, "Digital Forensic Readiness Framework for Ransomware Investigation", 10th International EAI Conference, ICDF2C 2018, New Orleans, LA, USA, pp. 91-105, 2019.

[28] D. Sun, X. Zhang, K.-K. R. Choo, L. Hu, and F. Wang, "NLP-based digital forensic investigation platform for online communications", Computer Security, vol. 104, pp. 10-22, 2021.

[29] X. Du, N.-A. Le-Khac, and M. Scanlon, "Evaluation of Digital Forensic Process Models with Respect to Digital Forensics as a Service", Multimedia Tools Applications, vol. 14, no. 4, pp. 50-54, 2017.

[30] M. Scanlon, "Battling the digital forensic backlog through data deduplication", Sixth International Conference on Innovative Computing Technology, pp. 10-14, 2016.

[31] M. D. Kohn, M. M. Eloff, and J. H. P. Eloff, "Integrated Digital Forensic Process Model", Computer Security, vol. 38, pp. 103-115, 2013.

[32] B. Hitchcock, N.-A. Le-Khac, and M. Scanlon, "Tiered forensic methodology model for Digital Field Triage by non-digital evidence specialists", Digital Investigation, vol. 16, pp. 75-85, 2016.

[33] C. Dietzel, T. U. Berlin, D. Cix, M. Wichtlhuber, G. Smaragdakis, and A. Feldmann, "Stellar: Network Attack Mitigation using Advanced Blackholing", pp. 3-9, 2019.

[34] DHS, "Access Data Forensic Toolkit (FTK) Version, Test Results for String Search Tool", Computer, pp. 19-23, 2016.

[35] K. M. A. Kamal, M. Alfadel, and M. S. Munia, "Memory forensics tools: Comparing processing time and left artifacts on volatile memory", International Workshop on Computational Intelligence, pp. 84-90, 2016.

[36] A. Dizdarević, S. Baraković, and J. Baraković Husić, "Examination of Digital Forensics Software Tools Performance: Open or Not?", International Symposium on Innovative and Interdisciplinary Applications of Advanced Technologies, pp. 442-451, 2020.

[37] D. Quick and K. K. R. Choo, "Digital Forensic Data Reduction by Selective Imaging", Computer, pp. 69-92, 2018.

[38] A. Shaaban and N. Abdelbaki, "Comparison study of digital forensics analysis techniques", Procedia Computer Science, vol. 141, pp. 545-551, 2018.

Aftab *et al. (IJECI)* 2024

International Journal for
Electronic Crime Investigation

DOI: https://doi.org/10.54692/ijeci.2024.0801190

(IJECI)
ISSN: 2522-3429 (Print)
ISSN: 2616-6003 (Online)

Research Article
Vol. 8 issue 1 Jan-Mar 2024

Role of Technology by Police to Maintain Peace During Muharram

Gulam Rasul Zahid[1], Gulam Abbas[2], Abdullah Hassan Hashmi[3] and Saima Sheikh[4]

[1]Police Service of Pakistan, Joint Director General Intelligence Bureau Islamabad, Pakistan.
[2]Riphah International University, Riphah College of Veterinary Sciences, Lahore, Pakistan.
[3]University Institute of Food Science and Technology, University of Lahore, Lahore, Pakistan.
[4]Admin Pakistan Association of Advancement of Sciences
Corresponding author: ghulamabbas_hashmi@yahoo.com

Received: January 03, 2023; **Accepted:** January 21, 2024; **Published:** March 15, 2024

ABSTRACT

Over the past several decades, policing agencies have implemented an array of technological advancements to improve operational efficiency. Pakistan has increased security measures across the country for muharram processions carried out for Ashura. Security has been enhanced across the country for the peaceful observance of Ashura due to wave of religious terrorism from previous few decads. For this, a huge personnel of regular police officers, along with an additional reserve officers, are deployed to guard mourning processions and gatherings across the country and to keep the law and order situation under control. Role of police in providing peace during muharram is very significant. Police personel perform their duties for late nights with great spirit during 1st week of Muharram under harsh weather (usually hot days), and danger of terrorism despite limited resources. Good behavior, determination, dedication of police force during the muharram duties is praise worthy and exemplary and it ensured maintenance of atmosphere of law and order work with diligence and commitment. Police officers remain alert to counter any untoward incident and keep keen eye on sensitive areas during Ashura. Police sealed many parts of inner as part of security for Ashura. They blocked roads and streets leading to these places. cellular phone signals remained suspended as part of upgraded security. Checkpoints are kept functional across the country and additional security personnel is served during 9th and10th muharram. District police personnel, bomb disposal units, Scouts, platoons of Constabulary and soldiers are deployed to protect processions and sacred places and authorities took extraordinary measures to confirm the security of people and religious gatherings during Ashura. Police installed closed circuit television cameras (CCTV) along the procession routes. Police champions bear tough circumstances and nothing can down the determination of the of Police force. We hope that the law enforcement agencies would continue to perform their duties in the same way to ensure the protection of life and property of the citizens across the country.

Keywords: Role, Police, Muharram, CCTV, Ashura, Pakistan.

1. INTRODUCTION

Technology and policing have been interconnected for decades, dating back to the advent of the telephone, the automobile, and the two-way radio. Today, technology seems to be advancing at an ever-accelerating pace, as seen through the propagation of mobile and wireless technology, high-powered computing, visual and audio technology, advanced analytics, and other technological advancements [1].

The Arabic term matam refers in general to an act or gesture of mourning; in Shia Islam, the term designates acts of lamentation for the martyrs of Karbala. Shia muslims gather in public for ceremonial chest beating as a display of their devotion to Imam Hussain (A.S.) and/or conduct majlis in remembrance of his suffering [2, 3]. In some Shi'a societies, such as those in Bahrain, Pakistan, India, Afghanistan, Iran, Syria, Bangladesh, and Iraq, male participants may incorporate knives or razors swung upon chains into their matam, there are two basic forms of matam, using one's hands only, that is, sineh-zani or chest-beating and matam with implements like chains, knives, swords and blades, that is, zanjeer-zani), qama-zani, etc [4].

Matam in South Asia is the most significant and sensitive Shia identity marker, although the act is also condemned by some Shi'a religious leaders. A form of ritual bloodletting, practiced as an act of mourning by some Shia Muslims (it is also forbidden according to Grand Ayatollahs), for the younger grandson of Muhammad, Hassain ibn Ali (A.S.) who was killed along with his family, children, companions and close relatives at the Battle of Karbala by the Umayyad leader Yazid [5,6].

The matam was first introduced by the Qizilbash tribe who were contributory in establishing the Safavid government and then a community of Shia maintain the [7] practice hence public nature of ashura endorse diverse spiritual, religious, and cultural ideas. One form of mourning is the theatrical re-enactment of the Battle of Karbala. In Iran, this is called taziya or taziyeh. Theatrical groups that specialize in taziya are called taziya groups. Taziyas were popular through the Qajar dynasty until the early twentieth century, but the re-enactments slowly declined until they were mostly abandoned in the large cities by the early 1940s. Nonetheless, taziyas continued to exist in Iran on a smaller scale, especially in more rural and traditional areas. Reza Shah, the first of the Pahlavi dynasty, outlawed taziyas [8]. Despite attempts since 1979, Muharram processions and various forms of the rawza khani are still more common [9].

By increasing the number of shia Muslims in cities and states, Muharram rituals have changed to a more elaborate form. In the 9th century, lamentation and wailing became propounded as a mourning tradition. Noha is the poem and story that was inspired by Maqtal al-Hassain A. S. The poet or another one reads the noha with a plaintive rhythm. The main subject of noha is the pain from the killing of Hussain (A.S.) ibn Ali. Noha consists of poems in different languages such as Arabic, Urdu, Farsi, Saraeki, Sindhi, and Punjabi.

The reaction of the audience in the reenactment of the Battle of Karbala episode is significant for the strengthening of distinct Shia identity and the weeping over the killing of Hussain ibn Ali (A.S.) and his followers is one of these reactions. There is a close relationship between lamentation and weeping [10]. According to the narration, Shia imams had emphasized weeping for them, so it was transmitted to future generations. According to Shia tradition, the weeping and the flow of tears provide condolences to Imam Hussain (A.S.)'s mother and his family and they believe that lamenting and weeping is just for offering condolences to Imam Hussain’s family, therefore, it is one of the good deeds done by the mourners of Hassain (A.S.) and will help them to save them from being condemned to hell on the day of judgment [11].

Depending on the condition of society, the Muharram processions rituals vary from one city to another. The common form is the starting of mourning processions from Imam Hassain (A.S.) marsiya and the participants would parade through the streets of their town or village, finally, they come back to Imam Hassain (A.S.) marsiya to perform other mourning of Muharram's ritual. The procession was a common ritual of mourning of dead persons in Arabic states before the appearance of Islam. The chest-beating, flagellation, and face-slapping are usual acts done during the mourning procession, but chest-beating and face-slapping have more precedence and the history of these acts has reached to Buyid dynasty period [12, 13].

In South Asia, literary and musical genres produced by both Shias and Sunnis, that have been inspired by the Battle of Karbala are performed during the month, such as marsiya, noha, and soaz. This is meant to increase the people's understanding of how the enemies fought The Battle of Karbala against Imam Hussain (A.S.) and his followers [14].

In Hyderabad, the Bibi-Ka-Alam procession is taken annually to mark the date. Speaking specifically of Pakistan, Ashura is observed across the country with solemnity every year to pay homage to Imam Imam Hussain (A.S.) and other martyrs of Karbala. Processions with elaborate tazias are carried out in various cities as thousands of security personnel protect the mourners [15]. The procession routes are dotted with sabeel (drinking stalls), which are especially set up to provide cold drinks and milk to participants, and the general public, after they have walked long distances in hot summer. Local administrations all over the country place hospitals and ambulance services on high alert. But while many of these procession routes have existed since before Partition, there are some interesting stories about how they came about [16]. Every year, on Muharram 9, as the sun sets, the central Ashura procession of Lahore departs from Nisar Haveli, Mochi Gate. After crossing some of the city’s prominent mosques, imambargahs, and bazaars, such as Mohalla Chehl Bibian, Imambargah Syed Wajid Ali Shah, Koocha Qazi Khana, Imambargah Maulvi Feroz Ali, Mohalla Pir Gilanian, Imambargah Syed Rajab Ali Shah, Chauhatta Mufti Baqir, Chowk Kotwali, Kashmiri Bazaar, Sunehri Masjid, Dabbi Bazaar, Gumti Bazaar, Tehsil Bazaar, Ucchi Masjid and Bhaati Chowk, it culminates at the historic Karbala Gamay Shah on Muharram 10. Some routs have been changed, but the route that was first adopted more than 200 years ago has not changed [17].

2. ROLE OF PAKISTAN GOVERNMENT TO PERMIT AZADARI IN THE COUNTRY

After the partition of the Subcontinent Indo-Pak, the government took responsibility for all the Shia's azadari. So, they issued licenses for their majalis and ways of jaloos. Govt. announced to protect them from the terrorist by police force. From 1972, Prime Minister Zulfiqar Ali Bhutto took stand terrorism against Azadari and after him, some governments of Pakistan could not do much for the protection of Azadari but Azadar promoted day by day. After 1999, President Musharraf took the important role in the protection of majalis and jaloos but terrorist attacks could not be stopped. After him, President Asif Ali Zardari provided generators to all the Imambargahs due to heavy load-shedding issues of light. After him, in 2013, Prime Minister Nawaz Sharif also support the same motto. Similarly, after the 2018 elections Prime Minister Imran Khan gave very much independence to azadari and also permission to do majlis in any houses after just informing the police station. So, that was an actual and good step for the ashura rituals [18].

2.1. Azadari and terrorism in Pakistan

Shia minority forms the second largest Shia population of any country, larger than the Shia majority in Iraq. In the last two decades, as many as 4,000 people are estimated to have died in sectarian fighting in Pakistan, 300 in 2006. On the day of Ashura, terrorist attacks occurred against Muharram processions in many countries of the world [19].

- In 1940 bomb was thrown on Ashura Procession in Delhi, 21 February [20].
- Raman [21] reported the explosion of a bomb at the sacred shrine of Hazrat Imam Ali
- Raza (A. S.) on 20 June, in Mashhad, Iran. A total of 25 people were killed during this attack and about 70 people got injured, it was the most horrible terrorist attack in Iran since 1981.
- In 2004 bomb attacks, during Shia pilgrimage to Karbala, March 2, Karbala, Iraq, and 178 people were killed whereas 5000 were injured [16].
- In 2008 due to clashes between Iraqi troops and members of a Shia cult 263 people were killed on 19 January in Basra and Nasiriya in Iraq [22].
- In 2009 (December 28), Karachi, Pakistan, an explosion of a bomb killed dozens of people during the Ashura procession, and hundreds were injured.
- 2010: detention of 200 Shia Muslims, at a shophouse in Sri Gombak known as Hauzah Imam Ali ar-Ridha (Hauzah ArRidha), 15 December, Selangor, Malaysia.
- In 2011: explosion of a bomb, during the Ashura procession on 28th December, Hilla and Baghdad, Iraq, 30 people were killed.
- In 2011(6th December) a suicide attack on Ashura procession, Kabul, Afghanistan, 63 people killed.
- In 2015 (on 24th October) three explosions, during the Ashura procession, at a mosque in Dhaka, Bangladesh, one person was killed, and 80 people were injured.

Suicidal Attack on Muharram processions

Although suicide bombing has long history, however, there is a recent increase in this crual incidents in Pakistan as over few last years thousands people have been killed in suicide bombing incidents by terrorists. Assessing the attitudes and perceptions of people toward suicide bombing can help understand some of the root causes of this phenomenon. The

majority of the muslims condemnsuicide bombing even no any religion supports this inhuman fundamentalism action that is probably due to some underlying psychiatric illness.
Detail of Suicidal attacks in 2022:

- On 20th January 2022 suicide bomb attack in Lahore
- On 25th January 2022 suicide bomb attack in Kech District attack
- On 2nd February 2022 suicide bomb attack in Panjgur and Naushki raids
- On 2nd March 2022 suicide bomb attack in Quetta
- On 3rd March 2022 Sibi suicide bombing
- On 4th March 2022 suicide bomb attack in Peshawar mosque
- On 15th March 2022 suicide bomb attack in Sibi IED explosion
- On 26th April 2022 suicide bomb attack in University of Karachi
- On 12th May 2022 suicide bomb attack in Karachi Saddar
- On 15th May 2022 suicide bomb attack in Miranshah
- On 16th May 2022 suicide bomb attack in Karachi Bolton Market

2.2. Role of Police in to conduct peaceful Muharram Processions

The social security is the right of every citizen and it is the duty of that state to provide religioua freedom to its inhabitants. For this purpose, Security Institutions (Army, Police, IB) of Pakistan do their duties for the the protection of the the public Public participating in any festival. In our country, on the the occasion of any festival (Political, Religious, Social) these institutions provide security to the the public.
Religious and cultural festivals are very important in any civilized society. The participation of the people in these shows their respect and interest in these festivals. Like other events, 9th and 10th Muharram (Called Youm Ashora) are great events for muslims. There is a great significance of these days because these days are mentioned to the martyred of Hazrat Imam Hussain (A.S).

Like other Muslims, people of Lahore also celebrate muharram with great excitement. In 1850, Syed Ghulam Ali Shah (Gammy Shah) arranged the first Azadari procession in Lahore. The procession was started from Mochi Gate and afterwards residents of Lahore followed and continued this Azadari procession till date.

Being a Republican State in Pakistan everybody can has/have his/her rights without any fear. Religious freedom is part of the Constitution of Pakistan. Everybody can live without any pressure and everyone has religious freedom. All the Muslims celebrate Youm Ashora in their own way whereas Fiqah Jaffria celebrates it by mourning and Azadari Processionis.

In Pakistan police had always provided foolproof security to all the majlis and ashura jaloos all over the country despite of their sect and interest. All the other institutions are on holiday but the police provide 24/7 duty on critical days of Muharram and face hardships these days to ready all the situations. A lot of sacrifice is given by the police officers and young person has been reported during these days in the country during terrorist attacks.

For the protection of azadar and azadari procession, all institutions launch their security programs. Like, Police establish a walk through gates at the the entry and exit points from the the majlis using intelligence ways. Traffic and Patroling Police also play its role to smooth control of traffic. That azadar can be azadari without any disturbance.

The police administration and founder of majlis finalize a fool proof plan to avoid any untoward incident. Strict ban enforced on provocative speeches, and material. 29 emergency ambulances, 11 fire vehicles, 4 rescue vehicles, water rescue teams, and rescuers on different mobile posts have been deployed to provide emergency cover in the Lahore during Muharram.

Activities of persons included in the Fourth Schedule and those of banned organizations are closely monitored and strict action is taken against the elements involved in delivering provocative speeches or wall-chalking etc irrespective of their sectarian, political, ethnic, or other affiliations without any discrimination. The security plan designed for the protection of majlis and processions are adhered to with its true spirit and all the departments concerned have been directed to remain active to maintain law and order. The Lahore police have prepared emergency contingency plans in close coordination with district administrations and Imamia. Police have sketched out a complete security plan for majalis and Jaloos during Muharram. For this over 7,000 emergency staff and scouts have been trained to control terrorists attck during Ashura. Every year before the Muharram government appoints duties of police officers in different cities for the security of majlis and jalooses of licensed imambargah. The paths of jalooses are covered by proper security, and checked by bomb detectors, and rescue teams are available every moment during the table. Police play a vital role in the safety of people. CCTV cameras are installed in the routes of jaloos and police officers are appointed for proper checking. The lady police officers are also appointed in ladies wings. The entrance of all the streets that are connected to the imambargah and jaloos paths arre strictly checked and closed through barriers and security police is appointed there with heavy weapons. So, police provide all the security for the safety of Muharram. Search, sweep, combing and/or intelligence-based operations are continued in and around the localities of gatherings and procession routes. A force of plain clothed commandos and snipers on rooftops of buildings, on the route of mourning processions/religious gatherings is also deployed [23]. Security forces are made being high alert across Pakistan and authorities likely elevate security around all azadari sites and near procession routes, especially in major urban centers.

Police personnel set up checkpoints on major roadways, increased patrolling hours at night, and increase security at malls, markets, and other soft targets as a precautionary measure [23]. Violence is possible in many areas, particularly in major cities such as Islamabad, Karachi, Lahore, Peshawar, Quetta, and Rawalpindi.

Major Ashura procession routes points are typically at the following locations:

- Islamabad: Markazi Imambargah in Sector G-6 to Melody Market, and back to Markazi Imambargah
- Karachi: Nishtar Park on M.A. Jinnah Road to Hussainian Iranian Imambargah in Kharadar
- Lahore: Mochi Gate to Lower Mall, as well as Nisar Haveli to Karbala Gamay Shah Multan: Mumtazabad to Shah Shams Shrine
- Peshawar: Qissa Khawani to Kohati Gate
- Quetta: Rehmatullah Chowk on Alamdar Road to Punjabi Imambargah
- Rawalpindi: Imambargah Col. Maqbool Hussain on College Road to Imambargah Qadeemi of Banni via Fawara Chowk.

During Pandemic police also successfully enforce COVID-19 restrictions in permitted gatherings. Officials may impose temporary localized telecommunication restrictions to prevent militant attacks and sectarian clashes. To ensure peace and security during Muharram, the police have sucessfully prepare the sound plan. A team of high police officials, led by the Capital City Police chief, arrange the security arrangements with stakeholders such as organizers of Majalis, Imambargahs caretakers and, religious leaders as well as members of divisional and executive peace committees. The high command police officers visit the Imambargahs and routes of mourning processions in the city. The police make peace and religious harmony possible by cooperation of people from various groups to maintain the spirit of brotherhood on the occasion of Muharram. Some 15,000 cops drawn from various squads and units of the police force are also deployed for security outside Imambargahs, along the routes of mourning processions, and at public places. The police, with the assistance of the Counter Terrorism Department and the Federal Investigation Agency keep a strict watch on social media that may try to cause hatred among people. The police also play role to draw attention of the people towards the external dangers the country is facing and to get support against enemy machinations. The security arrangements are discussed in a meeting presided over by the Senior Superintendent of Police (Operations) which is also attended among others by all Zonal SPs, SP (Investigation), Sub-Divisional Police Officers, and Station House Officers. All wings of police were given directions to ensure complete coordination for the success of this plan. The SSP has directed all SDPOs and SHOs to maintain close liaison with peace committees and organizers of 'majlis' and processions in Muharram-ul-Haram. The SSP is directed strict security arrangements for gatherings and special checking of participants. Police personel make it ensure to arrange strict checking of participants of processions using metal detectors. Strict vigilance is maintained to ensure security measures are in place by the police as well as peace committees, the SSP maintained. Police is directed to launch an effective search operation and combining activities in the slum areas of the city including Afghan habitats and enhancing vigilance at all entrance points of the city. All SPs are asked to monitor this search operation themselves and inform the SSP's office daily about progress in this regard.

3. TRAFFIC MANAGEMENT BY POLICE

As the Ashura is very sensitive ritual therefore traffic police provide foolproof security by strict traffic management this day. Police have deployed closed-circuit video cameras along all the procession routes to facilitate strict command and control monitoring cell at police headquarters in the country. The Police established commendable, foolproof security procedures the major procession of Muharram to proceed peacefully to their destinations. Considering the threat of placing and planting any subversive material within mosques or Imambargahs, at the locations of Majalis, along the routes of the processions, particularly in abandoned buildings, the administration enacted additional security measures.

According to the Police chief, the major goal of the police specified in this security order is to protect lives, maintain law and order, ensure the orderly and peaceful conduct of azadari processions, and to be alert to deal with any emergency or terrorist activity. The police forces of the districts and divisional police chiefs, SDPOs, and SHOs, as well as the police chiefs of the South and East Ranges are in charge of overseeing the security measures in their respective jurisdictions.

4. ENFORCEMT OF LAW AND REGULATION FOR SECURITY OF ASHURA PROCESSION

The authorities order to safeguard the safety and security of the people and belonging of mourners as well as the general public. The government also order station house officers to go-ahead to file complaints against anyone found to be in breach of the prohibition under Section 188 of the Pakistan Penal Code. The Punjab Home Department apply enforcement of Section 144 and enlist the assistance of the Pakistan Army in order to ensure the maintenance of law and order and public safety during the Muharram. In light of this decision, a notification is released to enforce Section 144 throughout Punjab during Ashur-e-Muharram. The notification states that the provincial government has called upon the Army personnel to remain on standby in order to provide support to the police force in maintaining law and order, particularly in areas deemed sensitive.

The Home Department maintains and ensure the presence of army, mostly on Muharram 8, 9, and 10th muharram to guarantee the safety of the public in sensitive zones. Army personnel is appointed along the routes of major processions on Muharram 9 and Ashura. These provisions encompass a prohibition on initiating new processions, congregating in groups, and engaging in any actions that could potentially impede public order. Furthermore, the utilization of knives, swords, and sticks during processions has been unequivocally forbidden. In order to maintain the safety of residents and participants during Muharram processions, the authorities have implemented a ban on individuals standing on rooftops of houses and shops along the procession routes. This measure has been put in place to prevent any accidents or incidents that may arise due to overcrowding or unsafe behavior.

Additionally, in an effort to minimize the risks associated with motorcycle-related incidents during the climax of the religious observance on 9th and 10th Muharram, a ban on pillion riding has been enforced. The intention behind this measure is to reduce the number of people on the roads, thereby decreasing the potential for accidents and ensuring public safety. Section 144 is a legal provision that grants authority to local administrations in order to prohibit gatherings of five or more individuals, public processions, and demonstrations, with the objective of averting any potential threats to public peace and tranquility. In regard to Muharram, the implementation of Section 144 aims at preventing any untoward incidents from occurring during this sensitive period [24].

For the 110 Imambargahs in the district Rawalpindi, there had been extensive security set up under the plan. On July 30, 2022, mourning majalis started. 112 radical zakirs from different sects are not permitted to enter the district during the month of Muharram, according to the capital government. It's interesting to note that the list also includes several zakirs and academics who have passed away. Approximately 3,500 police officers, Rangers, and volunteers had been stationed on procession routes throughout Muharram as part of the security plan, particularly on the 9th and 10th of Ashura. The district had planned the impenetrable protection of Imambargah for the tranquil conclusion of Muharram processions. Helicopters were used to observe the funeral processions from above. Drone coverage of funeral processions had been forbidden by the security arrangement.

The capital territory police also made sure that the Residents were prohibited from standing on rooftops, balconies, or plazas during the Ashura processions. Additionally, gathering bricks from the rooftops of homes on each side of procession routes was prohibited by the capital territory police. No one would be permitted to carry matches or lighters during the Muharram processions according to the security arrangement by the police chief. People entered mourning processions through walk-through gates, and barbed wire had been placed to close off all streets that were in the procession's path which were also guarded by police officers. With the use of carts and containers, central roads were also sealed.

5. SAFETY OF THE PROCESSION

All arrangements required to assist the mourning had been completed by the capital police department and law enforcement. By Imambargah, all of the processions had come to an end at 10:00 p.m. Religious academics and clerics would draw attention to the tragedy's different elements as well as Imam Hussain's (RA) teachings. No one other than approved police personnel was permitted to carry a weapon during the Muharram parade, a police spokesperson said. According to him, police had not permitted anyone to arrange a brand-new Muharram parade, and stern punishment had been taken against anyone found to have violated SOPs. He added that a separate, round-the-clock control room had been established in his workplace. Due to the installation of CCTV cameras along the routes of the main Zuljinnah processions and in the locations where majalis had been held, Ashura processions were strictly supervised. Before the processions began, the routes had been thoroughly checked, and a bomb disposal team had cleared them. In addition to District Police, Ladies Police, Elite Force, Anti-Riot Force, Traffic Police, Punjab Highway Patrol, Punjab Constabulary, and Dolphin Force personnel, 2000 volunteers were also in charge of security. The Ashura processions had been followed by 20 mobile rescue squads. A thorough plan had been developed to give the mourners first aid and on-site pre-hospital medical care while critical patients were transported to local hospitals.

6. PUNJAB POLICE ALL SET FOR STRICT SECURITY IN MUHARRAM

Conducting of peaceful Muharram, protection of people of all schools of thought, as well as full security for Muharram meetings and mourning processions is the top priority of police department. More than 5,000 majlis and 650 mourning processions provide foolproof security during the forthcoming Muharram, whereas more than 4000 majalis held from the 1st to the 10th of Muharram. The maximum number of 143 mourning processions out of the total would be held in the City division during Ashura [25]. The CCPO directed SPs and supervisory officers to stay in constant liaison with the organizers of majlis and mourning processions, license holders, scholars of all schools of thought, businessmen, and local representatives and all the supervisory officers are asked to identify trouble points in their respective areas of responsibility and resolve the disputes and conflicts in consultation with stakeholders, ensuring surety bonds, mandatory to ensure a feeling of peace.

The CCPO direct the SPs, Circle Officers, and SHOs to supervise operational and logistic matters including briefing the staff on the security of routes, ensuring adherence to the sound system act, enforcement of restrictions of routes and timings of Majalis and mourning processions. The social media monitoring, search and sweep operations, geo-tagging, police pickets, and registration of tenants is ensured to avoid any mischievous actions of anti-peace elements. Checking of all suspicious vehicles and persons entering the city is also being ensured through the e-police checking mechanism at the entry and exit points of the city. The construction of vantage points, installation of walk-through gates, metal detectors, and CCTV cameras at the Imambargahs are ensured in collaboration with the administrators. The participants of the main mourning procession are provided three layers of security and nobody is allowed to enter the mourning processions and Majalis without complete checking.

The central mourning procession and other sensitive programs are continuously monitored with the help of CCTV cameras of the Punjab Safe Cities Authority and the district administration, he said, and added that strict action would be taken against those spreading hate material, literature, and divisive propaganda. All police officers are fully aware of logistics and human resources at the police station, circles, and divisional levels, he added. DIG Operations, DIG Investigation, SSP Operations Mustansar Feroze, SSP Investigation, SPs, and concerned police officers attended the meeting. Inspector General Police (IGP) Punjab Faisal Shahkar reported that the security of majlis and processions during Ashura Muharram are made with consultation of all stakeholders including clerics, peace committees, civil society, and district administration and that the Special Branch and Counter Terrorism Department (CTD) provide all possible support and provide reports to field commanders for security arrangements of Muharram. The IG Punjab directed that all preparations for the security of Muharram should be completed on time so that there was no danger of disturbance of peace and that the process of punishment should be ensured immediately, and all supervisory officers should dispose of the pending show-cause notices at the earliest in a week. The existing workload is completed within 7 days and the progress report is submitted to the Central Police Office. Police officers are directed to keep constant contact with their Jawans so that they do not face any difficulty in solving their problems. The IGP directed all supervisory officers including DPOs, and SPs to ensure that the orderly room was held regularly.

The IG Punjab while giving instructions to the officers for a zero-tolerance policy against black sheep involved in corruption, mistreatment of citizens, and illegal activities be strictly punished against one-wheelers, kite flying, and drug dealers [26].

(Starts from 9th Muharram 11pm to 10th Muharram 9:30 pm).

Starts from Nisar Haweli at 11:00 pm and moves towards Chowk Nawab Sahab and turns back to Lal Khuh Bazar (Mochi gate). Then after passing from Tawela Nawab, Muhala Shiyan, it returns to Kashmiri bazar then moves to Chowk Rang Mehal. After that, it moves to Tehsil Bazar turns left from Judge Latif Chowk and moves straight to Bhati Gate.It ends at Karbala Gamy Shah.

Table 1: Detail of Road Map Of 9th ,10th Muharam Procession of Gamay Shah K Arbla LAhore

Streets	136
Mosques	37
Safe City Cameras	147
CCTV Camera	79
Darbar	04
Amam Bargah	38
Total Distance	5.1 Km

7. ROLE OF TRAFFIC POLICE TO CONTROL TRAFFIC DURING ASHURA

The Islamic month of Muharram-Ul-Haram is a month of mourning for all Muslim communities around the world as it includes the remembrance of happenings of ‘Karbala’. Mourning processions are regularly organized and practiced mostly among the ‘Shia’ community and groups from all over Pakistan commemorate to organize ‘Majaalis’.

Traffic police also have a greater role like other public services body because leading these groups safely to their ‘Imaam Baargah’ is a big responsibility. Stops and barriers are placed all along the cities and the main roads and routes so that there is no stoppage for the communities that are going to their processions or even other people who are just carrying out their normal routine. Multiple traffic check-posts ensure the safety of all citizens and meanwhile, they selflessly and fearlessly perform their duties whether it is raining or sunny. Keeping an eye on the traffic and keeping everyone on track is a major contribution by the traffic police in satisfying all communities during the Ashura days.

Moreover not only keeping traffic on track, the traffic wardens also contribute significant role in conducting the majlis by helping them mourners out in re-routing when lost as well as lending a hand to make it easier to prepare and districbute feasts (lungar) whenever and wherever necessary.Regardless of sects and beliefs as well as the difference in the school of thought these wardens selflessly and sincerely guide and perform all they can on duty without any hesitation and with commitment although the rest of the country practices holidays.

Keeping aside their family and children with sheer commitment and wholesomeness the traffic police department performs their duties as an obligation and never demands any reward in return. Technically they are assigned to just keep an eye on the activity of the citizens on the road and keep in check that everyone follows traffic rules but they always go a step further to ensure there is no ambiguous activity that causes any problem for anyone. In addition, they have a hawk-eyed observation of every movement and every act that is being performed so that the safety of the citizens and the mourners is never compromised. Providing alternate and short routes for ambulances and 1122 emergency first-aid bikes is a great and impeccable contribution of traffic police towards the nation which is always appreciated and blessed with the Du’as of the nation. Metro bus service has been suspended in view of the security of Ashura Day, speedobs service has also been limited from MAO College to Shahdara, orange line train operation is going on as usual.

They have always been alert and ready to sacrifice themselves wherever required just to maintain the peace and prosperity in the community among the citizens. Sights and scenes of wardens helping out the women, children, and elderly people at Majaalis are always heart-warming and the commitment and dedication that they show is definite that Allah Almighty will surely greatly reward them for these deeds that they perform. The traffic police department has intelligently performed its duties during Ashura keeping the cities working and hustling as well as the mourners carrying out their peaceful processions. The services of the real heroes of the land of the pure are and will always be appreciated in good words by the citizens Insh A Allah [27].

8. ROLE OF SPECIAL POLICE BRANCH TO ENSURE SECURITY FOR MUHARRAM PROCESSIONS

The organization is headed by an officer who is assisted by 4 Deputy Inspectors-General of Police and some senior superintendents, superintendents, deputy and assistant superintendents, and other staff. For administrative and operational purposes, the organization is divided into 9 regions, each headed by an officer of the rank of Senior Superintendent or Superintendent of Police. It has field offices in all districts of the Punjab. The main duty of this special police is to acquire and develop intelligence usually of political or sensitive nature and conducts investigation to protect the state from perceived threats of subversion and other terrorist activity and extremist political activity Special police is deployed during national events to ensure the safety of the public like the most active of Moharram procession. Multiple police branches officers are dispatched for example firefighters normal police bomb disposal squads etc Bomb disposal squads are always on call and are ready to jump into action as soon as they are needed. These are highly trained professionals who are very familiar with the mechanism inside the bomb. Bomb disposal squads continuously check the routes and areas surrounding the routes that the people take during the procession. Moreover, drones are also used to check the area from the high grounds. If anything suspicious is recorded then it is immediately informed the authorities and safety measures are take [28].

The network system during ashura is temporarily disabled to prevent any leaking of information and potential threats of hacking and viruses are also eliminated. The police disguise themselves as normal civilian and they provide information to other police officers. They mix themselves among normal people and perform the same activities as them to lower the suspicion while keeping a very close eye on their surroundings. They look closely at everyone and if they notice someone suspicious then they alert the other police officers and follow the suspect. If the person turns out to be a criminal then they are immediately sent to jail. Medical staff is also available at all times. If someone gets hurt then they are immediately transferred to hospitals.

Special gates are present which ensure proper arrangement of the people involved in the procession and everyone is monitored thoroughly to make sure no one is caring any weapon or firearm. Routs that will be used in the processions are closed of using large containers and no vehicles are allowed to enter professionals are hired who continuously monitor the footage provided by the cameras police officers and drones. They watch it carefully and check for any suspicious activity. Everything is directly reported to the higher offices who overlook all the departments involved in the procession [29].

9. CONCLUSION

To protect muharram processions in the future, today all the police and security authorities should check upcoming years' measures of security all over Pakistan. All SSPs, SPs, and SDPOs should make it possible to control wall chalking or pasting posters which can hurt the sentiments of people. Patrolling officers must conduct visits to all the routes of Muharram processions and ‘imambargahs’ and rooftops to be covered by the police officials. The lady

Police should be deputed for the female gatherings while police officers and 'jawans' would perform security duties outside the 'imambargahs'. All Station House Officers (SHOs) should get the complete bio-data of those volunteers performing security duties. Peace committees would be asked to ensure that no stranger is allowed to stay in the worship places for security reasons during the month of Muharram. In Punjab: In Muharram, more police officers, volunteers, special police, and Razakars should be deployed in Punjab for security. The officers should provide foolproof four-layered security cover to the sensitive Majalis and processions during Muharram to maintain law and order in all districts of the province. The police should work closely with the Punjab Safe City Authority to ensure security monitoring through maximum CCTVs installed in different parts of the city as part of a high-tech surveillance project. Similarly, the police should carry out the flag march in sensitive areas to promote a sense of security among citizens. Policewomen would be given security briefings before being deployed at the main gatherings and mourning processions. The Special Branch, an intelligence wing of the Punjab police, should assist the law enforcement agency during field operations. A regional police officers (RPO) conference should be called to point out sensitive and most sensitive zonesin major cities of Punjab in Muharram. To meet any untoward situation during the holy month, elite teams of IG's reserve will remain on standby in districts with Punjab Constabulary platoons of the IG's reserve on high alert. Companies of the army and Rangers should also remain on standby. Leaves of all operational and security officials should be canceled for Muharram. Special passes should be issued to all security personnel to prevent criminals from disguising themselves in police uniforms. Strict action should be taken against those displaying firearms, and weapons, and firing in the air. Security guards should be provided temporarily to personalities who have received threats. The IGs should direct officers to ensure the installation of searchlights and a public address system on security vehicles besides arranging for video recording of all processions with the cooperation of the counter-terrorism department and Special Branch. Installation of CCTV cameras, barbed wires, walkthrough gates, jammers, and metal detectors at all entry and exit points of cities and processions must be sured. Monitoring of houses on procession routes must also be included in the security plan. The decision to monitor anti-state elements involved in walk chalking, publication of hate material, misuse of loudspeakers, and ulemas banned and prohibited from addressing publicly should also made mandatory. For Emergency Plan, emergency ambulances; fire vehicles, rescue vehicles, water rescue teams, and rescuers on different mobile posts should be deployed to provide emergency cover to all mourning processions. All mobile posts and ambulances with medical kits should manage to move along with the processions to provide mourners with immediate medical treatment. Emergency officers, including trained doctors, should supervise the arrangements on 9th and 10th muharram. Rangers should also form a Crisis Management Cell Headquarters to deal with any untoward incidents. Moreover, troops should actively conduct snap-checking, and mobile patrolling in different parts of the province. Rangers should asked to warn and discourage the hate material, hate speeches and that strict restrictions be placed on aerial firing and show of weapons. Strict action will be taken against those who fail to follow the rules, according to the Rangers. Citizens should be advised to contact Rangers on their WhatsApp number 0316-2369996 and helpline 1101 to report any suspicious activity. Safe City Project can help in curbing street crimes during muharram.In the future, we hope that All institutions will provide security to azadar by maintaining their traditions. We hope that officers of all these departments will watch everything. And provide full proof security to azadar and azadari procession. And save the azadari procession from any disturbing situation. It is the moral and constitutional right of azadar to demand security from the Government. Because constitutionally that is their right.

REFERENCES

[1] Kevin Strom, “Research on the Impact of Technology on Policing Strategy in the 21st Century”, National Criminal Justice Reference Service, pp. 1-15, 2017.

[2] G. Rizvi, “The Rivalry Between India and Pakistan”, B. Buzan and G. Rizvi (eds.), pp. 101-112, 1986.

[3] M. Moufahimand M. Lichrou, "Pilgrimage, consumption and rituals: Spiritual authenticity in a Shia Muslim pilgrimage", Tourism Management, vol. 70, pp. 322-332, 2019.

[4] GOP, “Punjab Police over 175k Policemen to Perform Muharram Duty", Over 175k policemen to perform Muharram duty, Punjab Police. 2022.

[5] V. D. Volkan, “The Linking Objects of Pathological Mourners”. Archives of General Psychiatry, vol. 27, no. 2, pp. 215-221,1972.

[6] F. J. Korom and P. J. Chelkowski, "Community process and the performance of Muharram observances in Trinidad". TDR (1988), vol. 38, no. 2, pp. 150-175, 1994.

[7] S. M. Fazel, "Ethnohistory of the Qizilbash in Kabul: Migration, State, and a Shi’a Minority", Indiana University, 2017.

[8] A. S. K. Al-Khafaji, "A Study of Suffering and Martyrdom in Islamic Ta’ziya and Christian Passion Plays", Lublin Studies in Modern Languages and Literature, vol. 41, no. 1, pp. 7-19, 2017

[9] N. R. Keddie, "Shi’ism: Myth and Reality, Religion and Secularism, In Iran and the Muslim World: Resistance and Revolution", Springer, pp. 174-190, 1995.

[10] K. M. H. Al-Zubeidy and F. H. F. Al-Mufriji, "The Battle of Karbala in 1842 A Historic Study", Al-Ssebt Journal, vol. 1, no. 4, pp. 20-31, 2017

[11] J. R. Halverson, H. L. Goodall and S. R. Corman, "The Battle of Karbala”, Narratives of Islamist Extremism, pp. 81–93, 2011.

[12] K. N. Fattah, "Bihari Muharram processions in Dhaka city: religious performance as a means of territorialising urban public space by a disenfranchised ethnic minority". Visual Studies, vol. 35, no. 2, pp.161-168, 2020.

[13] M. R. Nejad, "Urban Margins, a Refuge for Muharram Processions in Bombay: Towards an Idea of Cultural Resilience", South Asia Chronicle, vol. 5, pp. 325-346. 2015.

[14] L. Carroll, "The Ithna Ashari Law of Intestate Succession: An Introduction to Shia Law Applicable in South Asia". Modern Asian Studies, vol. 19, no.1, pp. 85-124, 1985.

[15] S. B. Freitag, "Collective action and community: public arenas and the emergence of communalism in North India", University of California Press, 1989.

[16] K. G. Hjortshoj, "Kerbala in Context: A Study of Muharram in Lucknow, India", Cornell University, 1977.

[17] S. A. Raza, "Religious Festivals of Lahore", Journal of the Punjab University Historical Society, vol. 28, no.2, pp. 34-41, 2015.

[18] M. Abou Zahab, "Muharram Procession in Pakistani Punjab", South Asian Religion on Display: Religious Processions in South Asia and in the Diaspora, pp. 104-114, 2008.

[19] R. E. Hassner, "Religion on the Battlefield", Cornell University Press, pp. 40-45, 2016.

[20] J. N. Hollister, "The Shi'a of India", Oriental Books Reprint Corporation, 1979.

[21] B. Raman, "Sipah-E-Sahaba Pakistan, Lashkar-e-Jhangvi, Bin Laden and Ramzi Yousef", Archived, 2002.

[22] BBC, "Iraqi Shia pilgrims mark holy day", Iraqi Shia pilgrims mark holy day, 2008

[24] Dunya news, "Punjab imposes section 144 for Ashura", Punjab imposes Section 144 for Ashura - Pakistan - Dunya News, 2023.

[25] Punjab Police, "Security plan implemented successfully", Ashura security plan implemented successfully, Punjab Police, 2022.

[26] Dailytimes, "Faisal Shahkar officially takes over the command of Punjab Police", 2022.

[27] S. A. Dogra, "Living a piety-led life beyond Muharram: becoming or being a South Asian Shia Muslim in the UK", Contemporary Islam, vol. 13, no. 3, pp. 307-324, 2019.

[28] W. Ende, "The Nakhāwila, a Shite Community in Medina Past and Present", Die Welt Des Islams, pp. 263-348, 1997.

[29] S. Khan, W. I. Chaudhry and I. Badshah, "Ethnographic Study of Muharram Rituals in a Punjabi Village in Pakistan", Journal of Asian Civilizations, vol. 37, no. 2, pp. 123-131, 2014.

Taseer *et al. (IJECI)* 2024

International Journal for
Electronic Crime Investigation

DOI: https://doi.org/10.54692/ijeci.2024.0801191

(IJECI)
ISSN: 2522-3429 (Print)
ISSN: 2616-6003 (Online)

Research Article

Vol. 8 issue 1 Jan-Mar 2024

Malware Detection and Analysis Using Reverse Engineering

Muhammad Taseer Suleman

School of Electrical Engineering and Computer Sciences, NUST, Islamabad, Pakistan
Corresponding author:12msccsmsuleman@seecs.edu.pk

Received: January 06, 2023; **Accepted:** January 22, 2024; **Published:** March 15, 2024

ABSTRACT

The pervasive and persistent nature of malware in the contemporary digital realm demands sophisticated methodologies for detection and analysis. Reverse engineering has emerged as a pivotal strategy in malware analysis, offering the means to unravel the intricate workings of malicious code. This research paper presents a comprehensive exploration of the role of reverse engineering in the domain of malware detection and analysis. It delves into the fundamental stages of the reverse engineering process, encompassing code disassembly, static analysis, and dynamic analysis. Additionally, reverse engineering facilitates meticulous analysis of malware, encompassing intricate examination of its structural attributes, operational mechanisms, and behavioral characteristics. However, the landscape of reverse engineering is not devoid of challenges. Malware authors employ sophisticated obfuscation techniques and antianalysis mechanisms to impede reverse engineering endeavors. These measures encompass code encryption, packing, anti-debugging, and anti-virtualization strategies. By providing a comprehensive examination of the important role of reverse engineering in malware detection and analysis, this research paper will elucidate an extensive array of tools and methodologies.

Keywords: Malware, Reverse engineering, Code disassembly, Static analysis, Dynamic analysis

1. INTRODUCTION

The In the modern interconnected digital world, the growing prevalence of malware poses a significant and persistent threat to individuals, organizations, and even nations. Cybercriminals employ cunning tactics, utilizing malware to clandestinely track online activities and extract sensitive information like usernames and passwords from financial websites. Malware includes deliberately crafted programs or files designed to cause harm, infiltrate systems, or disrupt the normal operations of computers, servers, or networks [1]. This insidious software compromises system integrity, granting unauthorized access to confidential data and enabling cybercriminals to secretly monitor targeted computers and their owners. Its covert nature allows malware to remain undetected within a system for extended periods, evading discovery by unsuspecting users. The relentless evolution of malicious software, combined with its sophisticated techniques, underscores the critical need for robust and proactive malware detection and analysis methods. Thus, this study aims to comprehensively discuss the techniques, tools, and approaches used in analyzing malware, shedding light on effective strategies for countering this pervasive threat.

Reverse engineering emerges as a powerful technique empowering researchers and cybersecurity professionals to dissect malware and uncover its inner workings. It serves as a foundational approach for understanding the intricate structures and behaviors exhibited by malicious software. By carefully deconstructing and analyzing the code and functionalities of such programs, reverse engineering provides security experts with invaluable insights into the tactics used by cybercriminals. These insights play a pivotal role in developing effective countermeasures, fortifying existing security systems, and ultimately mitigating the risks posed by sophisticated malware attacks.

In simple terms, reverse engineering is a process that involves disassembling various items, including software, machinery, and architectural structures, to extract design data. It is also known as back engineering. In the context of malware analysis, reverse engineering focuses on disassembling the components of complex software products. Through this method, researchers gain a deeper understanding of the underlying principles, applications, stages, and future prospects of reverse engineering. Moreover, the article highlights the evolving nature of reverse engineering and its profound impact on the realm of cybersecurity.

2. LITERATURE REVIEW

Numerous research articles related to malware analysis using reverse engineering have been encountered, highlighting its wide-ranging applications. These articles encompass a diverse range of methodologies, including machine learning, artificial intelligence, and other innovative techniques. However, recent research articles have emphasized the utilization of both static and dynamic analysis, as well as hybrid analysis, to investigate malware.

One previous literature provided a comprehensive description of static and dynamic analysis, along with the tools and techniques employed. The authors explored the deep-seated connection between malware samples and the dark web, while also examining potential threats. The primary aim of reverse engineering, as discussed, is to replicate or enhance the functionality of the original product by identifying the underlying solution [2].

In another paper, an extensive analysis of static and dynamic analysis techniques was presented, accompanied by a detailed classification. The authors presented an implementation of malware analysis using these methods to provide guidance and an overview of the malware analysis process. They highlighted the advantages of static and dynamic analysis, as well as the associated challenges. The paper concluded by emphasizing the need to minimize the time required for malware analysis while still obtaining detailed results from the analysis.

3. MALWARE DETECTION AND PREVENTION

The protection against malware and its prevention play a vital role in maintaining the security and integrity of computer systems, networks, and sensitive data. Recognizing the critical significance of effective malware detection and prevention is essential to shield against the detrimental impacts caused by malware attacks. One of the primary reasons why these measures are indispensable is their role in safeguarding an organization's reputation[3]. Successful malware attacks have the potential to compromise customer data, violate privacy, and disrupt services, leading to a loss of trust from customers, damage to the brand's image, and potential legal consequences. Neglecting robust malware detection and prevention strategies can result in lasting damage to an organization's reputation, which can be extremely challenging to recover from.

Additionally, malware can cause significant financial losses. Cybercriminals use malware to gain unauthorized access to financial information, carry out fraudulent activities, and extort money from individuals and organizations. This can impact personal finances, disrupt business operations, and even create broader financial instability. Moreover, malware poses a critical threat to the loss of valuable data and intellectual property[4]. Advanced forms of malware are specifically designed to infiltrate systems, extract sensitive information, and gain unauthorized access to valuable intellectual property, leading to substantial losses, compromised business strategies, and erosion of competitive advantage. Furthermore, malware has the capability to disrupt the normal functioning of computer systems and networks, resulting in significant operational disruptions and downtime. This can cause lost productivity, hinder business continuity, and lead to potential financial ramifications.

To tackle the multifaceted nature of the malware threat, researchers are exploring diverse methodologies, including signature-based identification, behavior-based detection, anomaly detection, and proactive preventive measures. By delving into these methodologies, this study aims to provide valuable insights and discoveries, serving as a valuable resource for cybersecurity professionals, researchers, and policymakers striving to mitigate the widespread consequences of malware.

3.1. Signature-based Detection

Signature-based detection is a malware detection approach that involves comparing code bytes with pre-existing malware signatures stored in a Blacklist database. This method relies on distinct patterns to identify most malware instances, making it a widely used technique[5]. While newer and advanced detection methods have emerged, signature-based detection remains relevant and beneficial in specific scenarios. It offers several advantages:

Efficiency and Speed: Signature-based detection quickly scans files and systems using predefined patterns, making it time-efficient, especially during large-scale malware outbreaks or when a rapid response is needed.

Accuracy for Known Threats: It excels at identifying known malware strains by comparing files against an extensive signature database, effectively combating established threats.

Low False Positives: Signature-based detection generates fewer false alarms, minimizing the likelihood of legitimate files being flagged as malware and reducing disruptions to normal operations.

Ease of Implementation: Implementing this method is straightforward once the signature database is created and updated regularly, making it accessible for organizations with limited resources.

Cost-Effectiveness: It requires fewer computational resources, making it cost-effective for budget constrained organizations without the need for specialized equipment or extensive training.

Despite these strengths, signature-based detection has limitations. It struggles to detect new or zero-day malware without known signatures and can be evaded through obfuscation or encryption techniques employed by adversaries. To enhance malware detection efficacy and provide comprehensive defense against evolving threats, a combination of techniques such as behavior-based analysis, machine learning, and heuristics is recommended. Some of the limitations and the need for alternative detection methods include[6]:

Zero-Day Attacks: Signature-based detection may fail to detect zero-day attacks, which are newly discovered vulnerabilities or malware variants lacking known signatures.

Signature Updates: Timely updates to the signature database are crucial for maintaining the effectiveness of signature-based detection.

Polymorphic Malware: Signature-based detection faces challenges in identifying polymorphic malware due to its constantly changing patterns through code obfuscation.

3.2. Heuristic-based Detection

Heuristic malware detection methods utilize data mining and machine learning techniques to understand the behavioral patterns exhibited by executable files[7]. This approach focuses on analyzing file behavior and characteristics to assess potential threats, specifically targeting the identification of unknown or zero-day malware. By scrutinizing factors such as file activities, system interactions, code analysis, and anomaly detection, heuristic-based detection improves the capability to recognize suspicious or potentially dangerous files without known signatures. Leveraging heuristic analysis enables security systems to effectively identify and mitigate emerging threats, enhancing overall cybersecurity defenses. Some advantages and limitations of heuristic-based detection compared to signature-based detection are described below.

Detection of Unknown or Zero-Day Malware:
Heuristic-based detection effectively identifies previously unknown or zero-day malware by analyzing behavioral patterns and characteristics, detecting emerging threats before they are identified.

Adaptability to New Threats: Since it focuses on behavior rather than specific signatures, heuristic-based detection can detect variations and new strains of malware that may evade traditional signature-based methods.

Early Detection: By analyzing file activities, system interactions, and code, heuristic-based detection identifies potential threats in their early stages.

However, there are certain limitations associated with heuristic-based detection[8].

> **False Positives:** Heuristic-based detection may generate false positives, flagging legitimate files or programs as malicious due to certain behavioral patterns resembling malware. This can lead to unnecessary alerts and disruptions.
>
> **False Negatives:** Heuristic-based detection may also result in false negatives, failing to identify certain types of malware or sophisticated attack techniques, leaving systems vulnerable to undetected threats.
>
> **Performance Impact:** Real-time monitoring and analysis involved in heuristic-based detection can strain system resources, impacting overall system performance.
>
> **Continuous Updates and Maintenance:** Regular updates and maintenance are required for heuristic-based detection methods to keep up with evolving malware techniques, updating rules, algorithms, and behavioral models to effectively detect new threats.

Despite these limitations, heuristic-based detection remains a valuable tool in the fight against malware, complementing other detection methods to provide a comprehensive defense against a wide range of threats.

4. TOOLS AND TECHNIQUES IN REVERSE ENGINEERING

The significance of tools and techniques in reverse engineering cannot be overstated, as they play a crucial role in various domains, including software security, intellectual property protection, software maintenance, vulnerability analysis, and malware detection. These advanced tools empower analysts to delve deep into complex systems, unravel their inner workings, and comprehend their intricate architectural design. By utilizing debuggers and emulators, analysts can meticulously scrutinize program behavior during runtime, thoroughly inspect memory contents, and trace intricate paths of execution. This dynamic analysis approach helps in uncovering and understanding potential security vulnerabilities, such as buffer overflows, code injection points, and insecure cryptographic implementations. Such insights are invaluable for fortifying software against malicious attacks and enhancing overall system resilience. Additionally, these tools play a crucial role in safeguarding intellectual property rights. Decompilers, in particular, enable the retrieval of highlevel source code from compiled binaries, facilitating the identification of infringements and enabling appropriate legal action [9].

In the era of software maintenance and re-engineering, reverse engineering tools assume a vital role in dealing with legacy systems characterized by outdated documentation or code bases. Analysts leverage disassemblers and decompilers to reverse engineer software, extract relevant information, and gain a comprehensive understanding of the system's structure and behavior[10]. This information is invaluable for identifying code defects, improving software quality, and effectively updating the software to align with current standards and requirements. Furthermore, reverse engineering tools are indispensable in the field of malware analysis and detection. Equipped with disassemblers and debuggers, analysts can dissect malware samples, comprehend their behavior, identify potential attack vectors, and extract critical indicators of compromise (IOCs). Reverse engineering tools act as a catalyst for uncovering the functionality of malware, such as information theft, remote control, or persistence mechanisms. This knowledge drives the development of effective countermeasures, aids in updating antivirus signatures, and enhances network security to safeguard against an ever-evolving threat landscape. Reverse engineering necessitates the utilization of indispensable tools like debuggers and disassemblers, which bestow analysts with invaluable capabilities to scrutinize and comprehend intricate software systems. These tools empower analysts to navigate complex software landscapes and extract vital insights that contribute to various domains, ultimately enhancing the security, maintenance, and overall resilience of software systems [11].

4.1. Disassembler and Decompiler

A disassembler is a powerful tool used in reverse engineering to convert machine code or compiled binaries into humanreadable assembly code. It provides analysts with profound insights into the low-level operations of a program, enabling them to understand its internal mechanisms better. Let's explore the characteristics, advantages, and disadvantages of disassemblers. Characteristics of disassemblers are [12]:

Reverse Engineering: Disassemblers facilitate the reverse engineering of compiled code, revealing the underlying assembly instructions and aiding in comprehending the program's behavior, identifying vulnerabilities, and conducting security analysis.

Control Flow Analysis: Disassemblers assist in analyzing the control flow of a program by highlighting the sequence of instructions, helping identify loops, conditionals, function calls, and other control structures.

Symbolic Execution: Advanced disassemblers support symbolic execution, allowing analysts to reason about the program's behavior without directly executing it, instrumental in identifying vulnerabilities.

Platform Independence: Disassemblers are versatile tools capable of analyzing binaries from diverse platforms, making them invaluable for cross-platform analysis and compatibility assessment.

Code Annotation and Documentation: Some disassemblers allow annotating and documenting the disassembled code, enhancing code readability and facilitating collaboration among researchers.

Pros of using disassemblers include In-Depth Insights, Versatility, and Debugging Capabilities, while some cons include Lack of High-level Context, Obfuscation Challenges, and Maintenance Overhead. Examples of disassemblers are [13].

IDA Pro: Widely acclaimed as a comprehensive and robust disassembler, IDA Pro offers advanced analysis capabilities and supports various architectures and file formats.

Ghidra: Developed by the National Security Agency (NSA), Ghidra is a feature-rich and extensible disassembler and reverse engineering framework that provides a range of analysis tools.

Binary Ninja: Known for its modern and user-friendly interface, Binary Ninja is a popular disassembler with interactive features and powerful analysis plugins.

On the other hand, decompilers are sophisticated tools used in reverse engineering to convert compiled binaries or machine code back into high-level programming languages. This allows analysts to comprehend and modify the original source code, playing a crucial role in understanding the functionality and structure of software systems. Let's explore the intricacies of decompilers, including their unique features, advantages, and disadvantages. Characteristics of decompilers are.

Reverse Engineering: Decompilers enable analysts to reverse engineer compiled code, providing insights into the original source code and facilitating a deeper understanding of the program's logic and design.

High-Level Representation: Decompilers generate a high-level representation of the decompiled code, resembling the original source code, aiding analysts in comprehending the code's logic and facilitating further analysis and modifications.

Control Flow Reconstruction: Decompilers reconstruct the control flow of a program, including loops, conditionals, and function calls, helping understand the program's behavior and structure.

Variable and Function Naming: Advanced decompilers assign meaningful names to variables and functions in the decompiled code, enhancing code readability and comprehension.

Type Inference: Some decompilers perform type inference, deducing data types of variables and expressions, aiding in understanding the data flow and improving code comprehension.

Advantages of decompilers include Source Code Reconstruction, Vulnerability Discovery, Software Maintenance, and Legacy Code Understanding. Examples of Decompilers are [14].

IDA Pro: In addition to being a powerful disassembler, IDA Pro also offers decompilation capabilities, providing a high-level representation of the decompiled code.

Hex-Rays Decompiler: This commercial decompiler, integrated with IDA Pro, is renowned for its accuracy and ability to handle complex code structures.

4.2. Debuggers and Emulators

Debuggers are essential software tools used in software development and reverse engineering to analyze and understand program behavior during runtime [25]. They offer a wide range of features that help efficiently identify and resolve issues. Let's explore the unique features, advantages, and disadvantages of debuggers, along with some examples. Characteristics of debuggers are:

Breakpoints: Debuggers allow analysts to set breakpoints at specific code lines, pausing program execution at those points for detailed examination. This feature helps understand the execution flow and identify problematic areas.

Step-by-Step Execution: Debuggers enable code execution step-by-step, allowing analysts to closely scrutinize the program's state at each step. This aids in discovering bugs, logic errors, and unexpected behaviors.

Variable Inspection: Analysts can inspect variable values at any execution point using debuggers. This feature assists in understanding the program's state and diagnosing issues related to incorrect variable assignments or calculations.

Real-Time Analysis: Debuggers provide a dynamic analysis environment, allowing analysts to observe program behavior in real-time. This insight helps promptly identify and resolve issues.

Call Stack Analysis: Debuggers offer insights into the call stack, revealing the sequence of function calls and their order of occurrence. This helps understand program flow and identify errors associated with function calls.

Debuggers play a crucial role in software development, reverse engineering, and debugging processes. They empower analysts to understand program behavior, identify issues, and enhance software quality. However, it is essential to use debuggers judiciously, considering performance implications and adhering to legal and ethical considerations.

On the other hand, emulators are software or hardware systems designed to replicate the functionality of another system or device, enabling it to run on a different platform. Emulators find applications in various domains, such as gaming, software development, and system testing. They offer unique features, advantages, and disadvantages, along with certain limitations [15]. Let's explore the intricacies of emulators, including their features, advantages, and disadvantages, along with some examples. Characteristics of emulators are:

Platform Replication: Emulators replicate the hardware architecture and software environment of a target system, enabling programs or games designed for that system to operate on a different platform. This facilitates crossplatform compatibility and the execution of legacy software.

Performance Optimization: Emulators often incorporate performance optimization features to enhance the execution speed of the emulated system. These optimizations may include dynamic recompilation, justin-time compilation, or hardware acceleration to achieve acceptable performance levels.

Debugging and Analysis Tools: Emulators provide debugging and analysis tools that aid in software development and system analysis. These tools enable code inspection, memory monitoring, and performance profiling, allowing developers to diagnose and rectify issues efficiently.

Peripheral and Input Simulation: Emulators can simulate various peripherals and input devices of the emulated system. This includes emulating controllers, keyboards, mice, touchscreens, and other input/output devices, providing a complete user experience.

State Saving and Load: Emulators often offer the capability to save and load the state of the emulated system. This allows users to pause and resume emulation at any point, making it convenient for testing, debugging, and preserving progress in games or applications.

While emulators offer significant advantages, such as cross platform compatibility and preservation of legacy systems, they introduce additional overhead that can impact the performance of the emulated system [28]. The extent of performance degradation depends on factors such as hardware specifications, complexity of emulation, and employed optimization techniques. Examples of emulators:

Dolphin: Dolphin is a widely used emulator for Nintendo GameCube and Wii games, providing accurate emulation and numerous features for customization and enhancement.

QEMU: QEMU is a versatile emulator that supports various hardware architectures and can emulate entire computer systems, making it useful for virtualization and cross-platform development.

BlueStacks: BlueStacks is an emulator specifically designed for running Android applications on Windows and macOS systems, allowing users to experience Android apps in a desktop environment.

Emulators serve as powerful tools in various domains, facilitating software execution on different platforms, aiding in testing and development, and preserving legacy systems. However, it is crucial to consider performance implications and legal considerations while utilizing emulators.

5. MALWARE ANALYSIS USING REVERSE ENGINEERING

5.1. Static Analysis

Static analysis refers to the examination of software programs without executing them. This analysis can be applied to various representations of the software, including the binary representation. When source code is compiled into a binary executable, some information is lost, making it more challenging to study the code[16]. In the context of static PE malware detection using deep learning, there are two primary frameworks for feature extraction. The traditional approach is based on feature engineering, where features are manually extracted from specific file formats. All possible features are then aggregated into a total feature vector. While the advantage is that the extracted features are meaningful and can parse each section of a PE file separately, the downside is that it requires a lot of effort, and there's no guarantee that the extracted features will be practically useful[17].

Analyzing a given binary without executing it is often done manually. For example, if the source code is available, various interesting information, such as data structures and used functions, can be extracted. However, this information gets lost once the source code is compiled into a binary executable, making further analysis challenging. There are different techniques used for static malware analysis:

File Fingerprinting: This involves examining external features of the binary, such as computing a cryptographic hash (e.g., md5) to distinguish it from others and verify its integrity.

File Format Analysis: Leveraging metadata of a given file format can provide useful information. For example, from a Windows binary in PE format, details like compilation time, imported/exported functions, and strings can be extracted.

AV Scanning: Checking the binary against known malware in antivirus scanners can be time-consuming but is necessary at times.

Packer Detection: Malware is often distributed in an obfuscated form using packers, making it difficult to recover logic and metadata. Finding unpackers for specific cases can be challenging.

Disassembly: The main part of static analysis involves disassembling the binary, converting machine code to assembly language. This allows analysts to examine program logic and understand its purpose.

The main advantage of static malware analysis is that it allows a comprehensive analysis of a binary, covering all possible execution paths. It is also generally safer than dynamic analysis since the source code is not executed. However, it can be time-consuming and requires expertise. Malware samples' source code is typically not readily available, limiting the applicable static analysis techniques to those based on the binary representation. Moreover, static analysis becomes more complex when dealing with malicious code intentionally designed to resist analysis.

5.2. Dynamic Analysis

Dynamic malware analysis involves executing a given malware pattern within a controlled environment to monitor its actions and analyze its malicious behavior during runtime. Unlike static analysis, dynamic analysis overcomes the unpacking difficulty as the malware unpacks itself, providing a clear view of the program's actual behavior. However, dynamic analysis has some drawbacks, including incomplete code coverage due to monitoring only one execution path, and the risk of harming third-party systems if the analysis environment is not properly isolated. Moreover, malware samples may alter their behavior or cease execution altogether when detected in a managed analysis environment. There are two basic techniques for dynamic malware analysis:

Analyzing the Difference Between Defined Points: Malware is executed for a specific period, and then the changes made to the system are analyzed by comparing it to the initial state. This method provides a comparison report on the behavior of the malware.

Observing Runtime Behavior: Malicious activities launched by the malware application are monitored during runtime using specialized tools, often in a sandboxed environment that isolates the malicious process from the rest of the system using virtualization mechanisms[18].

For example, the Regshot tool allows taking snapshots of the registry before and after executing the binary and then comparing the two snapshots to identify file additions and changes. Dynamic analysis involves executing the malicious sample and monitoring its behavior, focusing on API calls and system invocations. Analyzing the parameters passed to these functions provides insights into the sample's interactions with the environment and how it processes sensitive data.

While dynamic analysis is a powerful method, static analysis based systems also exist, although they are less popular due to malware's protection against static methods. Malware often employs code obfuscation, encryption, and runtime packing to evade disassembly. The main advantage of static analysis lies in its ability to reason about all possible execution paths of the malware, while dynamic analysis is limited to a single execution path.

Dynamic malware analysis offers valuable insights into malware behavior during runtime, but it is not the only approach for analyzing malicious binaries. Static analysis based systems exist but face challenges due to the complexity of modern malware protection techniques. Both methods have their strengths and limitations, and a combination of approaches is often necessary to comprehensively analyze and combat malware threats.

6. CHALLENGES AND FUTURE DIRECTION

Although we have discussed numerous advantages and the necessity of reverse engineering in the field of Malware Detection and Analysis, there are challenges inherent in each developmental phase. It is crucial to acknowledge and address these challenges to establish reverse engineering as the optimal approach for Malware Analysis. The utilization of reverse engineering for malware detection and analysis presents several arduous obstacles that researchers and cybersecurity professionals must overcome. These challenges encompass :

Code Obfuscation: Malware authors frequently employ diverse obfuscation techniques with the deliberate intention of concealing the true nature and functionality of their malicious code. This deliberate obfuscation complicates the reverse engineers' comprehension of the underlying logic and objectives of the malware.

Anti-Analysis Techniques: Advanced malware strains are equipped with anti-analysis mechanisms designed to detect and thwart reverse engineering endeavors. These techniques may involve the identification of virtual environments, anti-debugging measures, or self-destruct mechanisms, rendering the extraction of meaningful insights from the malware a daunting task [16].

Complex Control Flows: Malware often employs intricate control flows, such as loop unrolling, encryption, or polymorphism, to obfuscate the understanding of its execution path. Analyzing such convoluted control flows can significantly impede the reverse engineering process.

Time and Resource Intensiveness: Reverse engineering demands expertise, patience, and substantial computational resources, making it a laborious and time consuming task. Unraveling the inner workings of complex malware and comprehending its intricacies can be a time-intensive endeavor, especially when dealing with highly sophisticated and intricate strains.

Rapidly Evolving Malware: Malware authors continuously evolve their techniques to evade detection and analysis. This necessitates that reverse engineers constantly update their skills, adapt to new evasion tactics, and remain abreast of emerging malware trends and strategies.

Legal and Ethical Considerations: Reverse engineering malware raises legal and ethical concerns as it involves analyzing software without the explicit consent of its creators. Researchers must adhere to relevant laws, regulations, and ethical guidelines to ensure their activities remain within legal boundaries and uphold ethical standards.

Overcoming these challenges necessitates a combination of technical expertise, innovative approaches, collaboration within the cybersecurity community, and a steadfast commitment to remain informed about the latest advancements in malware analysis and reverse engineering techniques.

Reverse engineering is becoming a prominent trend in the realm of Malware Analysis. However, it requires refinement in light of the challenges we discussed earlier. To establish it as the best approach, further creative work and mind-blowing techniques are necessary. In the field of Malware Detection and Analysis using reverse engineering, numerous future directions hold promise for advancements and research [13]. By exploring these directions, researchers can contribute to the continuous development of effective strategies and tools to combat the evolving landscape of malware threats. These directions include:

Automation and Machine Learning: Integrating automation and machine learning techniques can significantly enhance the efficiency and effectiveness of malware detection and analysis using reverse engineering. Developing intelligent algorithms and models capable of automatically analyzing and classifying malware samples can expedite the detection process, especially in the face of ever-evolving malware threats.

Behavioral Analysis: While static analysis of malware code is commonly employed, future research can focus on advancing behavioral analysis techniques. By monitoring the runtime behavior of malware samples, researchers can gain insights into the dynamic actions and interactions of malicious code, enabling more accurate and comprehensive detection and analysis.

i. **Threat Intelligence and Sharing:** Collaboration and information sharing among cybersecurity professionals, researchers, and organizations are critical in combating the evolving landscape of malware. Future research can explore frameworks and platforms that facilitate the sharing of threat intelligence, enabling faster detection, analysis, and response to new malware strains.

ii. **Evasion Techniques and Countermeasures:** As malware authors continue to develop sophisticated evasion techniques, future research should concentrate on identifying and understanding these techniques to develop effective countermeasures. This involves studying advanced obfuscation, anti-analysis, and evasion mechanisms employed by malware and devising strategies to overcome them.

iii. **Hardware-Level Analysis:** Traditional malware analysis primarily focuses on software-level analysis. However, future research can explore hardware-level analysis techniques, such as firmware analysis and hardware emulation, to uncover malware operating at a lower level, beyond the scope of traditional software-based analysis.

iv. Internet of Things (IoT) Malware: With the proliferation of IoT devices, the threat of malware targeting these interconnected devices is increasing. Future research can delve into reverse engineering techniques specifically tailored for IoT malware detection and analysis, addressing the unique challenges posed by this rapidly expanding ecosystem.

v. Privacy-Preserving Analysis: Malware analysis often involves handling sensitive and confidential data. Future research can explore techniques that ensure privacy preservation during reverse engineering and analysis processes, enabling effective analysis while safeguarding the privacy of individuals and organizations involved.

Real-Time and Proactive Detection: Malware attacks are growing more sophisticated and occur in real-time. Future research can focus on developing real-time and proactive detection mechanisms using reverse engineering. This involves continuous monitoring, analysis, and early detection of malware to minimize the impact and prevent further propagation [18].

7. CONCLUSION

In conclusion, this research paper has successfully outlined and discussed the main findings derived from the analysis and detection of malware using reverse engineering methods. The study has shed light on crucial insights obtained through the examination of malware samples using reverse engineering techniques, resulting in notable advancements in the field of malware analysis and detection. Notably, the paper has addressed existing limitations and bridged gaps identified in previous literature. The contributions of this research paper to the field of malware analysis and detection should not be underestimated. By employing reverse engineering techniques, this study has deepened the current understanding of malware behavior, exposed new techniques for malware detection, and enhanced the overall capabilities of the field. The findings have brought us closer to a more robust and effective approach to combating malware threats. Based on the outcomes of this research, several recommendations can be made for future research and development endeavors. Firstly, it is crucial to investigate advanced obfuscation techniques employed by malware authors to counteract detection efforts. By exploring ways to overcome obfuscation, researchers can stay ahead of evolving malware tactics. Additionally, efforts should be directed towards improving anti-reverse engineering methods to mitigate the impact of such techniques on analysis. The integration of artificial intelligence and machine learning into malware detection processes also shows promise and warrants further exploration. Lastly, enhancing the effectiveness of hybrid analysis approaches, which combine static and dynamic analysis techniques, is an area that should be prioritized for future development. The recommendations put forth in this paper will guide future efforts and enable researchers and practitioners to combat emerging malware threats effectively.

REFERENCES

[1] A. A. Malik, M. Asad and W. Azeem, “Detection and Control over the offences of White Collar Crime, Fraud and Hacking of information by using effectively the relevant software and Electronic Devices”, International Journal for Electronic Crime Investigation, vol. 7, no. 1, pp. 1-8, 2023.

[2] A. A. Malik, M. Asad and W. Azeem, “Frauds in Banking and entrepreneurs by electronic devices and combating using Software and employment of demilitarized zone in the Networking”, International Journal for Electronic Crime Investigation, vol. 6, no. 4, pp. 1-8, 2022.

[3] A. A. Malik, W. Azeem and M. Asad, “Role of Legislation, need of strong Legal Framework and Procedures to Contest Effectively with Cybercrime and Money Laundering”, International Journal of Crimes investigation, vol. 6, no. 2, pp. 1-8, 2022.

[4] A. A. Malik, W. Azeem and M. Asad, “Requirement of strong legal frame work and procedures to contest with Cybercrime in Pandemic Situation”, International Journal for Electronic Crimes Investigation, vol 5, no. 1, pp. 3-12, 2021.

[5] A. A. Malik, M. Asad and W. Azeem, “To Combat White Collar Crimes in Public and Private Sector and Need for Strong Legislation and Ethics”, International Journal for Electronic Crimes Investigation, vol. 4, no. 3, pp.1-8, 2020.

[6] A. A. Malik, “Standardization of forensic evidence its procurement preservation and presentation in court of using FBI techniques by FIA”, International Journal for Electronic Crimes Investigation, vol. 4, no. 1, pp. 1-6, 2020.

[7] A. A. Malik, "Bank Frauds Using Digital Devices and the Role of Business Ethics", International Journal for Electronic Crimes Investigation, vol. 2, no. 4, pp. 1-9, 2020.

[8] Pakistan's Payment System and Electronic Fund Transfers Act, 2007.

[9] A. Aseri, “Security issues for online shoppers”, International Journal of Scientific and Technology Research, vol. 10, no. 3, 2021.

[10] U. Doloto and Y. H. Chen-Burger, “A Survey of Business Models in eCommerce, In Agent and Multi-Agent Systems: Technologies and Applications”, 9th KES International Conference, KES-AMSTA, pp. 249-259, 2015.

[11] A. Hoek, D. Pearson, S. James, M. Lawrence and S. Friel, “Shrinking the food-print: A qualitative study into consumer perceptions, experiences and attitudes towards healthy and environmentally friendly food behaviors”, Appetite, pp. 117-131, 2017.

[12] A. Lugmayr and J. Grueblbauer, “Review of information systems research for media industry–recent advances, challenges, and introduction of information systems research in the media industry”, Electronic Markets, vol. 27, no. 1, pp. 33-47, 2017.

[13] S. Gensler, F. Volckner, M. Egger, K. Fischbach and D. Schoder, “Listen to your customers: Insights into brand image using online consumer-generated product reviews”, International Journal of Electronic Commerce, vol. 20, no. 1, pp. 112-141, 2015.

Editorial Policy and Guidelines for Authors

IJECI is an open access, peer reviewed quarterly Journal published by LGU. The Journal publishes original research articles and high quality review papers covering all aspects of crime investigation.

The following note set out some general editorial principles. All queries regarding publications should be addressed to editor at email IJECI@lgu.edu.pk. The document must be in word format, other format like pdf or any other shall not be accepted.
The format of paper should be as follows:

- Title of the study (center aligned, font size 14)
- Full name of author(s) (center aligned, font size 10)
- Name of Department
- Name of Institution
- Corresponding author email address.
- Abstract
- Keywords
- Introduction
- Literature Review
- Theoretical Model/Framework and Methodology
- Data analysis/Implementation/Simulation
- Results/ Discussion and Conclusion
- References.

Heading and sub-heading should be differentiated by numbering sequences like, 1. HEADING (Bold, Capitals) 1.1 Subheading (Italic, bold) etc. The article must be typed in Times New Roman with 12 font size 1.5 space, and should have margin 1 inches on the left and right. Table must have standard caption at the top while figures below with. Figure and table should be in continues numbering. Citation must be in according to the IEEE style.

LAHORE GARRISON UNIVERSITY

Lahore Garrison University has been established to achieve the goal of excellence and quality education in minimum possible time. Lahore Garrison University in the Punjab metropolis city of Lahore is an important milestone in the history of higher education in Pakistan. In order to meet the global challenges, it is necessary to touch the highest literacy rates while producing skillful and productive graduates in all fields of knowledge.

VISION

Our vision is to prepare a generation that can take the lead and put this nation on the path to progress and prosperity through applying their knowledge, skills and dedication. We are committed to help individuals and organizations in discovering their God-gifted potentials to achieve ultimate success actualizing the highest standards of efficiency, effectiveness, excellence, equity, trusteeship and sustainable development of global human society.

MISSION

At present, LGU is running Undergraduate, Graduate, Masters, M.Phil. and Ph.D. programs in various disciplines. Our mission is to serve the society by equipping the upcoming generations with valuable knowledge and latest professional skills through education and research. We also aim to evolve new realities and foresight by unfolding new possibilities. We intend to promote the ethical, cultural and human values in our participants to make them educated and civilized members of society.

Contact: For all inquiries, regarding call for papers, submission of research articles and correspondence, kindly contact at this address:
Sector C, DHA Phase-VI Lahore, Pakistan
Phone: +92- 042-37181823
Email: ijeci@lgu.edu.pk

Copyright @ 2017, Lahore Garrison University, Lahore, Pakistan. All rights reserved.

Published by: Digital Forensics Research and Service Center , Lahore Garrison University